Faster and easier: cross-validation and model robustness checks

by

William T. Stephenson
S.M., Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Electrical Engineering and Computer Science
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2022

© 2022 Massachusetts Institute of Technology. All Rights Reserved.
The author here by grants to MIT the permission to reproduce and to distribute publicly paper and electronic copies of the thesis document in whole or in part in any medium now known or hereafter created.
Faster and easier: cross-validation and model robustness checks
by
William T. Stephenson

Submitted to the Department of Electrical Engineering and Computer Science
on January 18, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Electrical Engineering and Computer Science

Abstract
Machine learning and statistical methods are increasingly used in high-stakes applications – for instance, in policing crime, making predictions about the atmosphere, or providing medical care. We want to assess the extent to which we can trust our methods, though, before we use them in such applications. There exist assessment tools, such as cross-validation (CV) and robustness checks, that help us understand exactly how trustworthy our methods are. In both cases (CV and robustness checks), a typical workflow follows the pattern of “change the dataset or method, and then rerun the analysis.” However, this workflow (1) requires users to specify the set of relevant changes, and (2) requires a computer to repeatedly refit the model. For methods involving large and complex models, (1) is expensive in terms of user time, and (2) is expensive in terms of compute time. So CV, which requires (2), and robustness checks, which often require both (1) and (2), see little use in the large and complex models that need them the most. In this thesis, we address these challenges by developing model evaluation tools that are fast in terms of both compute and user time. We develop tools to approximate CV when it is most computationally expensive: in high dimensional and complex, structured models. But approximating CV implicitly relies on the quality of CV itself. We show theory and empirics calling into question the reliability of the use of CV for quickly and automatically tuning model hyperparameters – even in cases where the behavior of CV is thought to be relatively well-understood. On the front of robustness checks, we note that a common workflow in Bayesian prior robustness requires users to manually specify a set of alternative reasonable priors, a task that can be time consuming and difficult. We develop automatic tools to search for a prediction-changing alternative prior for Gaussian processes, saving users from having to manually specify the set of alternative priors.

Thesis Supervisor: Tamara Broderick
Title: Associate Professor of Electrical Engineering and Computer Science
Acknowledgments

I firstly have to thank my advisor, Tamara Broderick. The vast majority of what I learned in graduate school has come from working with Tamara. Tamara has been an unending source of critical, but positive, patient, and encouraging feedback throughout the years I have been at MIT. Graduate school is often a struggle, but having Tamara for support and guidance has really helped me along the way. Her generosity with her time and energy has made me a better writer, communicator, critical thinker, and researcher overall.

I also want to thank my committee members, Alan Willsky and Stefanie Jegelka, for taking the time to be on my committee and providing helpful suggestions on earlier versions of this thesis.

Having a fantastic advisor helps with the struggles of graduate school, as does having an incredible research group. Trevor, Sam and Jonathan, as the initially more senior group members, were always around to help explain new concepts and help out newer students like myself. Lorenzo was an amazing person to start out in Tamara’s group alongside, and was constantly a fun (and funny) person to sit next to in the office. Sameer was a great officemate to have, and was always so generous with his time in explaining statistics concepts to me. Stan was an awesome person to bounce ideas off of, TA with, and to learn about MIT Supercloud from. Brian was never afraid to push back on my research ideas, leading to many productive conversations (Brian is also a great person to hang out with, etc.). Ryan was a likewise great person to debate with, especially at reading groups which could otherwise run a little dry! Diana organized some great reading groups which got me interested in robustness issues, resulting in some of the work in this thesis. Raj, Miriam, Yaroslav, Mikolaj, Richard, and Hannah, I wish we had had more time to work with and alongside each other, but it was wonderful being around such talented and engaged people nevertheless.

I also want to thank my co-authors and collaborators outside MIT. Soumya has been an amazing close collaborator on two of the chapters in this thesis; they would likely not have been so fleshed out if it were not for him. Misha was also a great source of insight on the work in this thesis, especially the chapter on Gaussian processes. I am so glad that Madeleine happened to hear about our work at a conference and reached out; her expertise on numerical linear algebra and overall insights as a researcher helped with a number of chapters in this thesis. Finally, Zach was a great person to discuss some of the ideas in this thesis with; I wish him the best of luck throughout the rest of his time as a graduate student!

To my family who have been with me every step of the way throughout my life: thank you. And to my friends who I met as early as elementary school and as recently as the last year: thank you for being such wonderful friends. I would not have made it without such great friends and family members.
Contents

1 Introduction

2 Approximate cross-validation for sparse models in high dimensions
   2.1 Introduction ................................................................. 22
   2.2 Overview of Approximations ........................................... 24
   2.3 Problems in high dimensions ......................................... 26
      2.3.1 Sparsity via \( \ell_1 \) regularization ......................... 27
   2.4 Approximation quality in high dimensions ....................... 28
      2.4.1 Linear regression .................................................... 30
      2.4.2 Logistic regression .................................................. 31
   2.5 Experiments ................................................................. 34
   2.6 Conclusions and future work ......................................... 35

3 Approximate cross-validation with low-rank data in high dimensions
   3.1 Introduction ................................................................. 36
   3.2 Background: approximate CV methods ............................. 37
   3.3 Methodology ................................................................. 39
   3.4 Accuracy of exact ACV with approximately low-rank data ....... 41
   3.5 Approximating the quadratic forms \( Q_n \) ......................... 43
   3.6 Experiments ................................................................. 45
   3.7 Conclusions ................................................................. 48

4 Approximate cross-validation for structured models
   4.1 Introduction ................................................................. 49
   4.2 Structured models and cross-validation ............................ 50
      4.2.1 Structured models .................................................... 50
      4.2.2 Challenges of cross-validation and approximate cross-validation in structured models ................................. 51
   4.3 Cross-validation and approximate cross-validation in structured models ................................................ 53
   4.4 IJ behavior under inexact optimization ............................ 55
   4.5 Experiments ................................................................. 57
### 5 Can we globally optimize cross-validation loss? Quasiconvexity in ridge regression

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Setup and notation</td>
</tr>
<tr>
<td>5.3</td>
<td>LOOCV loss is typically not convex and need not be quasiconvex</td>
</tr>
<tr>
<td>5.4</td>
<td>What does the quasiconvexity of $L$ depend on?</td>
</tr>
<tr>
<td>5.5</td>
<td>Quasiconvexity of $L$ with a nearly uniform spectrum $S$</td>
</tr>
<tr>
<td>5.6</td>
<td>Theorem 5 in practice</td>
</tr>
<tr>
<td>5.7</td>
<td>Discussion</td>
</tr>
</tbody>
</table>

### 6 Measuring the sensitivity of Gaussian processes to kernel choice

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>6.2</td>
<td>Our Workflow</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Nearby kernels and efficient optimization</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Assessing qualitative interchangeability</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Workflow illustration on synthetic data</td>
</tr>
<tr>
<td>6.3</td>
<td>Stationary perturbations to a model of heart rates</td>
</tr>
<tr>
<td>6.4</td>
<td>Non-stationary perturbations to a model of carbon dioxide emissions</td>
</tr>
<tr>
<td>6.5</td>
<td>Non-stationary perturbations in classifying MNIST digits</td>
</tr>
<tr>
<td>6.6</td>
<td>Discussion</td>
</tr>
</tbody>
</table>

### 7 Conclusions

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Summary</td>
</tr>
<tr>
<td>7.2</td>
<td>Future work</td>
</tr>
</tbody>
</table>

### A Appendix for: Approximate cross-validation for sparse models in high dimensions (Chapter 2)

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Cross-validation methods</td>
</tr>
<tr>
<td>A.2</td>
<td>Scaling of the leave-one-out objective</td>
</tr>
<tr>
<td>A.3</td>
<td>Approximately solving $IJ_n(R)$ and $NS_n(R)$</td>
</tr>
<tr>
<td>A.4</td>
<td>Further details of Eq. (2.2) and Eq. (2.3)</td>
</tr>
<tr>
<td>A.4.1</td>
<td>Derivation of $IJ_n(R)$</td>
</tr>
<tr>
<td>A.4.2</td>
<td>Invertibility in the definition of $NS_n(R)$ and $IJ_n(R)$</td>
</tr>
<tr>
<td>A.4.3</td>
<td>Accuracy of $IJ_n(R)$ for regularized problems</td>
</tr>
<tr>
<td>A.4.4</td>
<td>Derivation of $NS_n(R)$</td>
</tr>
<tr>
<td>A.4.5</td>
<td>Computation time of approximations</td>
</tr>
<tr>
<td>A.5</td>
<td>Derivation of $IJ_n$ and $NS_n$ via smoothed approximations</td>
</tr>
<tr>
<td>A.5.1</td>
<td>Proof of Proposition 12 using Eq. (A.14)</td>
</tr>
<tr>
<td>A.5.2</td>
<td>Proposition 12 using Eq. (A.15)</td>
</tr>
<tr>
<td>A.6</td>
<td>The importance of correct support recovery</td>
</tr>
<tr>
<td>A.7</td>
<td>Details of real experiments</td>
</tr>
<tr>
<td>A.8</td>
<td>Selection of $\lambda$</td>
</tr>
<tr>
<td>A.9</td>
<td>Proofs from Section 2.3</td>
</tr>
<tr>
<td>A.9.1</td>
<td>Assumptions from Li et al. [2015]</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>A.9.2 Local structured smoothness condition (LSSC)</td>
<td>108</td>
</tr>
<tr>
<td>A.9.3 Assumptions 18 to 22 imply supp $\hat{\theta}^n \subseteq S$ for all $n$</td>
<td>109</td>
</tr>
<tr>
<td>A.9.4 Useful results for proving Theorems 2 and 3</td>
<td>109</td>
</tr>
<tr>
<td>A.9.5 Proof of Theorem 2 (Linear Regression)</td>
<td>110</td>
</tr>
<tr>
<td>A.9.6 Linear regression: minimum eigenvalue</td>
<td>112</td>
</tr>
<tr>
<td>A.9.7 Linear regression: incoherence</td>
<td>114</td>
</tr>
<tr>
<td>A.9.8 Linear regression: bounded gradient</td>
<td>117</td>
</tr>
<tr>
<td>A.9.9 Linear regression: $\lambda$ small enough</td>
<td>118</td>
</tr>
<tr>
<td>A.9.10 Proof of Theorem 3 (Logistic Regression)</td>
<td>119</td>
</tr>
<tr>
<td>A.9.11 Logistic regression: lambda min</td>
<td>120</td>
</tr>
<tr>
<td>A.9.12 Logistic regression: incoherence</td>
<td>120</td>
</tr>
<tr>
<td>A.9.13 Logistic regression: bounded gradient</td>
<td>121</td>
</tr>
<tr>
<td>A.9.14 Logistic regression: $\lambda$ small enough</td>
<td>122</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B Appendix for: Approximate cross-validation with low-rank data in high dimensions (Chapter 3)</th>
<th>124</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1 Derivation of $x_n^T \mathcal{I}_n$ and $x_n^T \mathcal{I}_n$</td>
<td>124</td>
</tr>
<tr>
<td>B.2 Comparison to existing Hessian inverse approximation</td>
<td>125</td>
</tr>
<tr>
<td>B.3 Previous ACV theory</td>
<td>126</td>
</tr>
<tr>
<td>B.4 Proofs from Sections 3.3 and 3.4</td>
<td>127</td>
</tr>
<tr>
<td>B.4.1 Proving accuracy of $\mathcal{I}_n$ and $\mathcal{I}_n$ under exact low-rank data</td>
<td>127</td>
</tr>
<tr>
<td>B.4.2 Proving accuracy of $\mathcal{I}_n$ and $\mathcal{I}_n$ under ALR data (Lemma 2)</td>
<td>128</td>
</tr>
<tr>
<td>B.4.3 Proof of Theorem 4</td>
<td>131</td>
</tr>
<tr>
<td>B.4.4 Proof of Corollary 1</td>
<td>132</td>
</tr>
<tr>
<td>B.5 Proofs and discussion from Section 3.5</td>
<td>132</td>
</tr>
<tr>
<td>B.5.1 Proofs</td>
<td>132</td>
</tr>
<tr>
<td>B.5.2 Relation of Algorithm 2 to techniques from randomized linear algebra</td>
<td>133</td>
</tr>
<tr>
<td>B.5.3 Implementation of Algorithm 2</td>
<td>134</td>
</tr>
<tr>
<td>B.6 Error bound experiments</td>
<td>135</td>
</tr>
<tr>
<td>B.7 Real data experiments</td>
<td>135</td>
</tr>
<tr>
<td>B.8 Sensitivity of results to $\lambda$</td>
<td>137</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C Appendix for: Approximate cross-validation for structured models (Chapter 4)</th>
<th>138</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1 Related work: Approximate CV methods</td>
<td>138</td>
</tr>
<tr>
<td>C.2 Hidden Markov random fields</td>
<td>140</td>
</tr>
<tr>
<td>C.3 Leave structure out cross-validation (LSCV)</td>
<td>140</td>
</tr>
<tr>
<td>C.4 Efficient weighted marginalization (WEIGHTEDMARG) for chain-structured MRFs</td>
<td>140</td>
</tr>
<tr>
<td>C.5 Equivalence of weighting (A) and (B) for leave-future-out for chain-structured graphs</td>
<td>141</td>
</tr>
<tr>
<td>C.6 Conditional random fields</td>
<td>142</td>
</tr>
<tr>
<td>C.7 CV for conditional random fields</td>
<td>142</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>C.7.1</td>
<td>LSCV for CRFs</td>
</tr>
<tr>
<td>C.7.2</td>
<td>LWCV for CRFs</td>
</tr>
<tr>
<td>C.8</td>
<td>Computational cost of one Newton-step-based ACV</td>
</tr>
<tr>
<td>C.9</td>
<td>Comparison of approximations afforded by one Newton-step-based and IJ based ACV</td>
</tr>
<tr>
<td>C.10</td>
<td>Derivation of IJ approximations</td>
</tr>
<tr>
<td>C.11</td>
<td>Inexact optimization</td>
</tr>
<tr>
<td>C.12</td>
<td>Experimental details</td>
</tr>
<tr>
<td>C.12.1</td>
<td>Time varying Poisson processes</td>
</tr>
<tr>
<td>C.12.2</td>
<td>Neural CRF</td>
</tr>
<tr>
<td>C.12.3</td>
<td>Philadelphia crime experiment</td>
</tr>
<tr>
<td>C.13</td>
<td>Additional experiments</td>
</tr>
<tr>
<td>C.13.1</td>
<td>Motion capture analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>Appendix for: can we globally optimize cross-validation loss? Quasiconvexity in ridge regression (Chapter 5)</th>
<th>156</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.1</td>
<td>Real dataset descriptions</td>
<td>156</td>
</tr>
<tr>
<td>D.2</td>
<td>Proof of Proposition 9</td>
<td>157</td>
</tr>
<tr>
<td>D.3</td>
<td>Measuring severity of non-quasiconvexity with other losses and optimization methods</td>
<td>158</td>
</tr>
<tr>
<td>D.4</td>
<td>Empirical validation of Assumption 14</td>
<td>160</td>
</tr>
<tr>
<td>D.5</td>
<td>Proof of Theorem 5</td>
<td>161</td>
</tr>
<tr>
<td>D.5.1</td>
<td>Technical Lemmas</td>
<td>165</td>
</tr>
<tr>
<td>D.6</td>
<td>Proof of Corollary 3</td>
<td>169</td>
</tr>
<tr>
<td>D.7</td>
<td>Generating zero-mean orthonormal matrices uniformly at random</td>
<td>170</td>
</tr>
<tr>
<td>D.8</td>
<td>Replicating experiments with error bars</td>
<td>170</td>
</tr>
<tr>
<td>D.9</td>
<td>Implementation details of experiments</td>
<td>172</td>
</tr>
<tr>
<td>D.9.1</td>
<td>Efficiently computing $L(\lambda)$</td>
<td>172</td>
</tr>
<tr>
<td>D.9.2</td>
<td>Numerically checking for quasiconvexity</td>
<td>172</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>Appendix for: Measuring the sensitivity of Gaussian processes to kernel choice (Chapter 6)</th>
<th>173</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.1</td>
<td>Details of spectral density constraints</td>
<td>173</td>
</tr>
<tr>
<td>E.2</td>
<td>Additional details of synthetic-data experiment</td>
<td>174</td>
</tr>
<tr>
<td>E.3</td>
<td>Additional details for the heart rate example</td>
<td>174</td>
</tr>
<tr>
<td>E.4</td>
<td>Additional details for CO$_2$ experiment</td>
<td>176</td>
</tr>
<tr>
<td>E.5</td>
<td>More details on MNIST experiments</td>
<td>178</td>
</tr>
<tr>
<td>E.6</td>
<td>Additional Gram matrix comparisons</td>
<td>180</td>
</tr>
</tbody>
</table>
List of Figures

2-1 Log percent error (Eq. (2.10)) of existing approximate LOOCV methods ("IJ" and "NS") as a function of dataset size $N$ for $\ell_2$ regularized logistic regression. Dashed lines show Eq. (2.2) ("NS") and solid show Eq. (2.3) ("IJ"). Blue lines have fixed data/parameter dimension $D$, while red lines have $D = N/10$, although the true parameter has a fixed support size of $D_{\text{eff}} = 2$ (see Section 2.2 for a full description). IJ and NS fail to capture this low “effective dimension” and suffer from substantially worse performance in high dimensions.

2-2 Error (Eq. (2.10)) across approximations for $\ell_1$ LOOCV (legend shared with Fig. 2-3). The error for $I_{\text{syn}}$ (black dashed) is too small to see, but nonzero; it varies between $-0.06\%$ and $0.04\%$.

2-3 Runtimes for the experiments in Fig. 2-2 with exact CV (red) included for comparison. The $D \times D$ matrix inversion in the smoothed problem is so slow that even exact CV with an efficient $\ell_1$ solver is faster.

2-4 Log percent accuracy (Eq. (2.10)) for real data experiments. For each dataset, we give the accuracy of approximate CV compared to exact CV using both a smoothed approximation to $\ell_1$ and the $I_{\text{syn}}$, $NS_{\text{syn}}$ approximations. For the bcTCGA dataset (linear regression), the nearly quadratic objective seems to be extremely well approximated by one Newton step, making $NS_{\text{syn}}(R^p)$ significantly more accurate than $I_{\text{syn}}(R^n)$; see the note at the end of Appendix A.4.4 about the exactness of $NS_{\text{syn}}(R)$ on quadratic objectives.

2-5 Log runtimes for experiments in Fig. 2-4 with exact CV included for comparison.

3-1 Accuracy of the IJ approximation in Eq. (3.4) for a synthetic Poisson regression problem versus the dataset size $N$. Red shows the accuracy when the data dimension is fixed at $D = 40$, blue when the dimension grows as $D = N/10$, and black when the dimension grows as $D = N/10$ but with a fixed rank of 40. High-dimensional yet low-rank data has identical performance to low-dimensional data.
3-2 Experiments on real data. (Left): average percent error compared to
exact CV on a subset of datapoints, \((1/20) \sum_{b=1}^{20} |x^T_b \theta_b| / |x^T_b \hat{\theta}_b|\),
where approx. denotes NS_{\lambda_n}, NS_{\gamma_n}, IJ_{\lambda_n}, or IJ_{\gamma_n}. (Right): ACV run-
times with exact CV runtimes for comparison. ACV runtimes are given
for all \(N\) datapoints, and exact CV runtimes are estimated runtimes
for all \(N\) datapoints. 45

3-3 Comparisons to alternatives to Algorithm 1 for estimating
\(x^T_n \hat{\theta}_n\) via
the NS approximation on two versions of the blog dataset. (Left):
error versus \(K\) for a \(N = 500, D = 1,000\) subset. (Right): error versus
compute time for a \(N = 8000, D = 4,500\) subset; we use a larger subset
here as the differences in compute time grow larger with larger \(D\). As
the lower SVD / eigendecomposition have the same compute time as
their top counterparts, we omit the lower methods from this plot. 46

3-4 Error bounds implied by Theorem 4 on IJ_{\gamma_n}’s estimate of out-of-sample
error using squared loss, \(Err(x^T_n \theta, y_n) = (e^{x^T_n \theta} - y_n)^2\). (Left): Per data-
point error bounds. The bound is fairly loose for datapoints with larger
squared loss, but tighter for points with lower squared loss. (Right):
Five trials with estimates averaged across all \(n\). We compute the up-
per (respectively, lower) error bars for IJ_{\gamma_n} by averaging the upper (re-
spectively, lower) bounds. While our bounds overstate the difference
between exact CV and IJ_{\gamma_n}, they still give non-vacuous information on
value of exact CV. 47

4-1 Evaluation of approximate LWCV for time-varying Poisson processes.
(Top panel) A subset of the count series. (Lower left six panels) Scatter
plots comparing exact CV loss (horizontal axis) at each point in
each fold (red dots) to our approximation of CV loss (vertical axis).
Black dashed line shows perfect agreement. Three columns for per-
cent points left out; two rows for i.i.d. LWCV (upper) and contiguous
LWCV (lower). (Lower right two panels) Wall-clock time for exact and
approximate CV measured on a 2.5GHz quad core Intel i7 processor
with 16GB of RAM; same rows as left panels. 58

4-2 Behavior of ACV at different epochs in stochastic optimization for
a bilstm-crf. Scatter plots comparing held out probabilities under CV
(horizontal axis) at each point in each fold (red dots) to our approxima-
tion of CV (vertical axis). Black dashed line shows perfect agreement.

4-3 Evaluation of LWCV for loopy Markov random field. (Top left) Cen-
sus tracts data. (Upper right) Wall-clock time of approximate CV and
exact CV. (Lower) Scatter plots comparing CV loss (horizontal axis)
at each point in each fold (red dots) to our approximation of CV loss
(vertical axis). Black dashed line shows perfect agreement. Plots gen-
erated with different values of connectivity \(\beta\). 60
5-1 (Left): Idealized illustration of the leave-one-out CV loss $L$ (blue) and the true out-of-sample loss (black). The minimizer of each curve is marked with a vertical line of the corresponding color. (Center): CV loss for a life-expectancy prediction problem after some standard data pre-processing (Condition 2 of Section 5.2). (Right): CV loss for wine-quality prediction problem on a subset of $N = 50$ data points after standard data pre-processing (Condition 2 of Section 5.2).

5-2 How does quasiconvexity depend on $Y$ and $U$, for data with $N = 3$ and $D = 2$? The left, center, and right panels each correspond to a different setting of the singular values $S$. We divide the unit circle for each of $U$ and $Y$ into $225$ equally spaced points. We examine the severity of non-quasiconvexity in $L$ over this $225 \times 225$ grid. Each grid point is colored by Eq. (5.3), where dark purple corresponds to quasiconvexity (no local minima).

5-3 (Upper left): We generate many datasets and plot the fraction that are non-quasiconvex (“non-QVX”), varying $N$ and the distance of the spectrum from uniformity ($\|S-1\|_1$). (Upper right): We generate two sets (orange, blue) of left-singular vector matrices $U$. In the blue case, we check that the maximum of $\log \|u_{\text{max}}\|_2^2$ across all $U$ for a particular $N$ decreases roughly linearly on a log-log plot (i.e. the blue set satisfies Assumption 14). In the orange case, we check that the minimum of $\log \|u_{\text{max}}\|_2^2$ across all $U$ for a particular $N$ is roughly constant (i.e. the orange set does not satisfy Assumption 14). (Lower): For all the $U$ matrices from the upper right plot, we generate many datasets and plot the fraction that are not quasiconvex.

5-4 Checking Assumption 12. (Left): For each $N$ and each $\nu = \|E\|$, we generate many datasets and check if there is any non-quasiconvex (“non-QVX”) $L$. We plot the largest $\nu$ for which we find only quasiconvex $L$. The growth is roughly linear, which suggests Assumption 12 cannot be loosened. (Right): For each $N$ and each $\nu = \|E\|$, we generate many data sets and plot the fraction of $L$ that are non-quasiconvex. Vertical lines show $\nu_{\text{max}}$ for each $N$.

6-1 Where we sit in the modeling workflow.

6-2 (Far left): Synthetic data. (Center left:) Draws from the original prior $GP(0,k_0)$. (Center right): Draws from $GP(0,k^{(ex)})$. (Far right): Draws from $GP(0,k^{(in)})$. The prior draws are noise-matched to the draws from $k_0$ (Section 6.2.2). The gray area in the right three plots is the 99.7% uncertainty interval for the original prior ($GP(0,k_0)$).

6-3 (Left): Maximal value of the function $F^\star$ as a function of constraint set $\varepsilon$. Comparison of the 2-Wasserstein distance between $k_0(X,X)$ and $k_1(X,X)$ to the posterior variation due to hyperparameter uncertainty for extrapolation (middle) and interpolation (right). The red line corresponds to our decision-changing kernel $k_1$.
6-4  Sensitivity of heart rate analysis in Section 6.3. (Top row): (left) Observed data. (middle) Noise-matched draws from original prior $\mathcal{GP}(0,k_0)$. (right) Alternative prior $\mathcal{GP}(0,k_1)$. (Bottom row): (left) Comparison of the difference between $k_0$ and $k_1$ (red line) to bootstrapped hyperparameter uncertainty (histogram). (middle) Once we expand the constraint set to $\varepsilon = 0.24$ the predicted 95% quantile of heart rate at $x^*$ exceeds 130 bpm (red line). (right) Comparison of posterior distributions of $f^*$ computed using $k_0$ (blue) and $k_1(\varepsilon^*)$ (red). 

6-5  Sensitivity of the Mauna Loa analysis in Section 6.4. (Top-left): Predictions made with the original kernel $k_0$ (black) and a qualitatively interchangeable kernel $k_1$ (red). (Top-right): $F^*$, the mean CO$_2$ level in June 2020, as a function of $\varepsilon$. (Bottom): Noise-matched draws from a $\mathcal{GP}(0,k_0)$ (left) and $\mathcal{GP}(0,k_1(\varepsilon^*))$ (right) prior. See Appendix E.4 for a closer inspection of each prior draw.

6-6  Sensitivity of MNIST analysis in Section 6.5. (Left): $F^*$ as a function of regularizer strength. (Right): Histogram of the 2-Wasserstein distances between the 1000 (one for each test image) input-warped kernel Gram matrices (in red) plotted with those arising from kernel hyperparameter uncertainty around $k_0$ (in black).

A-1  Stochastic Hessian experiments from Appendix A.3. We show percent error of approximation versus compute time for two different dataset sizes. We show three techniques for computing CV: exactly computing CV (black dot, which naturally has 0% error), $IJ_n(R)$ with exactly computing the needed linear systems (red dot), and $IJ_n(R)$ with the stochastic solves described in Appendix A.3 (blue dots, one for each setting of the parameters $K$ and $M$). Values of $M$ and $K$ used are described in Appendix A.3; we use an extended range of settings for the smaller dataset to more extensively illustrate the approximation’s behavior. Settings of $K$ and $M$ for which the stochastic solves are roughly as fast as exactly computing $IJ_n(R)$ result in a significantly less accurate approximation of CV.

A-2  Illustration of the role of support recovery in the accuracy of $IJ_n$ in the case of linear regression. Left: Points show the average of $|\text{supp}\hat{\theta}^n|$ over random values of $n$. Error bars show the min and max $|\text{supp}\hat{\theta}^n|$ over these $n$. For $\lambda = 10.0\sqrt{\log(D)/N}$ (blue), the mean recovered support is constant with $N$. For $\lambda = 1.0\sqrt{\log(D)/N}$ (red), $|\text{supp}\hat{\theta}^n|$ grows with $N$, and varies dramatically for different values of $n$. Right: Percent error (Eq. (2.10)) as $D$ scales with $N$. When the support recovery is constant, we recover an error scaling of roughly $1/N^2$, whereas a growing support results in a much slower decay.
A-3 Experiment for selecting $\lambda$ from Appendix A.8. (Top:) Despite being very accurate for higher values of $\lambda$, the degradation of the accuracy of IJ$_n$ for lower values of $\lambda$ (which corresponds to a larger $S$) causes the selection of a $\lambda$ that is far from optimal in terms of test loss. (Bottom:) For a lower dimensional problem, the curve constructed by IJ$_n$ much more closely mirrors that of exact CV for all values of $\lambda$. In both cases, NS$_n$ performs well.

B-1 Experiment from Appendix B.2. Across four different dataset sizes, using the Neumann series approximation (orange) does not show any noticeable improvement on the time scale of running our approximation (green) for all possible values of $K$.

B-2 Quality of approximation of IJ$_n$ on a synthetic Poisson regression problem using the methods from Section 3.5 (Left): We show three options for the choice of the matrix $\Omega$. Blue shows the choice of $\Omega$ having orthonormal columns selected uniformly at random, orange the optimal choice of $\Omega$ from Proposition 4, and green our approximation to this optimal choice. Percent error $|\text{IJ}_n - \text{\tilde{IJ}}_n|/|\text{IJ}_n|$ is reported to give a sense of scale. (Right): Importance of Proposition 5 for approximating $Q_n$. We show two approximations along with our upper bounds on their error: (1) $Q_n \approx x_n^TH^{-1}_nx_n$ and (2) our recommended $Q_n \approx \text{\tilde{Q}}_n$ from Proposition 5. We report absolute error $|\text{IJ}_n - \text{\tilde{IJ}}_n|$ so that both actual and estimated error can be plotted.

B-3 Experiments from Appendix B.6. (Left): Error bounds can be vacuously large; for trial number 16, our bound exceeds exact CV by two orders of magnitude. (Right): By computing our bound for all $n$ and re-running exact CV for the two largest values, we obtain estimates that are much closer to exact CV.

B-4 Experiment from Appendix B.8. We report average error in the estimate of $Q_n$, $(1/N)\sum_n|Q_n - Q_n|/|Q_n|$ for both the p53 and blog datasets. We note that the errors when using $K = 300, 350, 400$ are visually indistinguishable from one another.

C-1 Comparison of NS and IJ approximate LWCV for time-varying Poisson processes. Scatter plots comparing NS based ACV loss (horizontal axis) at each point in each fold (red dots) to IJ based ACV loss (vertical axis). Black dashed line shows perfect agreement. Left plot contains i.i.d. LWCV results and the right plot contains contiguous LWCV results.

C-2 (Left panel) Error in our approximation relative to exact CV averaged across folds, as a function of wall clock time. (Right panel) Error in our approximation relative to exact CV and averaged across folds, as a function of log gradient norm in the optimization procedure.
C-3 Motion capture analysis through auto-regressive HMMs. (Top) A twelve dimensional MoCap sequence that serves as the observed data and the number of parameters $D$ for different models under consideration. The high dimensionality of the models make alternate ACV methods based on a single Newton step infeasible. (Middle) Scatter plots comparing leave one out loss, where x-axis is $-\ln p(x_n | \Theta(w_{(n)}))$ and y-axis is $-\ln p(x_n | \Theta_{IJ}(w_{(n)}))$ for different auto-regressive orders under exact and IJ approximated leave one out cross validation. Points along the diagonal indicate accurate IJ approximations. (Bottom) Timing and held out negative log probability across different models. For IJ and Exact the error bars represent two jackknife standard error. The IJ approximations are significantly faster but closely approximate exact leave one out loss across models and track well with test loss computed on the held out 20% of the dataset.

C-4 Within sequence leave out experiments. We took the longest MoCAP sequence containing 1484 measurements and fit a five state HMM with Gaussian emissions. We find that even for the MoCAP data IJ approximations to i.i.d. LWCV is very accurate. As the contiguous LWCV involves making larger scale changes to the sequence, for instance at 10% we end up dropping chunks of 140 time steps from the sequences, resulting in larger changes to the parameters, IJ approximations are relatively less accurate. (Top) Scatter plots comparing i.i.d LWCV loss $-\ln p(x_t | x_{[T]}; \Theta(w_o))$ (horizontal axis) with $-\ln p(x_t | x_{[T]}; \Theta_{IJ}(w_o))$ (vertical axis), for each point $x_t$ left out in each fold, computed under exact CV for different omission rates $m\% = 2\%, 5\%, \text{ and } 10\%$ on $M = 10$ trials. (Bottom) Results for contiguous LWCV.

C-5 Leave Future Out CV for MoCAP data on a single MoCAP sequence containing 1484 measurements. The scatter plots compare $-\ln p(x_{T'} | x_{[T]}; \Theta(w_o))$ (horizontal axis) with $-\ln p(x_{T'} | x_{[T]}; \Theta_{IJ}(w_o))$ (vertical axis), with $o = \{ T', T'+1, \ldots T \}$, for some $T' \leq T$; for a five state HMM with Gaussian emissions (left), order 1 auto-regressive emissions (middle), neural auto-regressive emissions (right). The rightmost plot shows the number of parameters in each model. We vary $T'$ from 1337 to 1484 for Gaussian and AR(1) emissions. Since exact fits the NAR model are more expensive we only vary $T'$ between 1455 and 1484 for NAR(1). We find that ACV to be accurate. The NAR model which is an instance of a higher dimensional optimization problem, leads to approximations that are less accurate than the lower dimensional AR(0) and AR(1) cases.
D-1 \((Left)\): Severity of non-quasiconvexity where \(\lambda_{\text{worst-min}}\) is the \(\lambda\) recovered by grid search. \((Right)\): Severity of non-quasiconvexity where \(\lambda_{\text{worst-min}}\) is the \(\lambda\) recovered by gradient descent. Note the color scales from Fig. 5-2 in the main text.

D-2 \((Left)\): Severity of non-quasiconvexity in 5-fold CV. \((Centre)\): Severity of non-quasiconvexity in 10-fold CV. \((Right)\): Severity of non-quasiconvexity in \(\mathcal{L}\), where the quality of \(\lambda_{\text{worst}}, \lambda^*\) and \(\lambda_{\text{worst-min}}\) are measured using the test loss. Note that the color scale on this plot is different from that on previous plots.

D-3 Experiment from Appendix D.4. Orange dots show the decay of \(\|u_{\text{max}}\|^2\) for uniformly drawn non-zero-column-mean orthonormal matrices \(U\); we see that, as proven by Candès and Recht [2009], these matrices satisfy Assumption 14. Blue dots show the decay of \(\|u_{\text{max}}\|^2\) for uniformly drawn zero-column-mean orthonormal matrices. While such matrices are not known to satisfy Assumption 14, we see that their \(\|u_{\text{max}}\|^2\) decays at exactly the same rate as the non-zero-column-mean matrices.

D-4 Replication of Fig. 5-3 with error bars; we repeat the original caption here for reading convenience. \((Upper\ left)\): We generate many datasets and plot the fraction that are not quasiconvex, varying \(N\) and the distance of the spectrum from uniformity \((\|S - 1\|_1)\). \((Upper\ right)\): We generate two sets (orange, blue) of left-singular vector matrices \(U\). In the blue case, we check that the maximum of \(\log\|u_{\text{max}}\|^2\) across all \(U\) for a particular \(N\) decreases roughly linearly on a log-log plot (i.e. the blue set satisfies Assumption 14). In the orange case, we check that the minimum of \(\log\|u_{\text{max}}\|^2\) across all \(U\) for a particular \(N\) is roughly constant (i.e. the orange set does not satisfy Assumption 14). \((Lower)\): For all the \(U\) matrices from the upper right plot, we generate many datasets and plot the fraction that are not quasiconvex.

E-1 Sensitivity of heart rate analysis in Appendix E.3 for an example where we do not find non-robustness. \((Top-left)\): Heart rate data; notice the data is trending downwards at the end of the time series. \((Top-right)\): Prior draws from our original kernel \(k_0\) from Eq. (E.6)). \((Bottom-left)\): Prior draws from our decision-changing kernel \(k_1\) that achieves \(F^* = L\), noise matched by color to the draws from \(k_0\). \((Bottom-right)\): Comparison of the difference between \(k_0\) and \(k_1\) (red line) to posterior hyperparameter uncertainty (histogram).
E-2 Sensitivity analysis of Mauna Loa. Each plot shows noise matched samples from a zero mean Gaussian process with original and perturbed kernel functions. These plots provide a zoomed in view of the prior samples shown in Fig. 6-5. We note that draws from $k_1(\varepsilon^*)$ are in-phase with those of $k_0$ (i.e. $k_1(\varepsilon^*)$ captures the seasonal maxima and minima of CO$_2$ just as well as $k_0$ does). Overall, there is high agreement between functions sampled from the two GPs.

E-3 Additional MNIST experiments. Here we visualize the training and test set performances along the hyperparameter grid used for assessing qualitative interchangeability. The train and test accuracies exhibit high performance and low variability across the grid.

E-4 Example test images from Section 6.5 $x^\star$ (upper) and their warps $g(x^\star)$ (lower) and predicted class labels (above and below).

E-5 Extra hyperparameter uncertainty histograms for our synthetic extrapolation example in Section 6.2.3 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.

E-6 Extra hyperparameter uncertainty histograms for our synthetic interpolation example in Section 6.2.3 in which we find do not find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.

E-7 Extra hyperparameter uncertainty histograms for our heart rate experiment in Section 6.3 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.

E-8 Extra hyperparameter uncertainty histograms for our Mauna Loa experiment in Section 6.4 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.

E-9 Extra hyperparameter uncertainty histograms for our MNIST experiment in Section 6.5 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances. Note, distances are plotted on the log-scale.

E-10 Extra hyperparameter uncertainty histograms for our additional heart rate experiment in Appendix E.3 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1</td>
<td>Evaluation of approximate LWCV for time-varying Poisson processes. Mean ACV relative error, $</td>
</tr>
<tr>
<td>C.2</td>
<td>Wall clock time from the two lower right panels in Fig. 4-1 at $T = 50000$ and with $m% = 10%$ of the data in the held out fold.</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Machine learning and statistical methods are increasingly used in high-stakes applications – for instance, in atmospheric modeling or providing medical care. We want evidence that our methods will be accurate, though, before we use them in such applications. Cross-validation (CV) [Geisser 1975, Stone 1974, Allen 1974] is often used as the gold standard for evaluating a method's accuracy on tasks similar to the training data. For example, given a method for predicting hospital patients' heart rates, CV might be used to measure accuracy for predicting heart rates for patients similar to patients in the training data. A long history of empirical and theoretical studies validates the use of CV in such settings [Kohavi 1995, Arlot and Celisse 2010, Patil et al. 2021]. However, in practice, users often want to solve tasks in settings that are dissimilar from, but related to, the setting where training data is collected; e.g. one may want to predict the heart rate of a novel patient or the future heart rate of an existing patient. A method can perform well in CV and still extrapolate poorly to such future tasks if we have unwittingly encoded incorrect assumptions into the method and the method is highly dependent on these assumptions. Robustness checks examine this dependency, and hence provide a complementary form of assessment to CV. In both cases (CV and robustness checks), a typical workflow follows the pattern of “change the dataset or method, and then rerun the method.” For instance, in the context of Bayesian methods, prior robustness checks often repeat inference for a few different user-selected priors.

However, for methods that use complex models trained on large datasets, tools like CV and robustness checks do not see widespread use – even though arguably these are the cases that need these checks the most. Why do these tools see little use? First, especially in these challenging model and data settings, it is often computationally expensive to repeatedly refit a model. Second, for robustness checks, it may be expensive in terms of user time to determine what the relevant set of model refits is; e.g. in the case of Bayesian prior robustness, it may take considerable time and effort for a user to specify what a reasonable set of alternative priors is. In this thesis, we address these challenges by developing model evaluation tools that are fast in terms of both compute and user time. In particular, we answer the questions:

1. Can we quickly and accurately approximate CV when we need it most? Existing work [Beirami et al. 2017, Giordano et al. 2019b, Rad and
Maleki [2020] addresses the computational expense of CV by providing fast and accurate approximations to CV. However, we observe that these approximations are both inaccurate and slow for high-dimensional models – i.e. when the number of observed datapoints $N$ is not large compared to the dimension of the model parameter $D$. This is especially concerning in the context of CV, as if $N \gg D$, the training error provides a reliable estimate of the out-of-sample performance of a model. That is, in low-dimensional problems, there is little need for CV. In Chapters 2 and 3, we identify conditions under which CV can be approximated quickly and accurately despite the presence of high dimensions. In particular, we note that a broad theme in high-dimensional statistics is to show how hidden low-dimensional structure in a problem can ameliorate issues due to high dimensions. We show the same is true for approximating CV in two cases that are common in applications: generalized linear models with sparsity and generalized linear models with low rank data.

In Chapter 2 we handle the case of approximating CV for high-dimensional generalized linear models with sparsity. We compare a number of pre-existing proposals in the literature for handling this case and show that only one is both empirically fast and accurate. This one successful proposal assumes that the sparse structure recovered under the full dataset is stable as each portion of the dataset is left out by CV. This reduces the problem to a lower dimensional space, automatically guaranteeing it will be fast. We show that this proposal has good theoretical guarantees on its accuracy if its core assumption is correct – that is, if the sparse structure is stable as each portion of the dataset is left out by CV. We prove conditions under which this stability does in fact hold and show in experiments that these conditions seem necessary to guarantee the high accuracy and speed of approximate CV methods in high-dimensional sparse settings.

In Chapter 3 we handle the case of approximating CV for high-dimensional generalized linear models with low-rank covariates. In this case, we prove that pre-existing approximate CV methods are automatically accurate despite the presence of high dimensions. However, existing methods require the inversion of a $D \times D$ matrix, where $D$ is the dimension of the model parameter. In high dimensions, when $D$ is large, this matrix inverse will be computationally prohibitive. We propose to approximate this matrix inverse using a new low-rank randomized matrix decomposition customized to the needs of approximate CV. We prove that this decomposition is somewhat counterintuitive but effective: while our low-rank decomposition is the worst possible low-rank approximation to our inverse matrix of interest in terms of the spectral norm, we show that it is in a sense the best possible approximation for the needs of approximate CV.

2. How broad a class of models can we approximate CV for? The approximations to CV discussed above focus on “non-structured” forms of CV: that is, models for which the data dropped out by each fold of CV are statistically independent of the rest of the data under the model. While this is true for
many models of interest, a broad class of models does not satisfy this assumption. For example, in a model with temporal structure, one may wish to drop out datapoints at the end of an observed time series, refit the model, and then predict the left-out datapoints as a measure of the model’s ability to predict future observations. A broad class of structured models have latent structure following a Markov random field (MRF). Examples include time series models, language models, and geospatial models. In Chapter 4, we show that structured forms of CV for such models (e.g. dropping out a site in a geospatial model) can be approximated quickly and accurately. A challenge along the way is that the log likelihood of such models is defined by marginalization over the latent MRF structure. Previous methods for approximating CV rely on differentiating the log likelihood; that marginalization of the latent MRF structure can be efficiently differentiated is one of the main insights behind the work in this chapter. We show that our algorithm can quickly and accurately evaluate a temporal model of highway traffic, a sequential language model, and a spatial model of crime in Philadelphia.

3. **Is cross-validation loss as easy to optimize as it seems?** A common use of CV is to select model hyperparameters by minimizing the CV loss. For widely used, simple models like ridge regression, a growing body of work validates this practice by showing that the global optimum of the CV loss has desirable theoretical properties [Hastie et al., 2020]. Practitioners typically use grid search or gradient descent to minimize this loss; however, these methods may struggle in the presence of local optima. So, to realize the promise of theory in practice, we need to understand whether the CV loss can have local optima.

In Chapter 5, we show the CV loss may have (non-global) local optima, even for simple models like ridge regression. In particular, we focus on the leave-one-out CV (LOOCV) loss for tuning the ridge parameter in ridge regression (i.e. $\ell_2$-regularized linear regression) and show that even this simple CV loss may have local optima. The presence of local optima is problematic as grid search and gradient descent – two of the most commonly used methods to optimize CV losses – can struggle to quickly find the global optimum of functions with local minima. We show that the presence of local optima is in general a complex issue, but we also prove conditions under which the LOOCV loss is guaranteed to have no local optima: when the singular values of the covariate matrix are sufficiently uniform.

4. **How sensitive are our models to arbitrary assumptions?** CV measures the accuracy of a statistical or machine learning model on tasks similar to the setting in which the training data is collected. However, in many cases, users want to solve tasks in settings that are dissimilar from, but related to, the setting where training data is collected. For example, given past estimates of carbon dioxide levels, will the carbon dioxide levels in five years be concerningly high? No matter how much historical data we observe, the answer to this question will always depend on the specified model. Some parts of a given model will
represent concrete scientific knowledge about the data, while some parts of the model will be arbitrary. How sensitive is our analysis to the arbitrary parts of our model?

In Chapter 6, we answer this question for Gaussian processes (GPs) [Rasmussen and Williams, 2006], which model unknown functions. To use a GP, one has to specify the kernel function $k$. Specification of $k$ often encodes certain beliefs about the underlying data. For example, a user may specify a squared exponential kernel because they believe the unknown function is smooth and has a certain length scale. However, there are almost certainly alternative kernels that equally well express these prior beliefs; we call such kernels qualitatively interchangeable. We study how sensitive decisions are when $k$ varies among all qualitatively interchangeable kernels. To do so, we formulate the search over alternative kernels as a finite dimensional optimization problem in order to find a kernel under which the posterior is substantially different than under the user-specified kernel. We then introduce visual tests to determine if the resulting posterior-changing kernel is qualitatively interchangeable with the user-specified kernel. We find substantial sensitivity in a variety of applications using GPs, including heart rate prediction in hospitals, atmospheric carbon dioxide level forecasting, and prediction of handwritten digits.

Chapter 2 contains material from [Stephenson and Broderick, 2020] completed in collaboration with Tamara Broderick. Chapter 3 contains material from [Stephenson et al., 2020] completed in collaboration with Madeleine Udell and Tamara Broderick. Chapter 4 contains material from Ghosh* and Stephenson* et al. [2020] (*) denotes equal contribution) completed in collaboration with Soumya Ghosh, Tin Nguyen, Sameer Deshpande and Tamara Broderick. Chapter 5 contains material from [Stephenson et al., 2021a] completed in collaboration with Zachary Frangella, Madeleine Udell, and Tamara Broderick. Chapter 6 contains material from [Stephenson et al., 2021b] completed in collaboration with Soumya Ghosh, Tin Nguyen, Mikhail Yurochkin, Sameer Deshpande, and Tamara Broderick.
Chapter 2

Approximate cross-validation for sparse models in high dimensions

2.1 Introduction

Assessing the performance of machine learning methods is an important task in medicine, genomics, and other applied fields. Experts in these areas are interested in understanding methods’ error or variability and, for these purposes, often turn to cross validation (CV); see, e.g., Saeb et al. [2017], Powers et al. [2019], Carrera et al. [2009], Joshi et al. [2009], Chandrasekaran et al. [2011], Biswal et al. [2001], Roff and Preziosi [1994]. Even after decades of use [Stone, 1974, Geisser, 1975], CV remains relevant in modern high-dimensional and complex problems. In these cases, CV provides, for example, better out-of-sample error estimates than simple test error or training error [Stone, 1974]. Moreover, among variants of CV, leave-one-out CV (LOOCV) offers to most closely capture performance on the dataset size of interest. For instance, LOOCV is particularly accurate for out-of-sample error estimation [Arlot and Celisse, 2010, Sec. 5].

Modern datasets, though, pose computational challenges for CV. For instance, CV requires running a machine learning algorithm many times, especially in the case of LOOCV. This expense has led to recent proposals to approximate LOOCV [Obuchi and Kabashima, 2016, 2018, Beirami et al., 2017, Rad and Maleki, 2020, Wang et al., 2018, Giordano et al., 2019b, Xu et al., 2021]. Theory and empirics demonstrate that these approximations are fast and accurate – as long as the dimension $D$ of the unknown parameter in a problem is low. Unfortunately a number of issues arise in high dimensions, the exact case of modern interest. First, existing error bounds for LOOCV approximations either assume a fixed $D$ or suffer from poor error scaling when $D$ grows with $N$. One might wonder whether the theory could be improved, but our own experiments (see, e.g., Fig. [2-1]) confirm that LOOCV approximations can suffer considerable error degradation in high dimensions in practice. Second, even if the approximations were accurate in high dimensions, these approximations require

\footnote{In the case of linear regression, LOOCV provides the least biased and lowest variance estimate of out-of-sample error among other CV methods [Burman, 1989].}
solving a $D$-dimensional linear system, which incurs an $O(D^3)$ cost.

Previous authors have proposed a number of potential solutions for one or both of these problems, but these methods have not yet been carefully evaluated and compared. (#1) Koh and Liang [2017] use a randomized solver [Agarwal et al., 2017] successfully for qualitative analyses similar to high-dimensional approximate CV, so it is natural to think the same technique might speed up approximate CV in high dimensions. Another option is to consider that the unknown parameter may effectively exist in some subspace with much lower dimension than $D$. For instance, $\ell_1$ regularization offers an effective and popular means to recover a sparse parameter support. Since existing approximate CV methods require twice differentiability of the regularizer, they cannot be applied directly with an $\ell_1$ penalty. (#2) Thus, a second proposal – due to Rad and Maleki [2020], Wang et al. [2018] – is to apply existing approximate CV methods to a smoothed version of the $\ell_1$ regularizer. (#3) A third proposal – made by, e.g., Burman [1989] – is to ignore modern approximate CV methods, and speed up CV by uniform random subsampling of LOOCV folds.

We show that all three of these methods fail to address the issues of approximate CV in high dimensions. (#4) A fourth proposal – due to Rad and Maleki [2020], Wang et al. [2018], Obuchi and Kabashima [2016, 2018], Beirami et al. [2017] – is to again consider $\ell_1$ regularization for sparsity. But in this case, the plan is to fit the model once with the full dataset to find a sparse parameter subspace and then apply existing approximate CV methods to only this small subspace.

In what follows, we demonstrate with both empirics and theory that proposal #4 is the only method that is fast and accurate for assessing out-of-sample error. We emphasize, moreover, its simplicity and ease of implementation. On the theory side, we show in Section 2.4 that proposal #4 will work if exact LOOCV rounds recover a shared support. Our major theoretical contribution is to prove that, under mild and interpretable conditions, the recovered support is in fact shared across rounds of LOOCV with very high probability (Sections 2.4.1 and 2.4.2). Obuchi and Kabashima [2016] have considered a similar setup and shown that the effect of the change in support is asymptotically negligible for $\ell_1$-regularized linear regression; however, they do not show the support is actually shared. Additionally, Beirami et al. [2017], Obuchi and Kabashima [2018] make the same approximation in the context of other GLMs but without theoretical justification. We justify such approximations by proving that the support is shared with high probability in the practical finite-data setting – even for the very high-dimensional case $D = o(e^N)$ – for both linear and logistic regression (Theorems 2 and 3). Our support stability result may be of independent interest and allows us to show that, with high probability under finite data, the error and time cost of proposal #4 will depend on the support size – typically much smaller than the full dimension – rather than $D$. Our experiments in Section 2.5 on real and simulated data confirm these theoretical results.

**Model assessment vs. selection.** Stone [1974], Geisser [1975] distinguish at

---

Note that sparsity, induced by $\ell_1$ regularization, is typically paired with a focus on generalized linear models (GLMs) since these models simplify when many parameters are set to zero, are tractable to analyze with theory, and typically form the building blocks for even more complex models.
least two uses of CV: model assessment and model selection. Model assessment refers to estimating the performance of a single, fixed model. Model selection refers to choosing among a collection of competing models. We focus almost entirely on model assessment – for two principal reasons. First, as discussed above, CV is widely used for model assessment in critical applied areas – such as medicine and genetics. Before we can safely apply approximate CV for model assessment in these areas, we need to empirically and theoretically verify our methods. Second, historically, rigorous analysis of the properties of model selection even for exact CV has required significant additional work beyond analyzing CV for model assessment. In fact, exact CV for model selection has only recently begun to be theoretically understood for $\ell_1$ regularized linear regression. Our experiments in Appendix A.8 confirm that approximate CV for model selection exhibits complex behavior. We thus expect significant further work, outside the scope of the present thesis, to be necessary to develop a theoretical understanding of approximate CV for model selection. Indeed, to the best of our knowledge, all existing theory for the accuracy of approximate CV applies only to model assessment.

### 2.2 Overview of Approximations

Let $\theta \in \Theta \subseteq \mathbb{R}^D$ be an unknown parameter of interest. Consider a dataset of size $N$, where $n \in [N] := \{1, 2, \ldots, N\}$ indexes the data point. Then a number of problems – such as maximum likelihood, general M-estimation, and regularized loss minimization
can be expressed as solving

\[ \hat{\theta} := \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} f_n(\theta) + \lambda R(\theta), \tag{2.1} \]

where \( \lambda \geq 0 \) is a constant, and \( R : \Theta \to \mathbb{R}_+ \) and \( f_n : \Theta \to \mathbb{R} \) are functions. For instance, \( f_n \) might be the loss associated with the \( n \)th data point, \( R \) the regularizer, and \( \lambda \) the amount of regularization. Consider a dataset where the \( n \)th data point has covariates \( x_n \in \mathbb{R}^D \) and response \( y_n \in \mathbb{R} \). In what follows, we will be interested in taking advantage of sparsity. With this in mind, we focus on generalized linear models (GLMs), where \( f_n(\theta) = f(x_n^T \theta, y_n) \), as they offer a natural framework where sparsity can be expressed by choosing many parameter dimensions to be zero.

In LOOCV, we are interested in solutions of the same problem with the \( n \)th data point removed. To that end, define \( \hat{\theta} \setminus n := \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{m: m \neq n} f_m(\theta) + \lambda R(\theta) \). Computing \( \hat{\theta} \setminus n \) exactly across \( n \) usually requires \( N \) runs of an optimization procedure – a prohibitive cost. Various approximations, detailed next, address this cost by solving Eq. (2.1) only once.

**Two approximations.** Assume that \( f \) and \( R \) are twice differentiable functions of \( \theta \). Let \( F(\theta) := (1/N) \sum_n f(x_n^T \theta, y_n) \) be the unregularized objective, and let \( H(\theta) := \nabla^2 \theta F(\theta) + \lambda \nabla^2 \theta R(\theta) \) be the Hessian matrix of the full objective. For the moment, we assume appropriate terms in each approximation below are invertible. Beirami et al. [2017], Rad and Maleki [2020], Wang et al. [2018], Koh et al. [2019] approximate \( \hat{\theta} \setminus n \) by taking a Newton step (“NS”) on the objective \( (1/N) \sum_{m: m \neq n} f_m + \lambda R \) starting from \( \hat{\theta} \); see Appendix A.4.4 for details. We thus call this approximation \( \tilde{\text{NS}} \setminus n(R) \) for regularizer \( R \):

\[ \tilde{\text{NS}} \setminus n(R) := \hat{\theta} + \frac{1}{N} \left( H(\hat{\theta}) - \frac{1}{N} \nabla^2 \theta f_n(\hat{\theta}) \right)^{-1} \nabla \theta f_n(\hat{\theta}). \tag{2.2} \]

In the case of GLMs, Theorem 8 of Rad and Maleki [2020] gives conditions on \( x_n \) and \( f(\cdot, \cdot) \) that imply, for fixed \( D \), the error of \( \tilde{\text{NS}} \setminus n(R) \) averaged over \( n \) is \( o(1/N) \) as \( N \to \infty. \)

Koh and Liang [2017], Beirami et al. [2017], Giordano et al. [2019b], Koh et al. [2019] consider a second approximation. As their approximation is inspired by the *infinitesimal jackknife* (“IJ”) [Jaeckel, 1972, Efron, 1982], we denote it by \( \tilde{\text{IJ}} \setminus n(R) \); see Appendix A.4.1.

\[ \tilde{\text{IJ}} \setminus n(R) := \hat{\theta} + \frac{1}{N} H(\hat{\theta})^{-1} \nabla \theta f_n(\hat{\theta}). \tag{2.3} \]

Giordano et al. [2019b] study the case of \( \lambda = 0 \), and, in their Corollary 1, show that

---

3See Appendix A.1 for a brief review of CV methods.

4Note our choice of \( 1/N \) scaling here – instead of \( 1/(N-1) \). While we believe this choice is not of particular importance in the case of LOOCV, this issue does not seem to be settled in the literature; see Appendix A.2.
the accuracy of Eq. (2.3) is bounded by $C/N$ in general or, in the case of bounded gradients $\|{\nabla}_{\theta} f(x^n_{n}\theta, y_n)\|_{\infty} \leq B$, by $C' B/N^2$. The constants $C, C'$ may depend on $D$ but not $N$. Our Proposition 10 in Appendix A.4.3 extends this result to the regularized case, $\lambda \geq 0$. Still, we are left with the fact that $C$ and $C'$ depend on $D$ in an unknown way.

In what follows, we consider both $\tilde{\mathcal{NS}}_{\lambda n}(R)$ and $\tilde{\mathcal{I}}_{\lambda n}(R)$, as they have complementary strengths. Empirically, we find that $\tilde{\mathcal{NS}}_{\lambda n}(R)$ performs better in our LOOCV GLM experiments. But $\tilde{\mathcal{I}}_{\lambda n}(R)$ is computationally efficient beyond LOOCV and GLMs. E.g., for general models, computation of $\tilde{\mathcal{NS}}_{\lambda n}(R)$ requires inversion of a new Hessian for each $n$, whereas $\tilde{\mathcal{I}}_{\lambda n}(R)$ needs only the inversion of $H(\hat{\theta})$ for all $n$. In terms of theory, $\tilde{\mathcal{NS}}_{\lambda n}(R)$ has a tighter error bound of $o(1/N)$ for GLMs. But the theory behind $\tilde{\mathcal{I}}_{\lambda n}(R)$ applies more generally, and, given a good bound on the gradients, may provide a tighter rate.

### 2.3 Problems in high dimensions

In the above discussion, we noted that there exists encouraging theory governing the behavior of $\tilde{\mathcal{NS}}_{\lambda n}(R)$ and $\tilde{\mathcal{I}}_{\lambda n}(R)$ when $D$ is fixed and $N$ grows large. We now describe issues with $\tilde{\mathcal{NS}}_{\lambda n}(R)$ and $\tilde{\mathcal{I}}_{\lambda n}(R)$ when $D$ is large relative to $N$. The first challenge for both approximations given large $D$ is computational. Since every variant of CV or approximate CV requires running the machine learning algorithm of interest at least once, we will focus on the cost of the approximations after this single run. Given $\hat{\theta}$, both approximations require the inversion of a $D \times D$ matrix. Calculation of $\tilde{\mathcal{I}}_{\lambda n}(R)$ across $n \in [N]$ requires a single matrix inversion and $N$ matrix multiplications for a runtime in $O(D^3 + ND^2)$. In general, calculating $\tilde{\mathcal{NS}}_{\lambda n}(R)$ has runtime of $O(ND^3)$ due to needing an inversion for each $n$. In the case of GLMs, though, $\nabla^2 f_{n}$ is a rank-one matrix, so standard rank-one updates give a runtime of $O(D^3 + ND^2)$ as well.

The second challenge for both approximations is the invertibility of $H(\hat{\theta})$ and $H(\hat{\theta}) - (1/N)\nabla^2 f(x^n_{n}\theta, y_n)$ that was assumed in defining $\tilde{\mathcal{NS}}_{\lambda n}(R)$ and $\tilde{\mathcal{I}}_{\lambda n}(R)$. We note that, if $\nabla^2 R(\hat{\theta})$ is only positive semidefinite, then invertibility of both matrices may be impossible when $D \geq N$; see Appendix A.4.2 for more discussion.

The third and final challenge for both approximations is accuracy in high dimensions. Not only do existing error bounds behave poorly (or not exist) in high dimensions, but empirical performance degrades as well. To create Fig. 2-1, we generated datasets from a sparse logistic regression model with $N$ ranging from 500 to 5,000. For the blue lines, we set $D = 2$, and for the red lines we set $D = N/10$. In both cases, we see that error is much lower when $D$ is small and fixed.

We recall that for large $N$ and small $D$, training error often provides a fine estimate of the out-of-sample error (e.g., see [Vapnik, 1992]). That is, CV is needed precisely in the high-dimensional regime, and this case is exactly where current approximations struggle both computationally and statistically. Thus, we wish to understand whether there are high-$D$ cases where approximate CV is useful. In what follows, we consider
a number of options for tackling one or more of these issues and show that only one method is effective in high dimensions.

**Proposal #1:** Use randomized solvers to reduce computation. Previously, Koh and Liang [2017] have utilized $\tilde{I}_n(R)$ for qualitative purposes, in which they are interested in its sign and relative magnitude across different $n$. They tackle the $O(D^3)$ scaling of $\tilde{I}_n(R)$ by using the randomized solver from Agarwal et al. [2017]. While one might hope to replicate the success of Koh and Liang [2017] in the context of approximate CV, we show in Appendix A.3 that this randomized solver performs poorly for approximating CV: while it can be faster than exactly solving the needed linear systems, it provides an approximation to exact CV that can be an order of magnitude less accurate.

2.3.1 **Sparsity via $\ell_1$ regularization.**

Intuitively, if the exact $\hat{\theta}_n$’s have some low “effective dimension” $D_{\text{eff}} \ll D$, we might expect approximate CV’s accuracy to depend only on $D_{\text{eff}}$. One way to achieve low $D_{\text{eff}}$ is sparsity: i.e., we have $\hat{D}_{\text{eff}} := |\text{supp} \hat{\theta}| \ll D$, where $\hat{S} := \text{supp} \hat{\theta}$ collects the indices of the non-zero entries of $\hat{\theta}$. A way to achieve sparsity is choosing $R(\theta) = \|\theta\|_1$.

However, note that $\tilde{N}_n(R)$ and $\tilde{I}_n(R)$ cannot be applied directly in this case as $\|\theta\|_1$ is not twice-differentiable. **Proposal #2:** Rad and Maleki [2020], Wang et al. [2018] propose the use of a smoothed approximation to $\|\cdot\|_1$; however, as we show in Section 2.5, this approach is often multiple orders of magnitude more inaccurate and slower than Proposal #4 below.

**Proposal #3:** Subsample exact CV. Another option is to bypass all the problems of approximate CV in high-D by uniformly subsampling a small collection of LOOCV folds. This provides an unbiased estimate of exact CV and can be used with exact $\ell_1$ regularization. However, our experiments (Section 2.5) show that, under a time budget, the results of this method are so variable that their error is often multiple orders of magnitude higher than Proposal #4 below.

**Proposal #4:** Use the sparsity from $\hat{\theta}$. Instead, in what follows, we take the intuitive approach of approximating CV only on the dimensions in $\text{supp} \hat{\theta}$. Unlike all previously discussed options, we show that this approximation is fast and accurate in high dimensions in both theory and practice. For notation, let $X \in \mathbb{R}^{N \times D}$ be the covariate matrix, with rows $x_n$. For $S \subset [D]$, let $X_{\cdot S}$ be the submatrix of $X$ with column indices in $S$; define $x_{nS}$ and $\theta_S$ similarly. Let $\hat{D}_{\text{eff}}^{(2)} := [d^2 f(z, y_n)/dz^2]_{z=x_n^T \hat{\theta}}$, and define the restricted Hessian evaluated at $\hat{\theta}$: $H_{\hat{\theta}} := X_{\hat{S}}^T \text{diag}(\hat{D}_{\text{eff}}^{(2)}) X_{\cdot \hat{S}}$. Further define the LOO restricted Hessian, $H_{\hat{\theta}}^{\text{LOO}} := H_{\hat{\theta}} - [\nabla^2 f(x_n^T \hat{\theta}, y_n)]_{\hat{S} \hat{S}}$. Finally, without loss of generality, assume $\hat{S} = \{1, 2, \ldots, \hat{D}_{\text{eff}}\}$. We now define versions of $\tilde{N}_n(R)$...
Recall that the accuracy of \( \hat{\theta} \) restricted to the entries in supp \( \hat{\theta} \):

\[
\text{NS}_{\hat{\theta}} := \left( \hat{\theta}_S + (H_{SS}^{-1})_{ij} \begin{bmatrix} \nabla \theta f(x_n^T \hat{\theta}, y_n) \\ 0 \end{bmatrix} \right)
\]

(2.4)

\[
\text{IJ}_{\hat{\theta}} := \left( \hat{\theta}_S + H_{SS}^{-1} \begin{bmatrix} \nabla \theta f(x_n^T \hat{\theta}, y_n) \\ 0 \end{bmatrix} \right)
\]

(2.5)

Other authors have previously considered \( \text{NS}_{\hat{\theta}} \). Beirami et al. [2017] directly propose \( \text{NS}_{\hat{\theta}} \) by considering a smooth approximation to \( \ell_1 \) and then taking the limit of \( \tilde{\text{NS}}_{\hat{\theta}}(R) \) as the amount of smoothness goes to zero. In Appendix A.5, we show a similar argument can yield \( \text{IJ}_{\hat{\theta}} \). Also, Obuchi and Kabashima [2016, 2018], Rad and Maleki [2020], Wang et al. [2018] derive \( \text{NS}_{\hat{\theta}} \) by considering a smooth approximation to \( \ell_1 \) and then taking the limit of \( \tilde{\text{NS}}_{\hat{\theta}}(R) \) as the amount of smoothness goes to zero. In Appendix A.5, we show a similar argument can yield \( \text{IJ}_{\hat{\theta}} \). Also, Obuchi and Kabashima [2016, 2018], Beirami et al. [2017] directly propose \( \text{NS}_{\hat{\theta}} \) without using \( \text{NS}_{\hat{\theta}}(R) \) as a starting point. We now show how \( \text{NS}_{\hat{\theta}} \) and \( \text{IJ}_{\hat{\theta}} \) avoid the three major high-dimensional challenges with \( \tilde{\text{NS}}_{\hat{\theta}}(R) \) and \( \tilde{\text{IJ}}_{\hat{\theta}}(R) \) we discussed above.

The first challenge was that compute time for \( \tilde{\text{NS}}_{\hat{\theta}}(R) \) and \( \tilde{\text{IJ}}_{\hat{\theta}}(R) \) scaled poorly with \( D \). That \( \text{NS}_{\hat{\theta}} \) and \( \text{IJ}_{\hat{\theta}} \) do not share this issue is immediate from their definitions.

**Proposition 1.** For general \( f_n \), the time to compute \( \text{NS}_{\hat{\theta}} \) or \( \text{IJ}_{\hat{\theta}} \) scales with \( \hat{D}_\text{eff} \), rather than \( D \). In particular, computing \( \text{NS}_{\hat{\theta}} \) across all \( n \in [N] \) takes \( O(N \hat{D}_\text{eff}^3) \) time, and computing \( \text{IJ}_{\hat{\theta}} \) across all \( n \in [N] \) takes \( O(\hat{D}_\text{eff}^3 + N \hat{D}_\text{eff}^2) \) time. Furthermore, when \( f_n \) takes the form of a GLM, computing \( \text{NS}_{\hat{\theta}} \) across all \( n \in [N] \) takes \( O(\hat{D}_\text{eff}^3 + N \hat{D}_\text{eff}^2) \) time.

The second high-dimensional challenge was that \( H \) and \( H^{\hat{n}} \) may not be invertible when \( D \geq N \). Notice the relevant matrices in \( \text{NS}_{\hat{\theta}} \) and \( \text{IJ}_{\hat{\theta}} \) are of dimension \( \hat{D}_\text{eff} = |\hat{S}| \). So we need only make the much less restrictive assumption that \( \hat{D}_\text{eff} < N \), rather than \( D < N \). We address the third and final challenge of accuracy in the next section.

### 2.4 Approximation quality in high dimensions

Recall that the accuracy of \( \tilde{\text{NS}}_{\hat{\theta}}(R) \) and \( \tilde{\text{IJ}}_{\hat{\theta}}(R) \) in general has a poor dependence on dimension \( D \). We now show that the accuracy of \( \text{NS}_{\hat{\theta}} \) and \( \text{IJ}_{\hat{\theta}} \) depends on (the hopefully small) \( \hat{D}_\text{eff} \) rather than \( D \). We start by assuming a “true” population parameter\(^5\) \( \theta^* \in \mathbb{R}^D \) that minimizes the population-level loss, \( \theta^* := \arg \min_x \mathbb{E}_{x,y}[f(x^T \theta, y)] \), where the expectation is over \( x, y \) from some population distribution. Assume \( \theta^* \) is sparse with \( S := \text{supp} \theta^* \) and \( D_\text{eff} := |S| \). Our parameter estimate would be faster and more accurate if an oracle told us \( S \) in advance and we worked just over \( S \):

\[
\hat{\phi} := \arg \min_{\phi \in \mathbb{R}^{D_\text{eff}}} \frac{1}{N} \sum_{n=1}^{N} f(x_n^T \phi, y_n) + \lambda \|\phi\|_1 .
\]

(2.6)

\(^5\)This assumption may not be necessary to prove the dependence of \( \text{NS}_{\hat{\theta}} \) and \( \text{IJ}_{\hat{\theta}} \) on \( \hat{D}_\text{eff} \), but it allows us to invoke existing \( \ell_1 \) support results in our proofs.
We define $\hat{\phi}_n$ as the leave-one-out variant of $\hat{\phi}$ (as $\hat{\theta}_n$ is to $\hat{\theta}$). Let $\text{RNS}_n$ and $\text{RIJ}_n$ be the result of applying the approximation in $\text{NS}_n$ or $\text{IJ}_n$ to the restricted problem in Eq. (2.6); note that $\text{RNS}_n$ and $\text{RIJ}_n$ have accuracy that scales with the (small) dimension $D_{\text{eff}}$.

Our analysis of the accuracy of $\text{NS}_n$ and $\text{IJ}_n$ will depend on the idea that if, for all $n$, $\text{NS}_n$, $\text{IJ}_n$, and $\hat{\theta}_n$ run over the same $D_{\text{eff}}$-dimensional subspace, then the accuracy of $\text{NS}_n$ and $\text{IJ}_n$ must be identical to that of $\text{RNS}_n$ and $\text{RIJ}_n$. In the case of $\ell_1$ regularization, this idea specializes to the following condition, under which our main result in Theorem 1 will be immediate.

**Condition 1.** For all $n \in [N]$, we have $	ext{supp } \text{IJ}_n = \text{supp } \text{NS}_n = \text{supp } \hat{\theta}_n = S$.

**Theorem 1.** Assume Condition 1 holds. Then for all $n$, $\hat{\theta}_n$ and $\text{IJ}_n$ are (1) zero outside the dimensions $S$ and (2) equal to their restricted counterparts from Eq. (2.6):

$$\hat{\theta}_n = \begin{pmatrix} \hat{\theta}_n[S] \\ 0 \end{pmatrix} = \begin{pmatrix} \hat{\phi}_n \\ 0 \end{pmatrix},$$

$$\text{IJ}_n = \begin{pmatrix} \text{IJ}_n[S] \\ 0 \end{pmatrix} = \begin{pmatrix} \text{RIJ}_n \\ 0 \end{pmatrix}. \quad (2.7)$$

It follows that the error is the same in the full problem as in the low-dimensional restricted problem: $\|\hat{\theta}_n - \text{IJ}_n\|_2 = \|\hat{\phi}_n - \text{RIJ}_n\|_2$. The same results hold for $\text{IJ}_n$ and $\text{RIJ}_n$ replaced by $\text{NS}_n$ and $\text{RNS}_n$.

Taking Condition 1 as a given, Theorem 1 tells us that for $\ell_1$ regularized problems, $\text{IJ}_n$ and $\text{NS}_n$ inherit the fixed-dimensional accuracy of $\tilde{\text{IJ}}_n(R)$ and $\tilde{\text{NS}}_n(R)$ shown empirically in Fig. 2-1 and described theoretically in the references from Section 2.1.

Taking a step further, one could show that $\text{IJ}_n$ and $\text{NS}_n$ are accurate for model assessment tasks by using results on the accuracy of exact CV for assessment (e.g., [Abou-Moustafa and Szepesvári, 2018, Steinberger and Leeb, 2018, Barber et al., 2021]).

Again, Theorem 1 is immediate if one is willing to assume Condition 1 but when does Condition 1 hold? There exist assumptions in the $\ell_1$ literature under which $\text{supp } \hat{\theta} = S$ [Lee et al., 2014, Li et al., 2015]. If one took these assumptions to hold for all $F^n := (1/N) \sum_{m:m \neq n} f_m$, then Condition 1 would directly follow. However, it is not immediate that any models of interest meet such assumptions. Rather than taking such uninterpretable assumptions or just taking Condition 1 as an assumption directly, we will give a set of more interpretable assumptions under which Condition 1 holds.

In fact, we need just four principal assumptions in the case of linear and logistic regression; we conjecture that similar results hold for other GLMs. The first assumption arises from the intuition that, if individual data points are very extreme, the support will certainly change for some $n$. To avoid these extremes with high probability, we assume that the covariates follow a sub-Gaussian distribution:

**Definition 1.** [e.g., Vershynin, 2018] For $c_x > 0$, a random variable $V$ is $c_x$-sub-Gaussian if $\mathbb{E}[\exp(V^2/c_x^2)] \leq 2$. 

29
**Assumption 1.** Each $x_n \in \mathbb{R}^D$ has zero-mean i.i.d. $c_x$-sub-Gaussian entries with $\mathbb{E}[x_{nd}^2] = 1$.

We conjecture that the unit-variance part of the assumption is unnecessary. Conditions on the distributions of the responses $y_n$ will be specific to linear and logistic regression and will be given in Assumptions 5 and 6, respectively. Our results below will hold with high probability under these distributions. Note there are reasons to expect we cannot do better than high-probability results. In particular, Xu et al. [2012] show that there exist worst-case training datasets for which sparsity-inducing methods like $\ell_1$ regularization are not stable as each datapoint is left out.

Our second principal assumption is an incoherence condition.

**Assumption 2.** The incoherence condition holds with high probability over the full dataset:

$$\Pr \left[ \left\| \nabla F(\theta^*)_{S_c,S} \left( \nabla^2 F(\theta^*)_{SS} \right)^{-1} \right\|_\infty < 1 - \alpha \right] \leq e^{-25},$$

Authors in the $\ell_1$ literature often assume that incoherence holds deterministically for a given design matrix $X$ – starting from the introduction of incoherence by Zhao and Yu [2006] and continuing in more recent work [Lee et al., 2014, Li et al., 2015]. Similarly, we will take our high probability version in Assumption 2 as given. But we note that Assumption 2 is at least known to hold for the case of linear regression with an i.i.d. Gaussian design matrix (e.g., see Exercise 11.5 of [Hastie et al., 2015]).

We next place some restrictions on how quickly $D$ and $D_{\text{eff}}$ grow as functions of $N$.

**Assumption 3.** As functions of $N$, $D$ and $D_{\text{eff}}$ satisfy: (1) $D = o(e^N)$, (2) $D_{\text{eff}} = o([N/ \log N]^{2/5})$, and (3) $D_{\text{eff}}^{3/2} \sqrt{\log D} = o(N)$.

The constraints on $D$ here are particularly loose. While those on $D_{\text{eff}}$ are tighter, we still allow polynomial growth of $D_{\text{eff}}$ in $N$ for some lower powers of $N$. Our final assumption is on the smallest entry of $\theta^*_S$. Such conditions – typically called beta-min conditions – are frequently used in the $\ell_1$ literature to ensure $\hat{S} = S$ [Wainwright, 2009, Lee et al., 2014, Li et al., 2015].

**Assumption 4.** $\theta^*$ satisfies $\min_{s \in S} |\theta^*_s| > \sqrt{D_{\text{eff}} T_{\text{min}} \lambda}$, where $T_{\text{min}}$ is some constant relating to the objective function $f$; see Assumption 23 in Appendix A.9.1 for an exact description.

### 2.4.1 Linear regression

We now give the distributional assumption on the responses $y_n$ in the case of linear regression and then show that Condition 1 holds.

**Assumption 5.** $\forall n, y_n = x_n^T \theta^* + \varepsilon_n$, where the $\varepsilon_n$ are i.i.d. $c_\varepsilon$-sub-Gaussian random variables.
Theorem 2 (Linear Regression). Take Assumptions 1 to 5. Suppose the regularization parameter $\lambda$ satisfies

$$\lambda \geq \frac{C}{\alpha - M_{\text{lin}}} \left( \sqrt{\frac{c_x^2 c_\varepsilon^2 \log D}{N}} + \frac{25 c_x^2 c_\varepsilon^2}{N} + \frac{4 c_x c_\varepsilon (\log(ND) + 26)}{N} \right),$$  

(2.8)

where $C > 0$ is a constant in $N, D, D_{\text{eff}}, c_x, c_\varepsilon$, and $M_{\text{lin}}$ is a scalar given by Eq. (A.28) in Appendix A.9 that satisfies, as $N \to \infty$, $M_{\text{lin}} = o(1)$. Then for $N$ sufficiently large, Condition 1 holds with probability at least $1 - 26e^{-25}$.

A full statement and proof of Theorem 2, including the exact value of $M_{\text{lin}}$, appears in Appendix A.9. A corollary of Theorem 1 and Theorem 2 together is that, under Assumptions 1 to 5, the LOOCV approximations $I_{J \setminus n}$ and $N_{S \setminus n}$ have accuracy that depends on (the ideally small) $D_{\text{eff}}$ rather than (the potentially large) $D$.

It is worth considering how the allowed values of $\lambda$ in Eq. (2.8) compare to previous results in the $\ell_1$ literature for the support recovery of $\hat{\theta}$. We will talk about a sequence of choices of $\lambda$ scaling with $N$ denoted by $\lambda_N$. Theorem 11.3 of Hastie et al. [2015] shows that $\lambda_N \geq c \sqrt{\log(D)/N}$ (for some constant $c$ in $D$ and $N$) is sufficient for ensuring that $\text{supp } \hat{\theta} \subseteq S$ with high probability in the case of linear regression. Thus, we ought to set $\lambda_N \geq c \sqrt{\log(D)/N}$ to ensure support recovery of $\hat{\theta}$. Compare this constraint on $\lambda_N$ to the constraint implied by Eq. (2.8). We have that $M_{\text{lin}} = o(1)$ as $N \to \infty$, so that, for large $N$, the bound in Eq. (2.8) becomes $\lambda_N \geq c' \sqrt{\log(D)/N}$ for some constant $c'$. Thus, the sequence of $\lambda_N$ satisfying Eq. (2.8) scales at exactly the same rate as those that ensure $\text{supp } \hat{\theta} \subseteq S$. The scaling of $\lambda_N$ is important, as the error in $\hat{\theta}$, $\| \hat{\theta} - \theta^* \|_2^2$, is typically proportional to $\lambda_N$. The fact that we have not increased the asymptotic scaling of $\lambda_N$ therefore means that we can enjoy the same decay of $\| \hat{\theta} - \theta^* \|_2^2$ while ensuring $\text{supp } \hat{\theta} \setminus n = S$ for all $n$.

2.4.2 Logistic regression

We now give the distributional assumption on the responses $y_n$ in the case of logistic regression.

Assumption 6. $\forall n$, we have $y_n \in \{\pm 1\}$ with $\Pr[y_n = 1] = 1/(1 + e^{-x_n^T \theta^*})$.

We will also need a condition on the minimum eigenvalue of the Hessian.

Assumption 7. Assume for some scalar $L_{\text{min}}$ that may depend on $N, D_{\text{eff}},$ and $c_x$, we have

$$\Pr[\lambda_{\text{min}} (\nabla^2_{\theta} F(\theta^*)_{SS}) \leq L_{\text{min}}] \leq e^{-25}.$$  

Furthermore, assume the scaling of $L_{\text{min}}$ in $N$ and $D_{\text{eff}}$ is such that, under Assumption 8 and for sufficiently large $N$, $L_{\text{min}} \geq CN$ for some constant $C$ that may depend on $c_x$.  

31
Figure 2-2: Error (Eq. (2.10)) across approximations for $\ell_1$ LOOCV (legend shared with Fig. 2-3). The error for $IJ_n$ (black dashed) is too small to see, but nonzero; it varies between $-0.06\%$ and $0.04\%$.

In the case of linear regression, we did not need an analogue of Assumption 7, as standard matrix concentration results tell us that its Hessian satisfies Assumption 7 with $L_{\text{min}} = N - Cc_x^2\sqrt{ND_{\text{eff}}}$ (see Lemma 4 in Appendix A.9). The Hessian for logistic regression is significantly more complicated, and it is typical in the $\ell_1$ literature to make some kind of assumption about its eigenvalues [Bach, 2010; Li et al., 2015]. Empirically, Assumption 7 is satisfied when Assumptions 1 and 6 hold; however we are unaware of any results in the literature showing this is the case.

**Theorem 3** (Logistic Regression). Take Assumptions 1 to 4, 6 and 7. Suppose the regularization parameter $\lambda$ satisfies:

$$\lambda \geq \frac{C}{\alpha - M_{\text{logr}}} \left(\sqrt{c_x^2} \frac{25 + \log D}{N} + \sqrt{2c_x^2 \log(ND) + \sqrt{50c_x^2 N}}\right),$$

(2.9)

where $C, C'$ are constants in $N, D, D_{\text{eff}}$, and $c_x$, and $M_{\text{logr}}$ is a scalar given by Eq. (A.59), that, as $N \to \infty$, satisfies $M_{\text{logr}} = o(1)$. Then for $N$ sufficiently large, Condition 2 is satisfied with probability at least $1 - 43e^{-25}$.

A restatement and proof of Theorem 3 are given as Theorem 7 in Appendix A.9. Similar to the remarks after Theorem 2, Theorem 3 implies that when applied to logistic regression, $IJ_n$ and $NS_n$ have accuracy that depends on (the ideally small) $D_{\text{eff}}$ rather than (the potentially large) $D$, even when $D = o(e^N)$.

Theorem 3 has implications for the work of Obuchi and Kabashima [2018], who conjecture that, as $N \to \infty$, the change in support of $\ell_1$ regularized logistic regression becomes negligible as each datapoint is left out; this assumption is used to derive a version of $NS_n$ for logistic regression. Our Theorem 3 confirms this conjecture by proving the stronger fact that the support is unchanged with high probability for finite data.
Figure 2-3: Runtimes for the experiments in Fig. 2-2 with exact CV (red) included for comparison. The $D \times D$ matrix inversion in the smoothed problem is so slow that even exact CV with an efficient $\ell_1$ solver is faster.

Figure 2-4: Log percent accuracy (Eq. (2.10)) for real data experiments. For each dataset, we give the accuracy of approximate CV compared to exact CV using both a smoothed approximation to $\ell_1$ and the $IJ_n$, $NS_n$ approximations. For the bcTCGA dataset (linear regression), the nearly quadratic objective seems to be extremely well approximated by one Newton step, making $\tilde{NS}_n(R^n)$ significantly more accurate than $\tilde{I}_n(R^n)$; see the note at the end of Appendix A.4.4 about the exactness of $\tilde{NS}_n(R)$ on quadratic objectives.
2.5 Experiments

We now empirically verify the good behavior of NS\(n\) and IJ\(n\) (i.e. proposal #4) and show that it far outperforms #2 (smoothing \(\ell_1\)) and #3 (subsampling) in our high-dimensional regime of interest. We focus comparisons in this section on proposals #2–#4, as they all directly address \(\ell_1\)-regularized problems. For an illustration of the failings of proposal #1, see Appendix A.3. To illustrate #2, we consider the smooth approximation given by Rad and Maleki [2020]:

\[
\begin{align*}
R_\eta(\theta) := \sum_{d=1}^{D} \frac{1}{\eta} \left( \log(1 + e^{\eta \theta_d}) + \log(1 + e^{-\eta \theta_d}) \right).
\end{align*}
\]

While \(\lim_{\eta \to \infty} R_\eta(\theta) = \|\theta\|_1\), we found that this approximation became numerically unstable for optimization when \(\eta\) was much larger than 100, so we set \(\eta = 100\) in our experiments.

Simulated experiments. First, we trained logistic regression models on twenty-five random datasets in which \(x_{nd} \sim \mathcal{N}(0, 1)\) with \(N = 500\) and \(D = 40,000\). We set \(\lambda = 1.5\sqrt{\log(D)/N}\) to mimic our condition in Eq. (2.9). The true \(\theta^*\) was supported on its first five entries. We evaluate our approximations by comparing the CV estimate of out-of-sample error (“LOO”) to the approximation \(\text{ALOO} := \frac{1}{N} \sum_{n=1}^{N} f(x^T IJ_n, y_n)\).

We report percent error:

\[
\frac{|\text{ALOO} - \text{LOO}|}{\text{LOO}}.
\]

(2.10)

Fig. 2-2 compares the accuracy and run times of proposals #2 and #3 versus IJ\(n\). We chose the number of subsamples so that subsampling CV would have about the same runtime as computing IJ\(n\) for all \(n\). We see that subsampling usually has much worse accuracy than IJ\(n\). Using \(\tilde{IJ}_n(R)\) with \(R^{100}(\theta)\) as a regularizer is even worse, as we approximate over all \(D\) dimensions; the resulting approximation is slower and less accurate – by multiple orders of magnitude – across all trials.

The importance of setting \(\lambda\). Our theoretical results heavily depend on particular settings of \(\lambda\) to obtain the fixed-dimensional error scaling shown in blue in Fig. 2-1. One might wonder if such a condition on \(\lambda\) is necessary for approximate CV to be accurate. We offer evidence in Appendix A.6 that this scaling is necessary by empirically showing that when \(\lambda\) violates our condition, the error in IJ\(n\) grows with \(N\).

Real data experiments. We next study how dependent our results are on the particular distributional assumptions Theorems 2 and 3. We explore this question with a number of publicly available datasets [bcTCGA, 2018; Lewis et al., 2004; Guyon et al., 2004]. We chose these datasets because they have a high enough dimension to observe the effect of our results, yet are not so large that running exact CV for comparison is prohibitively expensive; see Appendix A.7 for details (including our settings of \(\lambda\)). For each dataset, we approximate CV for the \(\ell_1\) regularized model using IJ\(n\) and NS\(n\). For comparison, we report the accuracy of \(\tilde{IJ}_n(R^\eta)\) and \(\tilde{NS}_n(R^\eta)\) with \(\eta = 100\). Our results in Fig. 2-4 show that IJ\(n\) is significantly faster and more accurate than exact CV or smoothing.

\[\text{Specifically, we computed 41 different } \hat{\theta}^n\text{ for each trial in order to roughly match the time cost of computing IJ}_n\text{ for all } N = 500 \text{ datapoints.}\]
To demonstrate the scalability of our approximations, we re-ran our RCV1 experiment on a larger version of the dataset with $N = 20,242$ and $D = 30,000$. Based on the time to compute exact LOOCV for twenty datapoints, we estimate exact LOOCV would have taken over two weeks to complete, whereas computing both NS$_n$ and IJ$_n$ for all $n$ took three minutes.

2.6 Conclusions and future work

We have provided the first analysis of when CV can be approximated quickly and accurately in high dimensions with guarantees on quality. We have seen that, out of a number of proposals in the literature, running approximate CV on the recovered support (i.e., NS$_n$ and IJ$_n$) forms the only proposal that reaches these goals both theoretically and empirically. We hope this analysis will serve as a starting point for further understanding of when approximate CV methods work for high-dimensional problems.

We see three interesting directions for future work. First, this work has focused entirely on approximate CV for model assessment. In Appendix A.8, we show that approximate CV for model selection can have unexpected and undesirable behavior; we believe understanding this behavior is one of the most important future directions in this area. Second, one could extend our results to results to the higher order infinitesimal jackknife presented in Giordano et al. [2019a]. Finally, it would be interesting to consider our approximations as a starting point for subsampling estimators, as proposed in Magnusson et al. [2019].
Chapter 3

Approximate cross-validation with low-rank data in high dimensions

3.1 Introduction

Recent machine learning advances are driven at least in part by increasingly rich data sets — large in both data size $N$ and dimension $D$. The proliferation of data and algorithms makes cross-validation (CV) [Stone 1974, Geisser 1975, Musgrave et al. 2020] an appealing tool for model assessment due its ease of use and wide applicability. For high-dimensional data sets, leave-one-out CV (LOOCV) is often especially accurate as its folds more closely match the true size of the data [Burman 1989]; see also Figure 1 of [Rad and Maleki 2020]. Traditionally many practitioners nonetheless avoid LOOCV due its computational expense; it requires re-running an expensive machine learning algorithm $N$ times. To address this expense, a number of authors have proposed approximate cross-validation (ACV) methods [Beirami et al. 2017, Rad and Maleki 2020, Giordano et al. 2019b]; these methods are fast to run on large data sets, and both theory and experiments demonstrate their accuracy. But these methods struggle in high-dimensional problems in two ways. First, they require inversion of a $D \times D$ matrix, a computationally expensive undertaking. Second, their accuracy can degrade in high dimensions; see Fig. 1 of Stephenson and Broderick 2020 for a classification example and Fig. 3-1 below for a count-valued regression example. Koh and Liang 2017, Lorraine et al. 2020 have investigated approximations to the matrix inverse for problems similar to ACV, but these approximations do not work well for ACV itself; see [Stephenson and Broderick 2020, Appendix B]. Stephenson and Broderick 2020 demonstrate how a practitioner might avoid these high-dimensional problems in the presence of sparse data. But sparsity may be a somewhat limiting assumption.

We here consider approximately low-rank (ALR) data. Udell and Townsend 2019 argue that ALR data matrices are pervasive in applications ranging from fluid dynamics and genomics to social networks and medical records — and that there are theoretical reasons to expect ALR structure in many large data matrices. For concreteness and to facilitate theory, we focus on fitting generalized linear models (GLMs). We
Figure 3-1: Accuracy of the IJ approximation in Eq. (3.4) for a synthetic Poisson regression problem versus the dataset size $N$. Red shows the accuracy when the data dimension is fixed at $D = 40$, blue when the dimension grows as $D = N/10$, and black when the dimension grows as $D = N/10$ but with a fixed rank of 40. High-dimensional yet low-rank data has identical performance to low-dimensional data.

note that GLMs are a workhorse of practical data analysis; as just one example, one of many popular books on GLMs [McCullagh, 1989] has been cited over 9,000 times since 2015 as of this writing. While accurate ACV methods for GLMs alone thus have potential for great impact, we expect many of our insights may extend beyond both GLMs and LOOCV (i.e. to other CV and bootstrap-like “retraining” schemes).

In particular, we propose an algorithm for fast, accurate ACV for GLMs with high-dimensional covariate matrices — and provide computable upper bounds on the error of our method relative to exact LOOCV. Two major innovations power our algorithm. First, we prove that existing ACV methods automatically obtain high accuracy in the presence of high-dimensional yet ALR data. Our theory provides cheaply computable upper bounds on the error of existing ACV methods. Second, we notice that the $D \times D$ matrix that needs to be inverted in ACV is ALR when the covariates are ALR. We propose to use a low-rank approximation to this matrix. We provide a computable upper bound on the extra error introduced by using such a low-rank approximation. By studying our bound, we show the surprising fact that, for the purposes of ACV, the matrix is well approximated by using its largest eigenvalues, despite the fact that ACV uses the matrix inverse. We demonstrate the speed and accuracy of both our method and bounds with a range of experiments.

### 3.2 Background: approximate CV methods

We consider fitting a generalized linear model (GLM) with parameter $\theta \in \mathbb{R}^D$ to some dataset with $N$ observations, $\{x_n, y_n\}_{n=1}^N$, where $x_n \in \mathbb{R}^D$ are covariates and $y_n \in \mathbb{R}$ are responses. We suppose that the $x_n$ are approximately low rank (ALR); that is, the matrix $X \in \mathbb{R}^{N \times D}$ with rows $x_n$ has many singular values near zero. These small singular values can amplify noise in the responses. Hence it is common to use $\ell_2$
regularization to ensure that our estimated parameter $\hat{\theta}$ is not too sensitive to the subspace with small singular values; the rotational invariance of the $\ell_2$ regularizer automatically penalizes any deviation of $\theta$ away from the low-rank subspace [Hastie et al., 2009, Sec. 3.4]. Thus we consider:

$$\hat{\theta} := \arg \min_{\theta \in \mathbb{R}^D} \frac{1}{N} \sum_{n=1}^{N} f(x_n^T \theta, y_n) + \frac{\lambda}{2} \|\theta\|_2^2,$$

(3.1)

where $\lambda \geq 0$ is some regularization parameter and $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is convex in its first argument for each $y_n$. Throughout, we assume $f$ to be twice differentiable in its first argument. To use leave-one-out CV (LOOCV), we compute $\hat{\theta} \setminus n$, the estimate of $\theta$ after deleting the $n$th datapoint from the sum, for each $n$. To assess the out-of-sample error of our fitted $\hat{\theta}$, we then compute:

$$\frac{1}{N} \sum_{n=1}^{N} \text{Err}(x_n^T \hat{\theta} \setminus n, y_n),$$

(3.2)

where $\text{Err} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is some function measuring the discrepancy between the observed $y_n$ and its prediction based on $\hat{\theta} \setminus n$ — for example, squared error or logistic loss.

Computing $x_n^T \hat{\theta} \setminus n$ for every $n$ requires solving $N$ optimization problems, which can be a prohibitive computational expense. Approximate CV (ACV) methods aim to alleviate this burden via one of two principal approaches below. Denote the Hessian of the objective by $H := (1/N) \sum_{n=1}^{N} \nabla^2 f(x_n^T \hat{\theta}, y_n) + \lambda I_D$ and the $k$th scalar derivative of $f$ as $\hat{D}^{(k)}_n := d^k f(z, y_n)/dz^k |_{z=x_n^T \hat{\theta}}$. Finally, let $Q_n := x_n^T H^{-1} x_n$ be the $n$th quadratic form on $H^{-1}$. The first approximation, based on taking a Newton step from $\hat{\theta}$ on the objective $(1/N) \sum_{m \neq n} f(x_m^T \hat{\theta}, y_m) + \lambda \|\theta\|_2^2$, was proposed by Obuchi and Kabashima [2016, 2018], Rad and Maleki [2020], Beirami et al. [2017]. We denote this approximation by NS$_{\setminus n}$; specializing to GLMs, we have:

$$x_n^T \hat{\theta} \setminus n \approx x_n^T \text{NS}_{\setminus n} := x_n^T \hat{\theta} + \frac{\hat{D}^{(1)}_n Q_n}{N - \hat{D}^{(2)}_n Q_n}.$$

(3.3)

Throughout we focus on approximating $x_n^T \hat{\theta} \setminus n$, rather than $\hat{\theta} \setminus n$, since $x_n^T \hat{\theta} \setminus n$ is the argument of Eq. (3.2). See Appendix [B.1] for a derivation of Eq. (3.3). The second approximation we consider is based on the infinitesimal jackknife [Jaeckel, 1972, Efron, 1982]; it was conjectured as a possible ACV method by Koh and Liang [2017], used in a comparison by Beirami et al. [2017], and studied in depth by Giordano et al. [2019b]. We denote this approximation by IJ$_{\setminus n}$; specializing to GLMs, we have:

$$x_n^T \hat{\theta} \setminus n \approx x_n^T \text{IJ}_{\setminus n} := x_n^T \hat{\theta} + \left( \frac{\hat{D}^{(1)}_n}{N} \right) Q_n.$$  

(3.4)

See Appendix [B.1] for a derivation. We consider both NS$_{\setminus n}$ and IJ$_{\setminus n}$ in what follows as the two have complementary strengths. In our experiments in Section 3.6, NS$_{\setminus n}$
tends to be more accurate; we suspect that GLM users should generally use NS\_n. On the other hand, NS\_n requires the inversion of a different $D \times D$ matrix for each $n$. In the case of LOOCV for GLMs, each matrix differs by a rank-one update, so standard matrix inverse update formulas allow us to derive Eq. (3.3), which requires only a single inverse across folds. But such a simplification need not generally hold for models beyond GLMs and data re-weightings beyond LOOCV (e.g. other forms of CV or the bootstrap). By contrast, even beyond GLMs and LOOCV, the IJ requires only a single matrix inverse for all $n$.

In any case, we notice that existing theory and experiments for both NS\_n and IJ\_n tend to either focus on low dimensions or show poor performance in high dimensions; see Appendix B.3 for a review. One problem is that error in both approximations can grow large in high dimensions. See [Stephenson and Broderick, 2020] for an example; also, in Fig. 3-1, we show the IJ\_n on a synthetic Poisson regression task. When we fix $D = 40$ and $N$ grows, the error drops quickly; however, if we fix $D/N = 1/10$ the error is substantially worse. A second problem is that both NS\_n and IJ\_n rely on the computation of $Q_n = x_n^T H^{-1} x_n$, which in turn relies on computation\[1\] of $H^{-1}$. The resulting $O(D^3)$ computation time quickly becomes impractical in high dimensions. Our major contribution is to show that both of these issues can be avoided when the data are ALR.

### 3.3 Methodology

**Algorithm 1** Approximation to $\{x_n^T \hat{\theta}^{(n)}\}_{n=1}^N$ for low-rank GLMs

1: procedure APPXLOOCV($\hat{\theta}, X, \lambda, \{\hat{D}^{(1)}_n\}_{n=1}^N, \{\hat{D}^{(2)}_n\}_{n=1}^N, K$)
2: $B \leftarrow X^T \text{diag}\{\hat{D}^{(2)}_n\}_{n=1}^N X$ $\triangleright$ The Hessian, $H$, equals $B + \lambda I_D$
3: $\{\tilde{Q}_n\}_{n=1}^N \leftarrow \text{APPXQN}(B, K, \lambda)$ $\triangleright$ Uses rank-$K$ decomposition of $B$
4: for $n = 1, \ldots, N$ do
5: either $x_n^T \tilde{\text{NS}} \_n \leftarrow x_n^T \text{NS} \_n(\tilde{Q}_n)$ $\triangleright$ i.e., compute Eq. (3.3) using $\tilde{Q}_n$ instead of $Q_n$
6: or $x_n^T \tilde{\text{IJ}} \_n \leftarrow x_n^T \text{IJ} \_n(\tilde{Q}_n)$ $\triangleright$ i.e., compute Eq. (3.4) using $\tilde{Q}_n$ instead of $Q_n$
7: end for
8: return $\{x_n^T \tilde{\text{NS}} \_n\}_{n=1}^N$ or $\{x_n^T \tilde{\text{IJ}} \_n\}_{n=1}^N$ $\triangleright$ User’s choice
9: end procedure

We now present our algorithm for fast, approximate LOOCV in GLMs with ALR data. We then state our main theorem, which (1) bounds the error in our algorithm relative to exact CV, (2) gives the computation time of our algorithm, and (3) gives

\[1\] In practice, for numerical stability, we compute a factorization of $H$ so that $H^{-1}x_n$ can be quickly evaluated for all $n$. However, for brevity, we refer to computation of the inverse of $H$ throughout.
the computation time of our bounds. Finally we discuss the implications of our theorem before moving on to the details of its proof in the remainder of this chapter.

Our method appears in Algorithm 1. To avoid the $O(D^3)$ matrix inversion cost, we replace $H$ by $\tilde{H} \approx H$, where $\tilde{H}$ uses a rank-$K$ approximation and can be quickly inverted. We can then use $\tilde{H}$ to compute $\tilde{Q}_n = Q_n$, which enters into either the NS or IJ approximation, as desired.

Before stating Theorem 4, we establish some notation. We will see in Proposition 3 of Section 3.4 that we can provide computable upper bounds $\eta_n \geq |\tilde{Q}_n - Q_n|$; $\eta_n$ will enter directly into the error bound for $x_n^T \tilde{I}_{\setminus n}$ in Theorem 4 below. To bound the error of $x_n^T \tilde{N}_{\setminus n}$, we need to further define

$$E_n := \max \left\{ \left\| \frac{\tilde{Q}_n + \eta_n}{|1 - \tilde{D}_n^{(2)}(\tilde{Q}_n + \eta_n)|} - \frac{\tilde{Q}_n}{|1 - \tilde{D}_n^{(2)}\tilde{Q}_n|^2} \right\|, \left\| \frac{\tilde{Q}_n - \eta_n}{|1 - \tilde{D}_n^{(2)}(\tilde{Q}_n - \eta_n)|} - \frac{\tilde{Q}_n}{|1 - \tilde{D}_n^{(2)}\tilde{Q}_n|^2} \right\| \right\}. $$

Additionally, we will see in Proposition 2 of Section 3.4 that we can bound the “local Lipschitz-ness” of the Hessian related to the third derivatives of $f$ evaluated at some $z$, $\tilde{D}_n^{(3)}(z) := d^3f(z, y_n)/dz^3 |_{z=z}$. We will denote our bound by $M_n$:

$$M_n \geq \left( \frac{1}{N} \sum_{m \neq n} \| x_m \|_2^2 \right) \max_{s \in [0, 1]} \left| \hat{D}_n^{(3)} \left( x_n^T((1-s)\hat{\theta} + s\hat{\theta}^n) \right) \right|, \quad (3.5)$$

We are now ready to state, and then discuss, our main result — which is proved in Appendix B.4.3.

**Theorem 4.** (1) **Accuracy:** Let $\eta_n \geq |Q_n - \tilde{Q}_n|$ be the upper bound produced by Proposition 3 and $M_n$ the local Lipschitz constants computed in Proposition 2. Then the estimates $x_n^T \tilde{N}_{\setminus n}$ and $x_n^T \tilde{I}_{\setminus n}$ produced by Algorithm 1 satisfy:

$$|x_n^T \tilde{N}_{\setminus n} - x_n^T \tilde{\theta}^n| \leq \frac{M_n}{N^2 \lambda^3} |\hat{D}_n^{(1)}|^2 \| x_n \|_2^2 + |\hat{D}_n^{(1)}| E_n \quad (3.6)$$

$$|x_n^T \tilde{I}_{\setminus n} - x_n^T \tilde{\theta}^n| \leq \frac{M_n}{N^2 \lambda^3} |\hat{D}_n^{(1)}|^2 \| x_n \|_2^2 + \frac{1}{N^2 \lambda^3} |\hat{D}_n^{(1)}| \| \hat{D}_n^{(2)} \|_2 \| x_n \|_2^2 + |\hat{D}_n^{(1)}| \eta_n. \quad (3.7)$$

(2) **Algorithm computation time:** The runtime of Algorithm 1 is in $O(NDK + K^3)$. (3) **Bound computation time:** The upper bounds in Eqs. (3.6) and (3.7) are computable in $O(DK)$ time for each $n$ for common GLMs such as logistic and Poisson regression.

To interpret the running times, note that standard ACV methods have total runtime in $O(ND^2 + D^3)$. So Algorithm 1 represents a substantial speedup when the dimension $D$ is large and $K \ll D$. Also, note that our bound computation time has no worse behavior than our algorithm runtime. We demonstrate in our experiments (Section 3.6) that our error bounds are both computable and useful in practice. To help interpret the bounds, note that they contain two sources of error: (A) the error of our additional approximation relative to existing ACV methods (i.e. the use of $\tilde{Q}_n$).
and (B) the error of existing ACV methods in the presence of ALR data. Our first corollary notes that (A) goes to zero as the data becomes exactly low rank.

**Corollary 1.** *As the data becomes exactly low rank with rank $R$ (i.e., $X$’s lowest singular values $\sigma_d \to 0$ for $d = R+1, \ldots, D$), we have $\eta_n, E_n \to 0$ if $K \geq R$."

See Appendix B.4.4 for a proof. Our second corollary gives an example for which the error in existing (exact) ACV methods (B) vanishes as $N$ grows.

**Corollary 2.** *Suppose the third derivatives $\hat{D}_n^{(3)}$ and the $x_n$ are both bounded and the data are exactly low-rank with constant rank $R$. Then with $N \to \infty$, $D$ growing at any rate, and $K$ arbitrary, the right hand sides of Eqs. (3.6) and (3.7) reduce to $|\hat{D}_n^{(1)}|E_n$ and $|\hat{D}_n^{(1)}|\eta_n$, respectively.*

We note that Corollary 2 is purely illustrative, and we strongly suspect that none of its conditions are necessary. Indeed, our experiments in Section 3.6 show that the bounds of Theorem 4 imply reasonably low error for non-bounded derivatives with ALR data and only moderate $N$.

### 3.4 Accuracy of exact ACV with approximately low-rank data

Recall that the main idea behind Algorithm 1 is to compute a fast approximation to existing ACV methods by exploiting ALR structure. To prove our error bounds, we begin by proving that the exact ACV methods NS\_ and IJ\_ approximately (respectively, exactly) retain the low-dimensional accuracy displayed in red in Fig. 3.6 when applied to GLMs with approximately (respectively, exactly) low-rank data.

Let us first define low-rank data. Let $X = U \Sigma V^T$ be the singular value decomposition of $X$, where $U \in \mathbb{R}^{N \times D}$ has orthonormal columns, $\Sigma \in \mathbb{R}^{D \times D}$ is a diagonal matrix, and $V \in \mathbb{R}^{D \times D}$ is an orthonormal matrix.

**Definition 2.** *We say that a matrix $X$ with singular value decomposition $X = U \Sigma V$ is of exactly low-rank $R$ if $\Sigma_{dd} = 0$ for all $d > R$. We say that $X$ is of approximately low rank (ALR) $R$ if $\Sigma_{dd} \approx 0$ for all $d > R$.*

We note that this definition of ALR is different from that in [Udell and Townsend 2019], which we gave in Section 3.1 as a motivation for considering ALR data. In particular, [Udell and Townsend 2019] define a matrix $X$ to be of ALR if it is entry-wise $\varepsilon$-close to some matrix of exactly low-rank $R$; such a matrix can be very different from our definition of ALR, as such a matrix can have $\Sigma_{R+1,R+1} = D\varepsilon$. While we only consider [Udell and Townsend 2019] as general motivation, we note that in our work below, we will consider the case of $\Sigma_{dd} \to 0$, making the two definitions of ALR equivalent.

We now show that existing ACV methods are accurate in the presence of exactly low-rank data. Let $V_R$ be the top $R$ right singular vectors of $X$ (i.e. the first $R$
columns of \(V\), and fit a model restricted to \(R\) dimensions as:
\[
\hat{\phi} := \arg\min_{\phi \in \mathbb{R}^R} \frac{1}{N} \sum_{n=1}^{N} f((V_R^n)^T \phi) + \frac{\lambda}{2} \|\phi\|_2^2.
\]

Let \(\hat{\phi}_n\) be the \(n\)th leave-one-out parameter estimate from this problem, and let \(\text{RNS}_n\) and \(\text{RIJ}_n\) be Eq. (3.3) and Eq. (3.4) applied to this restricted problem. We can now show that the error of \(\text{IJ}_n\) and \(\text{NS}_n\) applied to the full \(D\)-dimensional problem is exactly the same as the error of \(\text{RNS}_n\) and \(\text{RIJ}_n\) applied to the restricted \(R \ll D\) dimensional problem.

**Lemma 1.** Assume that the data matrix \(X\) is exactly low-rank \(R\). Then \(|x_n^T \text{NS}_n - x_n^T \hat{\theta}^n| = |(V_R^n)^T \text{RNS}_n - (V_R^n)^T \hat{\phi}_n|\) and \(|x_n^T \text{IJ}_n - x_n^T \hat{\theta}^n| = |(V_R^n)^T \text{RIJ}_n - (V_R^n)^T \hat{\phi}_n|\)

See Appendix B.4.1 for a proof. Based on previous work (e.g., Beirami et al., 2017, Rad and Maleki, 2020, Giordano et al., 2019b), we expect the ACV errors \(|(V_R^n)^T \text{RNS}_n - (V_R^n)^T \hat{\phi}_n|\) and \(|(V_R^n)^T \text{RIJ}_n - (V_R^n)^T \hat{\phi}_n|\) to be small, as they represent the errors of \(\text{NS}_n\) and \(\text{IJ}_n\) applied to an \(R\)-dimensional problem. We confirm Lemma 1 numerically in Fig. 3-1, where the error for the \(D = 40\) problems (red) exactly matches that of the high-\(D\) but exact low-rank \(R = 40\) problems (black).

However, real-world covariate matrices \(X\) are rarely exactly low-rank. By adapting results from Wilson et al., 2020, we can give bounds that smoothly decay as we leave the exact low-rank setting of Lemma 1. To that end, define:
\[
L_n := \left(\frac{1}{N} \sum_{m: m \neq n} \|x_m\|^2 \right) \max_{s \in [0, 1]} \hat{D}_n^{(3)} \left( x_n^T ((1 - s)\hat{\theta} + s\hat{\phi}_n) \right). 
\]

**Lemma 2.** Assume that \(\lambda > 0\). Then, for all \(n\):
\[
| x_n^T \text{NS}_n - x_n^T \hat{\theta}^n | \leq \frac{L_n}{N^2 \lambda^3} |\hat{D}_n^{(1)}| \| x_n \|^2 
\]
\[
| x_n^T \text{IJ}_n - x_n^T \hat{\theta}^n | \leq \frac{L_n}{N^2 \lambda^3} |\hat{D}_n^{(1)}| \| x_n \|^3 + \frac{1}{N^2 \lambda^3} |\hat{D}_n^{(1)}| |\hat{D}_n^{(2)}| \| x_n \|^2. 
\]

Furthermore, these bounds continuously decay as the data move from exactly to approximately low rank in that they are continuous in the singular values of \(X\).

The proofs of Eqs. (3.9) and (3.10) mostly follow from results in Wilson et al., 2020, although our results removes a Lipschitz assumption on the \(\hat{D}_n^{(2)}\); see Appendix B.4.2 for a proof.

Our bounds are straightforward to compute; we can calculate the norms \(\|x_n\|_2\) and evaluate the derivatives \(\hat{D}_n^{(1)}\) and \(\hat{D}_n^{(2)}\) at the known \(x_n^T \hat{\theta}\). The only unknown quantity is \(L_n\). However, we can upper bound the \(L_n\) using the following proposition.
Proposition 2. Let $\mathcal{Z}_n$ be the set of $z \in \mathbb{R}$ such that $|z| \leq |x_n^T \hat{\theta}| + |\hat{D}_n^{(1)}||x_n||_2^2/(N \lambda)$. For $L_n$ as defined in Eq. (3.8), we have the upper bound:

$$L_n \leq M_n := \max_{z \in \mathcal{Z}_n} |\hat{D}_n^{(3)}(z)| \left( \frac{1}{N} \sum_{m:m \neq n} \|x_m\|_2^2 \right).$$  

(3.11)

To compute an upper bound on the $M_n$ in turn, we can optimize $\hat{D}_n^{(3)}(z)$ for $|z| \leq |x_n^T \hat{\theta}| + |\hat{D}_n^{(1)}||x_n||_2^2/(N \lambda)$. This scalar problem is straightforward for common GLMs: for logistic regression, we can use the fact that $|\hat{D}_n^{(3)}| \leq \frac{1}{4}$, and for Poisson regression with an exponential link function (i.e., $y_n \sim \text{Poisson} \left( \exp(x_n^T \theta) \right)$), we maximize $\hat{D}_n^{(3)}(z) = e^z$ with the largest $z \in \mathcal{Z}_n$.

3.5 Approximating the quadratic forms $Q_n$

**Algorithm 2** Estimate $Q_n = x_n^T(B + \lambda I_D)^{-1}x_n$ via a rank-$K$ decomposition of PSD matrix $B$. Note: as written, this procedure is not numerically stable. See Appendix B.5.3 for an equivalent but numerically stable version.

1: procedure APPXQN($B, K, \lambda$)
2: for $k = 1, \ldots, K$ do
3:  $\mathcal{E}_k \leftarrow \mathcal{N}(0_D, I_D)$ \hfill $\triangleright \mathcal{E} \in \mathbb{R}^{D \times K}$ has i.i.d. $\mathcal{N}(0, 1)$ entries
4: end for
5: $\Omega \leftarrow \text{ORTHONORMALIZE} \left( \text{DIAG} \left\{ 1/(B_{dd} + \lambda) \right\}_{d=1}^D \right) X^T X \mathcal{E}$ \hfill $\triangleright$ Proposition 3
6: $M \leftarrow B\Omega$
7: $\tilde{H} \leftarrow M(\Omega^T M)^{-1}M^T + \lambda I_D$ \hfill $\triangleright$ Rank-$K$ Nyström approximation of $B$
8: for $n = 1, \ldots, N$ do
9:  $\tilde{Q}_n \leftarrow \min \left\{ x_n^T \tilde{H}^{-1}x_n, \|x_n\|_2^2 / (\lambda + \hat{D}_n^{(2)} \|x_n\|_2^2) \right\}$ \hfill $\triangleright$ Proposition 5
10: end for
11: return $\{\tilde{Q}_n\}_{n=1}^N$
12: end procedure

The results of Section 3.4 imply that existing ACV methods achieve high accuracy on GLMs with ALR data. However, in high dimensions, the $O(D^3)$ cost of computing $H^{-1}$ in the $Q_n$ can be prohibitive. Koh and Liang [2017], Lorraine et al. [2020] have investigated an approximation to the matrix inverse for problems similar to ACV; however, in our experiments in Appendix B.2, we find that this method does not work well for ACV. Instead, we give approximations $\tilde{Q}_n \approx Q_n$ in Algorithm 2 along with computable upper bounds on the error $|\tilde{Q}_n - Q_n|$ in Proposition 5. When the data has ALR structure, so does the Hessian $H$; hence we propose a low-rank matrix approximation to $H$. This gives Algorithm 2 a runtime in $O(NDK + K^3)$, which can result in substantial savings relative to the $O(ND^2 + D^3)$ time required to exactly compute the $Q_n$. We will see that the main insights behind Algorithm 2
come from studying an upper bound on the approximation error when using a low-rank approximation.

Observe that by construction of $\Omega$ and $\tilde{H}$ in Algorithm 2, the approximate Hessian $\tilde{H}$ exactly agrees with $H$ on the subspace $\Omega$. We can compute an upper bound on the error $|x_n^T \tilde{H}^{-1} x_n - Q_n|$ by recognizing that any error originates from components of $x_n$ orthogonal to $\Omega$:

**Proposition 3.** Let $\lambda > 0$ and suppose there is some subspace $\mathcal{B}$ on which $H$ and $\tilde{H}$ exactly agree: $\forall v \in \mathcal{B}, Hv = \tilde{H}v$. Then $H^{-1}$ and $\tilde{H}^{-1}$ agree exactly on the subspace $\mathcal{A} := H\mathcal{B}$, and

$$|x_n^T \tilde{H}^{-1} x_n - Q_n| \leq \frac{\|P_A x_n\|_2^2}{\lambda}, \text{ for all } n = 1, \ldots, N,$$

where $P_A$ denotes projection onto the orthogonal complement of $\mathcal{A}$.

For a proof, see Appendix B.5.1. The bound from Eq. (3.12) is easy to compute in $O(DK)$ time given a basis for $\mathcal{B}$. It also motivates the choice of $\Omega$ in Algorithm 2. In particular, Proposition 4 shows that $\Omega$ approximates the rank-$K$ subspace $\mathcal{B}$ that minimizes the average of the bound in Eq. (3.12).

**Proposition 4.** Let $V_K \in \mathbb{R}^{D \times K}$ be the matrix with columns equal to the right singular vectors of $X$ corresponding to the $K$ largest singular values. Then the rank-$K$ subspace $\mathcal{B}$ minimizing $\sum_n \|P_A x_n\|_2^2$ is an orthonormal basis for the columns of $H^{-1}V_K$.

**Proof.** $\sum_n \|P_A x_n\|_2^2 = \|(I_D - P_A)X^T\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm. Noting that the given choice of $\mathcal{B}$ implies that $\mathcal{A} = V_K$, the result follows from the Eckart-Young theorem.

We now see that the choice of $\Omega$ in Algorithm 2 approximates the optimal choice $H^{-1}V_K$. In particular, we use a single iteration of the subspace iteration method [Bathe and Wilson 1973] to approximate $V_K$ and then multiply by the diagonal approximation $\text{diag}\{1/\tilde{H}_{dd}\} \approx H^{-1}$. This approximation uses the top singular vectors of $X$. We expect these directions to be roughly equivalent to the largest eigenvectors of $B := \sum_n \tilde{D}_n^{(2)} x_n x_n^T$, which in turn are the largest eigenvectors of $H = B + \lambda I_D$. Thus we are roughly approximating $H$ by its largest $K$ eigenvectors.

Why is it safe to neglect the small eigenvectors? At first glance this is odd, as to minimize the operator norm $\|H^{-1} - \tilde{H}^{-1}\|_{op}$, one would opt to preserve the action of $H$ along its smallest $K$ eigenvectors. The key intuition behind this reversal is that we are interested in the action of $H^{-1}$ in the direction of the datapoints $x_n$, which, on average, tend to lie near the largest eigenvectors of $H$.

Algorithm 2 uses one additional insight to improve the accuracy of its estimates. In particular, we notice that, by the definition of $H$, each $x_n$ lies in an eigenspace of $H$ with eigenvalue at least $\tilde{D}_n^{(2)} \|x_n\|_2^2 + \lambda$. This observation undergirds the following result, which generates our final estimates $\tilde{Q}_n \approx Q_n$, along with quickly computable bounds on their error $|\tilde{Q}_n - Q_n|$.
Figure 3-2: Experiments on real data. (Left): average percent error compared to exact CV on a subset of datapoints, \( \frac{1}{20} \sum_{b=1}^{20} |x_b^T b_{\text{approx}} - x_b^T \hat{\theta}_b| / |x_b^T \hat{\theta}_b| \), where approx. denotes NS\(_n\), NS\(_n\), IJ\(_n\), or IJ\(_n\). (Right): ACV runtimes with exact CV runtimes for comparison. ACV runtimes are given for all \( N \) datapoints, and exact CV runtimes are estimated runtimes for all \( N \) datapoints.

Proposition 5. The \( Q_n = x_n^T H^{-1} x_n \) satisfy \( 0 < Q_n \leq \|x_n\|_2^2 / (\lambda + \hat{D}_n(2) \|x_n\|_2^2) \). Furthermore, letting \( \hat{Q}_n := \min \{ x_n^T H^{-1} x_n, \|x_n\|_2^2 / (\lambda + \hat{D}_n(2) \|x_n\|_2^2) \} \), we have the error bound

\[
|\hat{Q}_n - Q_n| \leq \min \left\{ \frac{\|x_n\|_2^2}{\lambda}, \frac{\|x_n\|_2^2}{\lambda + \hat{D}_n(2) \|x_n\|_2^2} \right\}. \tag{3.13}
\]

See Appendix B.5.1 for a proof. We finally note that Algorithm 2 strongly resembles algorithms from the randomized numerical linear algebra literature. Indeed, the work of Tropp et al. [2017] was the original inspiration for Algorithm 2, and Algorithm 2 can be seen as an instance of the algorithm presented in Tropp et al. [2017] with specific choices of various tuning parameters optimized for our application. For more on this perspective, see Appendix B.5.2.

3.6 Experiments

Algorithm 1 on real data. We begin by confirming the accuracy and speed of Algorithm 1 on real data compared to both exact CV and existing ACV methods. We apply logistic regression to two datasets (p53 and rcv1) and Poisson regression to one dataset (blog). p53 has a size of roughly \( N = 8,000, D = 5,000 \), and the remaining two have roughly \( N = D = 20,000 \); see Appendix B.7 for details. For all experiments we fix \( \lambda = 5.0 \); we use this moderate value of \( \lambda \) to make the underlying optimization problems regular enough so that exact CV’s runtime is reasonable. In Appendix B.8 we show that these results are not sensitive to the particular value of \( \lambda \). To further speed up computation of exact LOOCV, we only run over twenty randomly chosen datapoints. We report mean percent error, \( (1/20) \sum_{b=1}^{20} |x_b^T b_{\text{approx}} - x_b^T \hat{\theta}_b| / |x_b^T \hat{\theta}_b| \) for each exact ACV algorithm and the output of Algorithm 1. For the smaller dataset
Alternatives for estimating $Q_n$. Given our use of low-rank approximations to $H = X^T \text{diag}\{\hat{D}_n^{(2)}\}_n X + \lambda I_D$ to approximate $Q_n$, one might first consider a few more straightforward options before reaching for Algorithm 2. One might consider using principle components analysis, which is a common method for dimensionality reduction in generalized linear models. Here, this corresponds to taking the SVD of $X$ and then computing $H^{-1}$ using only the top-$K$ singular vectors and values. As discussed in Section 3.5, one might consider using the lower $K$ singular vectors and values given our use of $H^{-1}$. Similarly, one might consider using the upper or lower $K$ eigenvectors and eigenvalues of $H$. In Fig. 3-3, we study these options using the blog dataset. On the left, we see the value of our analysis in Propositions 3 and 4 as for a given value of $K$, the lower SVD and eigendecomposition give an extremely poor approximation to CV. Further, we see the value of the truncation present in our final estimate $	ilde{Q}_n$ in Proposition 5. In particular, recall that the NS$_\backslash n$ approximation depends on $1/(1 - \hat{D}_n^{(2)} Q_n)$. Thus if, for any values of $n$ and $K$, the estimate of $\hat{D}_n^{(2)} Q_n$ passes near 1, the resulting estimate of NS$_\backslash n$ will become unstable. We observe this instability for $K \leq \sim 150$ for the top SVD and eigendecomposition on the left of Fig. 3-3. We note that for a fixed value of $K$, the upper eigendecomposition of $H$ can
Figure 3-4: Error bounds implied by Theorem 4 on $\tilde{\mathcal{I}}_{\backslash n}$’s estimate of out-of-sample error using squared loss, $\text{Err}(x_n^T \theta, y_n) = (e^{x_n^T \theta} - y_n)^2$. (Left): Per datapoint error bounds. The bound is fairly loose for datapoints with larger squared loss, but tighter for points with lower squared loss. (Right): Five trials with estimates averaged across all $n$. We compute the upper (respectively, lower) error bars for $\tilde{\mathcal{I}}_{\backslash n}$ by averaging the upper (respectively, lower) bounds. While our bounds overstate the difference between exact CV and $\tilde{\mathcal{I}}_{\backslash n}$, they still give non-vacuous information on value of exact CV.

outperform our method. However, there is a significant difference in compute time: the right of Fig. 3-3 shows that the top SVD and eigendecomposition$^2$ are slower than our method for achieving a fixed accuracy.

**Accuracy of error bounds.** We next empirically check the accuracy of the error bounds from Theorem 4. We generate a synthetic Poisson regression problem with i.i.d. covariates $x_{nd} \sim \mathcal{N}(0, 1)$ and $y_n \sim \text{Poisson}(e^{x_n^T \theta^*})$, where $\theta^* \in \mathbb{R}^D$ is a true parameter with i.i.d. $\mathcal{N}(0, 1)$ entries. We generate a dataset of size $N = 800$ and $D = 500$ with covariates of approximate rank 50. To speed up the runtime of exact CV, we use $\lambda = 1.0$. In Fig. 3-4, we illustrate the utility of the bounds from Theorem 4 by estimating the out-of-sample loss with $\text{Err}(x_n^T \theta, y_n) = (e^{x_n^T \theta} - y_n)^2$. Across five trials, we show the results of exact LOOCV, our estimates provided by $\tilde{\mathcal{I}}_{\backslash n}$, and the bounds on the error of $\tilde{\mathcal{I}}_{\backslash n}$ given by Theorem 4. While our error bars in Fig. 3-4 tend to overestimate the difference between $\tilde{\mathcal{I}}_{\backslash n}$ and exact CV, they typically provide upper bounds on exact CV on the order of the exact CV estimate itself. In some cases, we have observed that the error bars can overestimate exact CV by many orders of magnitude (see Appendix B.6), but this failure is usually due to one or two datapoints $n$ for which the bound is vacuously large. As these failure cases are easy to spot by inspection, a simple fix is to resort to exact CV just for these datapoints.

$^2$We use `scipy.sparse.linalg.svds`, which uses a variant of the Arnoldi iteration, to compute the partial SVDs and eigendecompositions used here.
3.7 Conclusions

We provide an algorithm to approximate CV accurately and quickly in high-dimensional GLMs with ALR structure. Additionally, we provide quickly computable upper bounds on the error of our algorithm. We see two major directions for future work. First, while our theory and experiments focus on ACV for model assessment, the recent work of [Wilson et al., 2020] has provided theoretical results on ACV for model selection (e.g. choosing $\lambda$). It would be interesting to see how dimensionality and ALR data plays a role in this setting. Second, as noted in the introduction, we hope that the results here will provide a springboard for studying ALR structure in models beyond GLMs and CV schemes beyond LOOCV.
Chapter 4

Approximate cross-validation for structured models

4.1 Introduction

Models with complex dependency structures have become standard machine learning tools in analyses of data from science, social science, and engineering fields. These models are used to characterize disease progression [Sukkar et al., 2012, Wang et al., 2014, Sun et al., 2019], to track crime in a city [Balocchi and Jensen, 2019, Balocchi et al., 2019], and to monitor and potentially manage traffic flow [Ihler et al., 2006, Zheng and Liu, 2017] among many other applications. The potential societal impact of these methods necessitates that they be used and evaluated with care. Indeed, recent work [Musgrave et al., 2020] has emphasized that hyperparameter tuning and assessment with cross-validation (CV) [Stone, 1974, Geisser, 1975] is crucial to trustworthy and meaningful analysis of modern, complex machine learning methods.

While CV offers a conceptually simple and widely used tool for evaluation, it can be computationally prohibitive in complex models. These models often already face severe computational demands to fit just once, and CV requires multiple refits. To address this cost, recent authors [Beirami et al., 2017, Rad and Maleki, 2020, Giordano et al., 2019b] have proposed approximate CV (ACV) methods; their work demonstrates that ACV methods perform well in both theory and practice for a collection of practical models. These methods take two principal forms: one approximation based on a Newton step (NS) [Beirami et al., 2017, Rad and Maleki, 2020, Giordano et al., 2019b] and one based on the classical infinitesimal jackknife (IJ) from statistics [Koh and Liang, 2017, Beirami et al., 2017, Giordano et al., 2019b]. Though both ACV forms show promise, there remain major roadblocks to applying either NS or IJ to models with dependency structure. First, all existing ACV theory and algorithms assume that data dropped out by each CV fold are independent of the data in the other folds. But to evaluate time series models, for instance, we often drop out data points in various segments of time. Or we might drop out data within a geographic region to evaluate a spatiotemporal model. In all of these cases, the independence assumption would not apply. Second, NS methods require recomputation and inversion of a
model’s Hessian matrix at each CV fold. In the complex models we consider here, this cost can itself be prohibitive. Finally, existing theory for IJ methods requires an exact initial fit of the model – and authors so far have taken great care to obtain such a fit [Giordano et al., 2019b; Stephenson and Broderick, 2020]. But practitioners learning in e.g. large sequences or graphs typically settle for an approximate fit to limit computational cost. In this chapter, we address these concerns and thereby expand the reach of ACV to include more sophisticated models with dependencies among data points and for which exact model fits are infeasible. To avoid the cost of matrix recomputation and inversion across folds, we here focus on the IJ, rather than the NS. In particular, in Section 4.3, we develop IJ approximations for dropping out individual nodes in a dependence graph. Our methods allow us e.g. to leave out points within, or at the end of, a time series – but our methods also apply to more general Markov random fields, without a strict chain structure. In Section 4.4, we demonstrate that the IJ yields a useful ACV method even without an exact initial model fit. In fact, we show that the quality of the IJ approximation decays with the quality of the initial fit in a smooth and interpretable manner. Finally, we demonstrate our method on a diverse set of real-world applications and models in Section 4.5 and Appendix C.13. These include count data analysis with time-varying Poisson processes, named entity recognition with neural conditional random fields, motion capture analysis with auto-regressive hidden Markov models, and a spatial analysis of crime data with hidden Markov random fields.

4.2 Structured models and cross-validation

4.2.1 Structured models

Throughout we consider two types of models: (1) hidden Markov random fields (MRFs) with observations \(x\) and latent variables \(z\) and (2) conditional random fields (CRFs) with inputs (i.e., covariates) \(x\) and labels \(z\), both observed. Our developments for hidden MRFs and CRFs are very similar, but with slight differences. We detail MRFs in the main text; throughout, we will refer the reader to the appendix for the CRF treatment. We first give an illustrative example of MRFs and then the general formulation; a CRF overview appears in Appendix C.6.

Example: Hidden Markov Models (HMMs) capture sequences of observations such as words in a sentence or longitudinally measured physiological signals. Consider an HMM with \(N\) (independent) sequences, \(T\) time steps, and \(K\) states. We take each observation to have dimension \(R\). So the \(t\)th observed element in the \(n\)th sequence is \(x_{nt} \in \mathbb{R}^R\), and the latent \(z_{nt} \in [K] := \{1, \ldots, K\}\). The model is specified by (1) a distribution on the initial latent state \(p(z_{n1}) = \text{Cat}(z_{n1} | \pi)\), where \(\text{Cat}\) is the categorical distribution and \(\pi \in \Delta_{K-1}\), the \(K-1\) simplex; (2) a \(K \times K\) transition matrix \(A\) with columns \(A_k \in \Delta_{K-1}\) and \(p(z_{nt} | z_{n,t-1}) = \text{Cat}(z_{nt} | A_{z_{n,t-1}})\); and (3) emission distributions \(F\) with parameters \(\theta_k\) such that \(p(x_{nt} | z_{nt}) = F(x_{nt} | \theta_{z_{nt}})\).

We collect all parameters of the model in \(\Theta := \{\pi, \{A_k\}_{k=1}^K, \{\theta_k\}_{k=1}^K\}\). We consider
\( \Theta \) as a vector of length \( D \). We may have a prior \( p(\Theta) \).

More generally, we consider (hidden) MRFs with \( N \) structured observations \( x_n \) and latents \( z_n \), independent across \( n \in [N] \). We index single observations of dimension \( R \) (respectively, latents) within the structure by \( t \in [T] \): \( x_{nt} \in \mathbb{R}^R \) (respectively, \( z_{nt} \)). Our experiments will focus on bounded, discrete \( z_{nt} \) (i.e., \( z_{nt} \in [K] \)), but we use more inclusive notation (that might e.g. apply to continuous latents) when possible. We consider models with parameters \( \Theta \in \mathbb{R}^D \) and a single emission factor for each latent.

\[
- \log p(x, z; \Theta) = Z(\Theta) + \sum_{n=1}^{N} \left\{ \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \sum_{c \in \mathcal{F}} \phi_c(z_{nc}; \Theta) \right\},
\]

where \( z_{nc} := (z_{nt})_{t \in c} \) for \( c \subseteq [T] \); \( \psi_t \) is a log factor mapping \( (x_{nt}, z_{nt}) \) to \( \mathbb{R} \); \( \phi_c \) is a log factor mapping collections of latents, indexed by \( c \), to \( \mathbb{R} \); \( \mathcal{F} \) collects the subsets indexing factors; and \( Z(\Theta) \) is a negative log normalizing constant. HMMs, as described above, are a special case; see Appendix C.2 for details. For any MRF, we can learn the parameters by marginalizing the latents and maximizing the posterior, or equivalently the joint, in \( \Theta \). Maximum likelihood estimation is the special case with formal prior \( p(\Theta) \) constant across \( \Theta \).

\[
\hat{\Theta} := \arg \min_{\Theta} -\log p(x; \Theta) - \log p(\Theta) = \arg \min_{\Theta} -\log \int p(x, z; \Theta) \, dz - \log p(\Theta).
\]

4.2.2 Challenges of cross-validation and approximate cross-validation in structured models

In CV procedures, we iteratively leave out some data in order to diagnose variation in \( \hat{\Theta} \) under natural data variability or to estimate the predictive accuracy of our model. We consider two types of CV of interest in structured models; we make these formulations precise later. (1) We say that we consider leave-within-structure-out CV (LWCV) when we remove some data points \( x_{nt} \) within a structure and learn on the remaining data points. For instance, we might try to predict crime in certain census tracts based on observations in other tracts. Often in this case \( N = 1 \) [Celeux and Durand, 2008], [Hyndman and Athanasopoulos, 2018, Chapter 3.4], and we assume LWCV has \( N = 1 \) for notational simplicity in what follows. (2) We say that we consider leave-structure-out CV (LSCV) when we leave out entire \( x_n \) for either a single \( n \) or a collection of \( n \). For instance, with a state-space model of gene expression, we might predict some individuals’ gene expression profiles given other individuals’ profiles. In this case, \( N \gg 1 \) [Rangel et al., 2004, DeCaprio et al., 2007]. In either (1) or (2), the goal of CV is to consider multiple folds, or subsets of data, left out to assess variability and improve estimation of held-out error. But every fold incurs the cost of the learning procedure in Eq. (4.2). Indeed, practitioners have explicitly noted the high cost of using multiple folds and have resorted to using only a few, large folds
Celeux and Durand, 2008, leading to biased or noisy estimates of the out-of-sample variability.

A number of researchers have addressed the prohibitive cost of CV with approximate CV (ACV) procedures for simpler models [Beirami et al., 2017, Rad and Maleki, 2020, Giordano et al., 2019b]. Existing work focuses on the following learning problem with weights \( w \in \mathbb{R}^J \):

\[
\hat{\Theta}(w) = \arg \min_{\Theta} \sum_{j \in [J]} w_j f_j(\Theta) + \lambda R(\Theta),
\]

where \( \forall j \in [J], f_j : \mathbb{R}^D \rightarrow \mathbb{R} \) and \( \lambda \in \mathbb{R}^+ \). When the weight vector \( w \) equals the all-ones vector \( 1_J \), we recover a regularized empirical loss minimization problem. By considering all weight vectors with one weight equal to zero, we recover the folds of leave-one-out CV; other forms of CV can be similarly recovered. The notation \( \hat{\Theta}(w) \) emphasizes that the learned parameter values depend on the weights.

To see if this framework applies to LWCV or LSCV, we can interpret \( f_j \) as a negative log likelihood (up to normalization) for the \( j \)th data point and \( \lambda R \) as a negative log prior. Then the likelihood corresponding to the objective of Eq. (4.3) factorizes as

\[
p(x \mid \Theta) = \prod_{j \in J} p(x_j \mid \Theta) \propto \prod_{j \in J} \exp(-f_j(\Theta)).
\]

This factorization amounts to an independence assumption across the \( \{x_j\}_{j \in [J]} \). In the case of LWCV, with \( N = 1 \), \( j \) must serve the role of \( t \), and \( J = T \). But the \( x_t \) are not independent, so we cannot apply existing ACV methods. In the LSCV case, \( N \geq 1 \), and \( j \) in Eq. (4.3) can be seen as serving the role of \( n \), with \( J = N \). Since the \( x_n \) are independent, Eq. (4.3) can express LSCV folds.

Previous ACV work provides two primary options for the LSCV case. We give a brief review here, but see Appendix C.1 for a more detailed review. One option is based on taking a single Newton step on the LSCV objective starting from \( \hat{\Theta}(1_T) \) [Beirami et al., 2017, Rad and Maleki, 2020]. Except in special cases – such as leave-one-out CV for generalized linear models – this Newton-step approach requires both computing and inverting a new Hessian matrix for each fold, often a prohibitive expense; see Appendix C.8 for a discussion. An alternative method [Koh and Liang, 2017, Beirami et al., 2017, Giordano et al., 2019b] based on the infinitesimal jackknife (IJ) from statistics [Jaekel, 1972, Efron, 1981] constructs a Taylor expansion of \( \hat{\Theta}(w) \) around \( w = 1_T \). For any model of the form in Eq. (4.3), the IJ requires just a single Hessian matrix computation and inversion. Therefore, we focus on the IJ for LSCV and use the IJ for inspiration when developing LWCV below. However, all existing IJ theory and empirics require access to an exact minimum for \( \hat{\Theta}(1_J) \).

Indeed, previous authors [Giordano et al., 2019b, Stephenson and Broderick, 2020] have taken great care to find an exact minimum of Eq. (4.3). Unfortunately, for most complex, structured models with large datasets, finding an exact minimum requires an impractical amount of computation. Others [Bürkner et al., 2020] have developed ACV methods for Bayesian time series models and for Bayesian models without dependence structures [Vehtari et al., 2017]. Our development here focuses on empirical risk minimization and is not restricted to temporal models.

In the following, we extend the reach of ACV beyond LSCV and address the issue
of inexact optimization. In Section 4.3, we adapt the IJ framework to the LWCV problem for structured models. In Section 4.4, we show theoretically that both our new IJ approximation for LWCV and the existing IJ approximation applied to LSCV are not overly dependent on having an exact optimum. We support both of these results with practical experiments in Section 4.5.

### 4.3 Cross-validation and approximate cross-validation in structured models

We first specify a weighting scheme, analogous to Eq. (4.3), to describe LWCV in structured models; then we develop an ACV method using this scheme. Recall that CV in independent models takes various forms such as leave-$k$-out and $k$-fold CV. Similarly, we consider the possibility of leaving out $1$ multiple arbitrary sets of data indices $o \in \mathcal{O}$, where each $o \subseteq [T]$. We have two options for how to leave data out in hidden MRFs; see Appendix C.7 for CRFs. (A) For each data index $t$ left out, we leave out the data point $x_t$ but we retain the latent $z_t$. For instance, in a time series, if data is missing in the middle of the series, we still know the time relation between the surrounding points, and would leave in the latent to maintain this relation. (B) For each data index $t$ left out, we leave out the data point $x_t$ and the latent $z_t$. For instance, consider data in the future of a time series or pixels beyond the edge of a picture. We typically would not include the possibility of all possible adjacent latents in such a structure, so leaving out $z_t$ as well is more natural. In either case, analogous to Eq. (4.3), $\hat{\Theta}(w)$ is a function of $w$ computed by minimizing the negative log joint $-\log p(x; \Theta, w) - \log p(\Theta)$, now with $w$ dependence, in $\Theta$. For case (A), we adapt Eq. (4.1) (with $N = 1$) and Eq. (4.2) with a weight $w_t$ for each $x_t$ term:

$$\hat{\Theta}(w) = \arg \min_{\Theta} \left( Z(\Theta, w) + \int_z \sum_{t \in [T]} w_t \psi_t(x_t, z_t; \Theta) \right) + \left[ \sum_{c \in \mathcal{C}} \phi_c(z_c; \Theta) \right] d\mathbf{z} - \log p(\Theta) \right).$$

Note that the negative log normalizing constant $Z(\Theta, w)$ may now depend on $w$ as well. For case (B), we adapt Eq. (4.1) and Eq. (4.2) with a weight $w_t$ for each term.

---

1Note that the weight formulation could be extended to even more general reweightings in the spirit of the bootstrap. Exploring the bootstrap for structured models is outside the scope of this thesis.
with \( x_t \) or \( z_t \):

\[
\hat{\Theta}(w) = \arg\min_{\Theta} \left( Z(\Theta, w) + \int_z \left[ \sum_{t \in [T]} w_t \psi_t(x_t, z_t; \Theta) \right] + \left[ \sum_{c \in \mathcal{F}} \left( \prod_{t \in c} w_t \right) \phi_c(z_c; \Theta) \right] \, dz - \log p(\Theta) \right). 
\] (4.5)

In both cases, the choice \( w = 1_T \) recovers the original learning problem. Likewise, setting \( w = w_o \), where \( w_o \) is a vector of ones with \( w_t = 0 \) if \( t \in o \), drops out the data points in \( o \) (and latents in case (B)). We show in Appendix C.5 that these two schemes are equivalent in the case of chain-structured graphs when \( o = \{T', T'+1, \ldots, T\} \) but also that they are not equivalent in general. We thus consider both schemes going forward.

The expressions above allow a unifying viewpoint on LWCV but still require resolving \( \hat{\Theta}(w_o) \) for each new CV fold \( o \). To avoid this expense, we propose to use an IJ approach. In particular, as discussed by [Giordano et al. 2019b], the intuition of the IJ is to notice that, subject to regularity conditions, a small change in \( w \) induces a small change in \( \hat{\Theta}(w) \). So we propose to approximate \( \hat{\Theta}(w_o) \) with \( \hat{\Theta}_{IJ}(w_o) \), a first-order Taylor series expansion of \( \hat{\Theta}(w) \) as a function of \( w \) around \( w = 1_T \). We follow [Giordano et al. 2019b] to derive this expansion in Appendix C.10. Our IJ based approximation is applicable when the following conditions hold,

**Assumption 8.** The model is fit via optimization (e.g. MAP or MLE).

**Assumption 9.** The model objective is twice differentiable and the Hessian matrix is invertible at the initial model fit \( \hat{\Theta} \).

**Assumption 10.** The model fits across CV folds, \( \hat{\Theta}^o \), can be written as optima of the same weighted objective for all folds \( o \) (e.g. as in Eqs. (4.4) and (4.5)).

We summarize our method and define \( \hat{\Theta}_{ACV} \), with three arguments, in Algorithm 3; we define \( \hat{\Theta}_{IJ}(w_o) := \hat{\Theta}_{ACV}(\hat{\Theta}(1_T), x, o) \).

First, note that the proposed procedure applies to either weighting style (A) or (B) above; they each determine a different \( \log p(x, \Theta; w) \) in Algorithm 3. We provide analogous LSCV algorithms for MRFs and CRFs in Algorithms 7 and 8 (Appendices C.3 and C.7). Next, we compare the cost of our proposed ACV methods to exact CV. In what follows, we consider the initial learning problem \( \hat{\Theta}(1_T) \) a fixed cost and focus on runtime after that computation. We consider running CV for all folds \( o \in \mathcal{O} \) in the typical case where the number of data points left out of each fold, \( |o| \), is constant.

**Proposition 6.** Let \( M \) be the cost of a marginalization, i.e., running Weighted-Marg; let \( N \geq 1 \) be the number of independent structures; and let \( S \) be the maximum number of steps used to fit the parameter in our optimization procedure. The cost of any one of our ACV algorithms (Algorithms 3, 7 and 8) is in \( O(MN + D^3 + D^2 |o||\mathcal{O}|) \). Exact CV is in \( O(MNS|\mathcal{O}|) \).
Algorithm 3 Approximate leave-within-structure-out cross-validation for all folds $o \in O$

1: **procedure** Appx LWCV($\Theta_1, x, O$)
2: Define *weighted* marginalization over $z$: $\log p(x; \Theta, w) = \text{WeightedMarg}(x, \Theta, w)$.
3: Compute $H = \frac{\partial^2 \log p(x; \Theta, w) + \log p(\Theta)}{\partial \Theta \partial \Theta'} |_{\Theta=\Theta_1, w=1_T}$
4: Compute matrix $J = (J_{dt}) := \left( \frac{\partial^2 \log p(x; \Theta, w) + \log p(\Theta)}{\partial \Theta_d \partial w_t} \right) |_{\Theta=\Theta_1, w=1_T}$
5: **for** $o \in O$, **do**: $\hat{\Theta}_{ACV}(\Theta_1, x, o) := \Theta_1 + \sum_{t \in o} H^{-1} J_t \quad \triangleright J_t$ is the $t$th column of $J$
6: **end procedure**

**Proof.** For each of the $|O|$ folds of CV and each of the $N$ structures, we compute the marginalization (cost $M$) at each of the $S$ steps of the optimization procedure. In our ACV algorithms, we compute $H$ and $J$ with automatic differentiation tools [Baydin et al., 2018]. The results of Bartholomew-Biggs et al. [2000] demonstrate that $H$ and $J$ each require the same computation (up to a constant) as WeightedMarg. So, across $N$, we incur cost $MN$. We then incur a $O(D^3)$ cost to invert $H$. The remaining cost is from the for loop.

In structured problems, we generally expect $M$ to be large; see Appendix C.4 for a discussion of the costs, including in the special case of chain-structured MRFs and CRFs. And for reliable CV, we want $|O|$ to be large. So we see that our ACV algorithms reap a savings by, roughly, breaking up the product of these terms into a sum and avoiding the further $S$ multiplier.

4.4 IJ behavior under inexact optimization

By envisioning the IJ as a Taylor series approximation around $\hat{\Theta}(1_T)$, the approximations for LWCV (Algorithm 3) and LSCV (Algorithms 7 and 8 in the appendix) assume we have access to the exact optimum $\hat{\Theta}(1_T)$. In practice, though, especially in complex problems, computational considerations often require using an inexact optimum. More precisely, any optimization algorithm returns a sequence of parameter values $(\Theta^{(s)})_{s=1}^{S}$. Ideally the values $\Theta^{(S)}$ will approach the optimum $\hat{\Theta}(1_T)$ as $S \to \infty$. But we often choose $S$ such that $\Theta^{(S)}$ is much farther from $\hat{\Theta}(1_T)$ than machine precision. In practice, then, we input $\Theta^{(S)}$ (rather than $\hat{\Theta}(1_T)$) to Algorithm 3. We now check that the error induced by this substitution is acceptably low.

We focus here on a particular use of CV: estimating out-of-sample loss. For simplicity, we discuss the $N = 1$ case here; see Appendix C.11 for the very similar $N \geq 1$

---

2In practice, for numerical stability, we compute a Cholesky factorization of $H$. 

55
Eq. (4.6), which will control the difference between IJ approximation, \( \varepsilon \) are two sources of error. (1) The difference in loss between exact CV and the exact CV estimate of the out-of-sample loss is the loss (in our experiments here, negative log likelihood) on the left-out points. For each fold \( o \in \mathcal{O} \), we compute \( \hat{\Theta}(w_o) \) from the points kept in and then calculate the loss (in our experiments here, negative log likelihood) on the left-out points. I.e. the CV estimate of the out-of-sample loss is \( L_{\text{CV}} := (1/|\mathcal{O}|) \sum_{x_o \in \mathcal{O} - o} - \log p(x_o | x_{T[-o]}^T, \hat{\Theta}(w_o)) \), where \( - \log p \) may come from either weighting scheme (A) or (B). See Appendix C.11 for an extension to CV computed with a generic loss \( \ell \). We approximate \( L_{\text{CV}} \) using some \( \Theta \) as input to Algorithm 3; we denote this approximation by \( \mathcal{L}_{\text{ACV}}(\Theta, x, o) \).

Below, we will bound the error in our approximation: \( |L_{\text{CV}} - \mathcal{L}_{\text{IJ}}(\Theta^{(S)})| \). There are two sources of error. (1) The difference in loss between exact CV and the exact IJ approximation, \( \varepsilon_{IJ} \) in Eq. (4.6). (2) The difference in the parameter value, \( \varepsilon_{\Theta} \) in Eq. (4.6), which will control the difference between \( \mathcal{L}_{\text{IJ}}(\hat{\Theta}(1_T)) \) and \( \mathcal{L}_{\text{IJ}}(\Theta^{(S)}) \).

\[
\varepsilon_{IJ} := |L_{\text{CV}} - \mathcal{L}_{\text{IJ}}(\hat{\Theta}(1_T))|, \quad \varepsilon_{\Theta} := \|\Theta^{(S)} - \hat{\Theta}(1_T)\|_2 \tag{4.6}
\]

Our bound below will depend on these constants. We observe that empirics, as well as theory based on the Taylor series expansion underlying the IJ, have established that \( \varepsilon_{IJ} \) is small in various models; we expect the same to hold here. Also, \( \varepsilon_{\Theta} \) should be small for large enough \( S \) according to the guarantees of standard optimization algorithms. We now state some additional regularity assumptions before our main result.

**Assumption 11.** Take any ball \( B \subset \mathbb{R}^D \) centered on \( \hat{\Theta}(1_T) \) and containing \( \Theta^{(S)} \). We assume the objective \( -\log p(x; \Theta, 1_T) - p(\Theta) \) is strongly convex with parameter \( \lambda_{\text{min}} \) on \( B \). Additionally, on \( B \), we assume the derivatives \( g_t(\Theta) := \partial^2 \log p(x; \Theta, w_t) / \partial \Theta \partial w_t \) are Lipschitz continuous with constant \( L_g \) for all \( t \), and the inverse Hessian of the objective is Lipschitz with parameter \( L_{\text{Hinv}} \). Finally, on \( B \), take \( \log p(x; \Theta, w_o) \) to be a Lipschitz function of \( \Theta \) with parameter \( L_p \) for all \( w_o \).

We make a few remarks on the restrictiveness of these assumptions. First, while few structured models have objectives that are even convex (e.g., the label switching problem for HMMs guarantees non-convexity), we expect most objectives to be locally convex around an exact minimum \( \hat{\Theta}(1_T) \); Assumption 11 requires that the objective in fact be strongly locally convex. Next, while the Lipschitz assumption on the \( g_t \) may be hard to interpret in general, we note that it takes on a particularly simple form in the setup of Eq. (4.3), where we have \( g_t = \nabla f_t \). Finally, we note that the condition that the inverse Hessian is Lipschitz is not much of an additional restriction. E.g., if \( \nabla p(\Theta) \) is also Lipschitz continuous, then the entire objective has a Lipschitz gradient, and so its Hessian is bounded. As it is also bounded below by strong convexity, we find that the inverse Hessian is bounded above and below, and thus is Lipschitz continuous. We now state our main result.

**Proposition 7.** The approximation error of \( \mathcal{L}_{\text{IJ}}(\Theta^{(S)}) \) satisfies the following bound:

\[
|\mathcal{L}_{\text{IJ}}(\Theta^{(S)}) - \mathcal{L}_{\text{CV}}| \leq C\varepsilon_{\Theta} + \varepsilon_{IJ}, \tag{4.7}
\]

where \( C := L_p + \frac{L_p L_g}{\lambda_{\text{min}}} + \frac{L_p L_{H\text{inv}}}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \left\| \sum_{t \in o} \nabla g_t(\hat{\Theta}(1_T)) \right\|_2 \).
See Appendix [C.11] for a proof. Note that, while $C$ may depend on $\Omega$ or $\mathcal{O}$, we expect it to approach a constant as $T \to \infty$ under mild distributional assumptions on $\|g_t\|_2$; see Appendix C.11. We finally note that although the results of this section are motivated by structured models, they apply to, and are novel for, the simpler models considered in previous work on ACV methods.

4.5 Experiments

We demonstrate the effectiveness of our proposed ACV methods on a diverse set of real-world examples where data exhibit temporal and spatial dependence: namely, temporal count modeling, named entity recognition, and spatial modeling of crime data. Additional experiments validating the accuracy and computational benefits afforded by LSCV are available in Appendix [C.13.1] where we explore auto-regressive HMMs for motion capture analysis – with $N = 124$, $T$ up to 100, and $D$ up to 11,712.

Approximate leave-within-sequence-out CV: Time-varying Poisson processes. We begin by examining approximate LWCV (Algorithm 3) for maximum a posteriori (MAP) estimation. We consider a time-varying Poisson process model used by [Ihler et al., 2006] for detecting events in temporal count data. We analyze loop sensor data collected every five minutes over a span of 25 weeks from a section of a freeway near a baseball stadium in Los Angeles. For this problem, there is one observed sequence ($N = 1$) with $T = 50,400$ total observations. There are $D = 11$ parameters. Full model details are in Appendix [C.12.1].

To choose the folds in both exact CV and our ACV method, we consider two schemes, both following style (A) in Eq. (4.4); i.e., we omit observations (but not latents) in the folds. First, we follow the recommendation of Celeux and Durand [2008]; namely, we form each fold by selecting $m\%$ of measurements to omit (i.e., to form $\mathcal{O}$) uniformly at random and independently across folds. We call this scheme i.i.d. LWCV. Second, we consider a variant where we omit $m\%$ of observations in a contiguous block. We call this scheme contiguous LWCV; see Appendix [C.12.1].

In evaluating the accuracy of our approximation, we focus on a subset of $T_{\text{sub}} = 10,000$ observations, plotted in the top panel of Fig. 4.1. The six panels in the lower left of Fig. 4.1 compare our ACV estimates to exact CV. Columns range over left-out percentages $m = 2, 5, 10$ (all on the data subset); rows depict i.i.d. LWCV (upper) and contiguous CV (lower). For each of $|\mathcal{O}| = 10$ folds and for each point $x_t$ left out in each fold, we plot a red dot with the exact fold loss $-\log p(x_t | x_{[T]-\mathcal{O}}; \Theta(w_o))$ as its horizontal coordinate and our approximation $-\log p(x_t | x_{[T]-\mathcal{O}}; \Theta_{\text{IJ}}(w_o))$ as its vertical coordinate. We can see that every point lies close to the dashed black $x = y$ line; that is, the quality of our approximation is uniformly high across the thousands of points in each plot.

In the two lower right panels of Fig. 4.1, we compare the speed of exact CV to our approximation and the Newton step (NS) approximation [Beirami et al., 2017; Rad and Maleki 2020] on two data subsets (size 5,000 and 10,000) and the full data. No reported times include the initial $\Theta(1_T)$ computation since $\Theta(1_T)$ represents the unavoidable cost of the data analysis itself. I.i.d. LWCV appears in the upper plot,
and contiguous LWCV appears in the lower. For our approximation, we use 1,000 folds. Due to the prohibitive cost of both exact CV and NS, we run them for 10 folds and multiply by 100 to estimate runtime over 1,000 folds. We see that our approximation confers orders of magnitude in time savings both over exact CV and approximations based on NS. In Appendix C.9, we show that the approximations based on NS do not substantively improve upon those provided by the significantly cheaper IJ approximations.

Figure 4-1: Evaluation of approximate LWCV for time-varying Poisson processes. *(Top panel)* A subset of the count series. *(Lower left six panels)* Scatter plots comparing exact CV loss (horizontal axis) at each point in each fold (red dots) to our approximation of CV loss (vertical axis). Black dashed line shows perfect agreement. Three columns for percent points left out; two rows for i.i.d. LWCV (upper) and contiguous LWCV (lower). *(Lower right two panels)* Wall-clock time for exact and approximate CV measured on a 2.5GHz quad core Intel i7 processor with 16GB of RAM; same rows as left panels.

**Robustness to inexact optimization: Neural conditional random fields.** Next, we examine the effect of using an inexact optimum $\Theta^{(S)}$, instead of the exact initial optimum $\hat{\Theta}(1_T)$, as the input in our approximations. We consider LSCV for a bidirectional LSTM CRF (bilstm-crf) [Huang et al., 2015], which has been found [Lample et al., 2016; Ma and Hovy, 2016; Reimers and Gurevych, 2017] to perform well for named entity recognition. In this case, our problem is supervised; the words in input sentences ($x_n$) are annotated with entity labels ($z_n$), such as organizations or locations. We trained the bilstm-crf model on the CoNLL-2003 shared task benchmark [Sang and De Meulder, 2003] using the English subset of the data and the pre-defined train/validation/test splits containing $14,987(=N)/3,466/3,684$ sentence annotation pairs. Here $T$ is the number of words in a sentence; it varies
by sentence with a max of 113 and median of 9. The number of parameters \( D \) is 99. Following standard practice, we optimize the full model using stochastic gradient methods and employ early stopping by monitoring loss on the validation set. See Appendix C.12.2 for model architecture and optimization details. In our experiments, we hold the other network layers (except for the CRF layer) fixed, and report epochs for training on the CRF layer after full-model training; this procedure mimics some transfer learning methods [Huh et al., 2016].

We consider 500 LSCV folds with one sentence (i.e., one \( n \) index) per fold; the 500 points are chosen uniformly at random. The four panels in Fig. 4-2 show the behavior of our approximation (Algorithm 8 in Appendix C.7) at different training epochs during the optimization procedure. To ensure invertibility of the Hessian when far from an optimum, we add a small \( 10^{-5} \) regularizer to the diagonal. At each epoch, for each fold, we plot a red dot with the exact fold held out probability \( p(z_n \mid x_n; \hat{\Theta}(w_{\{n\}})) \) as its horizontal coordinate and our approximation \( p(z_n \mid x_n; \hat{\Theta}^{IJ}(w_{\{n\}})) \) as the vertical coordinate. Note that the LSCV loss has no dependence on other \( n \) due to the model independence across \( n \); see Appendix C.7. Even in early epochs with larger gradient norms, every point lies close to the dashed black \( x = y \) line. Fig. C-2 of Appendix C.12.2 further shows the mean absolute approximation error between the exact CV held out probability and our approximation, across all 500 folds as a function of log gradient norm and wall clock time. As expected, our approximation has higher quality at better initial fits. Nonetheless, we see that decay in performance away from the exact optimum is gradual.

**Beyond chain-structured graphs:** Crime statistics in Philadelphia. The models in our experiments above are all chain-structured. Next we consider our approximations to LWCV in a spatial model with more complex dependencies. Balocchi and Jensen [2019], Balocchi et al. [2019] have recently studied spatial models of crime in the city of Philadelphia. We here consider a (simpler) hidden MRF model for exposition: a Poisson mixture with spatial dependencies, detailed in Appendix C.12.3. Here, there is a single structure observation \( (N = 1) \); there are \( T = 384 \) census tracts in the city; and there are \( D = 2 \) parameters. The data is shown in the upper lefthand panel of Fig. 4-3.
We choose one point per fold in style (B) of LWCV here, for a total of 384 folds. We test our method across four fixed values of a hyperparameter $\beta$ that encourages adjacent tracts to be in the same latent state. For each fold, we plot a red dot comparing the exact fold loss $-\log p(x_t \mid x_{[T]-\{t\}}; \hat{\Theta}(w_{\{t\}}))$ with our approximation $-\log p(x_t \mid x_{[T]-\{t\}}; \hat{\Theta}_{II}(w_{\{t\}}))$. The results are in the lower four panels of Fig. 4-3, where we see uniformly small error across folds in our approximation. In the upper right panel of Fig. 4-3, we see that our method is orders of magnitude faster than exact CV.

![Figure 4-3: Evaluation of LWCV for loopy Markov random field. (Top left) Census tracts data. (Upper right) Wall-clock time of approximate CV and exact CV. (Lower) Scatter plots comparing CV loss (horizontal axis) at each point in each fold (red dots) to our approximation of CV loss (vertical axis). Black dashed line shows perfect agreement. Plots generated with different values of connectivity $\beta$.]

Discussion. In this work, we have demonstrated how to extend approximate cross-validation (ACV) techniques to CV tasks with non-trivial dependencies between folds. We have also demonstrated that IJ approximations can retain their usefulness even when the initial data fit is inexact. While our motivation in the latter case was formed by complex models of dependent structures, our results are also applicable to, and novel for, the classic independence framework of ACV. An interesting remaining challenge for future work is to address other sources of computational expense in structured models. For instance, even after computing $\hat{\Theta}^{\alpha}$, inference can be expensive in very large graphical models; it remains to be seen if reliable and fast approximations can be found for this operation as well.
Chapter 5

Can we globally optimize cross-validation loss?
Quasiconvexity in ridge regression

5.1 Introduction

Linear models, including LASSO and ridge regression, are widely used for data analysis across diverse applied disciplines. Linear models are often preferred since they are straightforward to apply in various senses. In particular, (1) their parameters are readily interpretable. (2) They have strong theoretical guarantees on quality. And (3) standard optimization tools are often assumed to find useful parameter and hyperparameter values. Despite their seeming simplicity, though, mysteries remain about the quality of inference in linear models. Consider cross-validation (CV) \cite{Stone1974, Allen1974}, the de facto standard for hyperparameter selection across machine learning methods \cite{Musgrave2020}. CV is an easy-to-evaluate proxy for the true out-of-sample loss. Is it a good proxy? \cite{Homrighausen2013, Chetverikov2021, Hastie2020, Patil2021} give conditions under which the global minimum of the CV loss (possibly with some mild corrections) matches the optimum of the out-of-sample loss in LASSO and ridge regression. To complete the picture, we must understand whether standard methods for minimizing the CV loss find a global minimum.

It would be easy to find a unique minimum of the CV loss if the CV loss were convex. Alas (though perhaps unsurprisingly), we show below that in essentially every case of interest the CV loss is not convex. Indeed, the usual introductory cartoon of CV loss (left panel of Fig. 5-1; see also Fig. 5.9 of \cite{Hastie2009} or Fig. 1 of \cite{Rad2020}) is not convex. But the cartoon CV loss still exhibits a single global minimum and would be easy to globally minimize with popular approaches like gradient-based methods \cite{Do2007, Maclaurin2015, Pedregosa2016, Lorraine2020} or grid search \cite{Bergstra2012, Pedregosa2011, Hsu2003}. Indeed, a more plausible possibility (which holds for the typical cartoon CV loss) is that the CV loss might be quasiconvex; i.e. its level sets are
Figure 5-1: (Left): Idealized illustration of the leave-one-out CV loss $\mathcal{L}$ (blue) and the true out-of-sample loss (black). The minimizer of each curve is marked with a vertical line of the corresponding color. (Center): CV loss for a life-expectancy prediction problem after some standard data pre-processing (Condition 2 of Section 5.2). (Right): CV loss for wine-quality prediction problem on a subset of $N = 50$ data points after standard data pre-processing (Condition 2 of Section 5.2).

convex. The benefit of quasiconvexity is that, in one dimension, any local optimum is a global optimum.

Unfortunately, this cartoon need not hold in general, even in simple models like $\ell_2$-regularized linear regression (i.e. ridge regression). Consider minimizing leave-one-out CV (LOOCV) loss as a function of the ridge regularization parameter; we denote this loss by $\mathcal{L}$. Wilson et al. [2020, Fig. 1] detail a simulated example in which $\mathcal{L}$ can be non-quasiconvex. We first demonstrate that $\mathcal{L}$ can be non-quasiconvex in real-data examples; see the middle and right panel of Fig. 5-1 which we describe in detail in Section 5.3. We next characterize which aspects of the covariate matrix and observed responses affect quasiconvexity. We prove that the norm of the responses, the scale of the singular values of the covariate matrix, and the right singular vectors of the covariate matrix all have no effect on the quasiconvexity of $\mathcal{L}$. While this result places substantial constraints on what drives the quasiconvexity of $\mathcal{L}$, we show that the quasiconvexity of $\mathcal{L}$ is unfortunately still a complex function of the remaining quantities. Our third contribution is to prove conditions under which $\mathcal{L}$ is guaranteed to be quasiconvex. In particular, we show that if (1) the covariate matrix has a singular value spectrum sufficiently close to uniform, (2) the least-squares estimator fits the training data sufficiently well, and (3) the left singular vectors of the covariate matrix are sufficiently regular, then $\mathcal{L}$ is guaranteed to be quasiconvex. While the conditions of our theory are deterministic, we show that they have natural probabilistic interpretations; as a corollary to our theory, we demonstrate that many of our conditions are satisfied either empirically or theoretically by well-specified linear regression problems with i.i.d. sub-Gaussian covariates and moderate signal-to-noise ratios. Through empirical studies, we validate the conclusions of our theory and the necessity of our assumptions.
5.2 Setup and notation

For \( n \in \{1, \ldots, N\} \), we observe covariates \( x_n \in \mathbb{R}^D \) and responses \( y_n \in \mathbb{R} \). We are interested in learning a linear model between the covariates and responses, \( \langle x_n, \theta \rangle \approx y_n \), for some parameter \( \theta \in \mathbb{R}^D \). In ridge regression, i.e. \( \ell_2 \)-regularized linear regression, we take some \( \lambda > 0 \) and estimate:

\[
\hat{\theta}(\lambda) := \arg\min_{\theta \in \mathbb{R}^D} \sum_{n=1}^{N} (\langle x_n, \theta \rangle - y_n)^2 + \frac{\lambda}{2} \|\theta\|_2^2 .
\]  
(5.1)

The regularization parameter \( \lambda \) is typically chosen by minimizing the cross-validation (CV) loss. Here we study the leave-one-out CV (LOOCV) loss:

\[
\mathcal{L}(\lambda) := \sum_{n=1}^{N} (\langle x_n, \hat{\theta}^{\backslash n}(\lambda) \rangle - y_n)^2 ,
\]  
(5.2)

where \( \hat{\theta}^{\backslash n}(\lambda) \) is the solution to Eq. (5.1) with the \( n \)th datapoint left out.

Let the covariate matrix \( X \in \mathbb{R}^{N \times D} \) be the matrix with rows \( x_n \), and let the vector \( Y \in \mathbb{R}^N \) be the vector with entries \( y_n \). We consider the low to modest-dimensional case where \( D < N \) and assume the covariate matrix \( X \) is full-rank. We further assume \( X \) and \( Y \) have undergone standard data pre-processing, as described next.

**Condition 2.** \( Y \) is zero-mean, and \( X \) has zero-mean, unit variance columns. Equivalently, where \( 1 \in \mathbb{R}^N \) is the vector of all ones, \( 1^T Y = 0 \) and \( X^T 1 = 0 \in \mathbb{R}^D \) and for all \( d = 1, \ldots, D \), \( \sum_{n=1}^{N} x_{nd}^2 = N \).

Preprocessing \( X \) and \( Y \) to satisfy Condition 2 represents standard best practice for ridge regression. First, using an unregularized bias parameter in Eq. (5.1) and setting \( Y \) to be zero-mean are equivalent; we choose to make \( Y \) zero-mean, as it simplifies our analysis below. The conditions on the covariate matrix \( X \) are important to ensure the use of \( \ell_2 \)-regularization is sensible. In particular, Eq. (5.1) penalizes all coordinates of \( \theta \) equally. If e.g. some columns of \( X \) are measured in different scales or are centered differently, this uniform penalty will be inappropriate.

5.3 LOOCV loss is typically not convex and need not be quasiconvex

If the LOOCV loss \( \mathcal{L} \) were convex or quasiconvex in \( \lambda \), then any local minimum of \( \mathcal{L} \) would be a global minimum, and we could trust gradient-based optimization methods or grid search methods to return a value near a global minimum. We next see that unfortunately \( \mathcal{L} \) is typically not convex and is often not even quasiconvex. First we show that, in essentially all cases of interest, \( \mathcal{L} \) is not convex.

**Proposition 8.** If \( \lambda = \infty \) is not a minimum of \( \mathcal{L} \), then \( \mathcal{L} \) is not a convex function.
Proof. For the sake of contradiction, assume $L$ is convex and $\lambda = \infty$ is not a minimum of $L$. This implies that there is some maximal $\lambda^* < \infty$ such that $L'(\lambda^*) = 0$. Let $\delta := L'(\lambda^* + 1)$. By convexity, $L'' \geq 0$, so we know that $\delta > 0$ and that for $\lambda \geq \lambda^* + 1$, we have $L'(\lambda) \geq \delta$. Thus for $\lambda \geq \lambda^* + 1$, we have $L(\lambda) \geq \delta(\lambda - \lambda^* - 1)$. So $\lim_{\lambda \to \infty} L(\lambda) = \infty$. However, inspection of $L$ shows $\lim_{\lambda \to \infty} L(\lambda) = \sum_{n=1}^{N} y_n^2 < \infty$, which is a contradiction.

We say that the result covers essentially all cases of interest: if $L$ continues to decrease as $\lambda \to \infty$, then there is so little signal in the data that the zero model $\theta = 0 \in \mathbb{R}^D$ is the optimal predictor according to LOOCV.

Although $L$ is generally not convex, $L(\lambda)$ might still be easy to optimize if it satisfies an appropriate generalized notion of convexity. To that end, we recall the definition of quasiconvexity.

**Definition 3.** A function $f : \mathbb{R}^p \to \mathbb{R}$ is quasiconvex if its level sets are convex.

In one dimension (i.e. $p = 1$ in Definition 3), quasiconvexity guarantees that any local optimum is a global optimum, just as convexity does. This property is often the key consideration in practical optimization algorithms. Moreover, it is not unreasonable to hope that the CV loss is quasiconvex: typical illustrations of the CV loss are not convex but are quasiconvex; see e.g. Hastie et al. [2015, Fig. 5.9], Rad and Maleki [2020, Fig. 1], or the left panel of Fig. 5-1. Illustrations of the out-of-sample loss are also typically quasiconvex; see e.g. Fig. 3.6 of Bishop [2006].

Unfortunately, we next demonstrate that the CV loss derived from real data analysis problems can be non-quasiconvex. Our first dataset contains $N = 2,938$ observations of life expectancy, along with $D = 20$ covariates such as country of origin or alcohol use; see Appendix D.1 for a full description. In this case, after pre-processing according to Condition 2, $L$ for the full dataset is quasiconvex. But now consider some additional standard data pre-processing. Practitioners often perform principal component regression (PCR) with the aim of reducing noise in the estimated $\theta$. That is, they take the singular value decomposition of $X = USV$ and produce an $N \times R$ dimensional covariate matrix $X'$ by retaining just the top $R$ singular values of $X$: $X' = U_{:,R}S_{R}$. With this pre-processing, the resulting LOOCV curve $L$ is non-quasiconvex for many values of $R$; in the center panel of Fig. 5-1 we show one example for $R = 15$.

Our second dataset consists of recorded wine quality of $N = 1,599$ red wines. The goal is to predict wine quality from $D = 11$ observed covariates relating to the chemical properties of each wine; see Appendix D.1 for a full description. We find that subsets of this dataset often exhibit non-quasiconvex $L$. In the right panel of Fig. 5-1 we show $L$ for a subset of size $N = 50$. We see that this plot contains at least two local optima, with substantially different values of $\lambda$ and substantially different values of the loss. A gradient-based algorithm initialized sufficiently far to the left would not find the global optimum, and grid search without sufficiently large values would not find the global optimum.

Now we know that $L$ can be non-quasiconvex for real data. Given the difficulty of optimizing a function with several local minima, we next seek to understand when
\( \mathcal{L} \) is quasiconvex or not.

5.4 What does the quasiconvexity of \( \mathcal{L} \) depend on?

We have seen that \( \mathcal{L} \) can be quasiconvex or non-quasiconvex, depending on the data at hand. If we could determine the quasiconvexity of \( \mathcal{L} \) from the data, we might better understand how to efficiently tune hyperparameters from the CV loss. In what follows, we start by showing that the quasiconvexity of \( \mathcal{L} \) is, in fact, independent of many aspects of the data (Proposition\textsuperscript{9}). We will see, however, that the dependence of quasiconvexity on the remaining aspects (though they are few) is complex.

A linear regression problem has a number of moving parts. The response vector \( Y \) may be an arbitrary vector in \( \mathbb{R}^N \), and the covariate matrix \( X \) can be written in terms of its singular values and left and right singular vectors. More precisely, let \( X = U \text{diag}(S)V^T \) be the singular value decomposition of the covariate matrix \( X \), where \( U \in \mathbb{R}^{N \times D} \) is an \( N \times D \) matrix with orthonormal columns, \( S \in \mathbb{R}^D \) is a vector with positive entries, and \( V \in \mathbb{R}^{D \times D} \) is an orthonormal matrix. Note we use the “compact” singular value decomposition, where \( U \) is a \( N \times D \) matrix, rather than a full \( N \times N \) matrix. With this notation in hand, we can identify aspects of the problem that do not contribute to quasiconvexity in the following result, which is proved in Appendix D.2.

Proposition 9. The quasiconvexity of \( \mathcal{L} \) is independent of

1. the matrix of right singular vectors, \( V \),
2. the norm of the responses, \( \|Y\|_2 \), and
3. the scaling of the singular values (i.e. changing \( S \) into \( S/c \) for \( c \in \mathbb{R}_{>0} \)),

in the sense that altering any of these quantities does not change whether or not \( \mathcal{L} \) is quasiconvex.

Remark 1. Assume Condition \textsuperscript{2} holds. Then by Proposition 9, without loss of generality we may (and do) assume that \( V = I_D \) and that \( Y \) is a vector on the unit \((N-1)\)-sphere.

Recall \( X \) has zero-mean columns by pre-processing the data (Condition \textsuperscript{2}). By Proposition\textsuperscript{9} we assume without loss of generality \( V = I_D \). Thus, the columns of \( X \) have zero mean when \( U^T 1 = 0 \), where \( 1 \in \mathbb{R}^N \) is the vector of all ones. Also, while Remark\textsuperscript{1} notes that we can consider \( Y \) to be on the unit \( (N-1)\)-sphere, note that the condition \( 1^T Y = 0 \) from Condition\textsuperscript{2} allows us to further constrain \( Y \) to be parameterized by a vector on the unit \( (N-2)\)-sphere. Hence the quasiconvexity of \( \mathcal{L} \) depends on three quantities: (1) the matrix of left singular vectors, \( U \), an orthonormal matrix with \( 1 \in \mathbb{R}^N \) in its left null-space, (2) the (normalized) vector \( Y \) which sits on the unit \( (N-2)\)-sphere, and (3) the (normalized) singular values.

Now that we know the quasiconvexity of \( \mathcal{L} \) depends on only three quantities, we might hope that quasiconvexity would be a simple function of the three. To
Figure 5-2: How does quasiconvexity depend on $Y$ and $U$, for data with $N = 3$ and $D = 2$? The left, center, and right panels each correspond to a different setting of the singular values $S$. We divide the unit circle for each of $U$ and $Y$ into 225 equally spaced points. We examine the severity of non-quasiconvexity in $L$ over this $225 \times 225$ grid. Each grid point is colored by Eq. (5.3), where dark purple corresponds to quasiconvexity (no local minima).

To investigate this dependence, we consider the case of $N = 3$ and $D = 2$, since this case is particularly easy to visualize. To see why it is easy to visualize, first note that $Y$ is a three-dimensional vector; thus by our discussion above, we can parameterize $Y$ by a vector on the unit circle (i.e. a scalar between 0 and $2\pi$). Second, note that the matrix of left singular vectors $U$ is parameterized by two orthonormal vectors, $U_1$ and $U_2$, each on the unit 2-sphere. As both vectors must be orthogonal to $1 \in \mathbb{R}^3$, we can parameterize $U_1$ and $U_2$ by two orthonormal vectors on the unit circle. We parametrize $U_1$ by a scalar that determines $U_2$ up to a rotation of $U_2$ by $\pi$. We fix a rotation for $U_2$ relative to $U_1$ so that, for fixed singular values $S$, the quasiconvexity of $L$ is parameterized by two scalars.

Fig. 5-2 is the resulting visualization. Precisely, to make Fig. 5-2, we fix an orientation between $U_1$ and $U_2$: here, a rotation of $\pi/2$ on the unit circle. We create a uniform grid over the unit circles for $U_1$ and $Y$. Fig. 5-2 visualizes the severity of non-quasiconvexity of $L$ as we move over this grid for three different settings of the singular values. To define the severity of non-quasiconvexity, let $\lambda_{\text{worst}}$, $\lambda^*$, and $\lambda_{\text{worst-min}}$ correspond to the $\lambda$ maximizing $L$, the $\lambda$ minimizing $L$, and the $\lambda$ corresponding to the local minimum with largest $L$, respectively. We then compute

$$\text{severity} := \frac{L(\lambda_{\text{worst-min}}) - L(\lambda^*)}{L(\lambda_{\text{worst}}) - L(\lambda^*)}$$

(5.3)

If the severity is near 1, then finding the worst local minimum is nearly as bad as selecting the worst possible $\lambda$. In Fig. 5-2, we color pixels according to this measure of severity and do in fact see values near 1. We see that, in general, $U$, $Y$, and $S$ have a complicated interaction to determine the quasiconvexity of $L$. The values of $Y$ that produce quasiconvexity depend on $U$. No setting of $Y$ or $U$ guarantees quasiconvexity.

Fig. 5-2 demonstrates the poor performance of a hypothetical optimizer that always finds the worst local minimum. In Appendix D.3, we show that poor performance is not just hypothetical; in particular, the most commonly used methods for...
optimizing $\mathcal{L}$ – grid search and gradient descent – encounter difficulties due to non-quasiconvexity. Issues due to non-quasiconvexity are also not specific to measuring performance in terms of $\mathcal{L}$ nor to optimizing the LOOCV loss. In Appendix D.3, we show that the $K$-fold CV loss also suffers from non-quasiconvexity and that computing Eq. (5.3) with the test loss instead of $\mathcal{L}$ also exhibits issues due to non-quasiconvexity.

Despite the complexity of Fig. 5-2, one trend does seem clear: as the singular values become more similar (moving from left to right in the panels of Fig. 5-2), the fraction of $Y$ values and $U$ values that correspond to quasiconvexity (dark purple regions) grows. Based on this behavior, one might conjecture that a sufficiently uniform spectrum of the covariate matrix could guarantee the quasiconvexity of $\mathcal{L}$.

### 5.5 Quasiconvexity of $\mathcal{L}$ with a nearly uniform spectrum $S$

We now build on the conjecture of the previous section to show that we can, in fact, guarantee quasiconvexity in certain cases. In particular, we will show conditions under which a sufficiently uniform spectrum $S$ of the covariate matrix $X$ guarantees that $\mathcal{L}$ is quasiconvex.

One might hope that for large $N$, eventually any $Y$ or $U$ would yield quasiconvexity. However, even when $S$ is exactly uniform, our experiments in Section 5.6 show that we cannot expect such a statement. Rather, we will devise conditions on $Y$ and $U$ to avoid “extreme” settings for either quantity. With these conditions, our main theorem will show that a sufficiently flat spectrum $S$ does indeed guarantee quasiconvexity of $\mathcal{L}$. When our theorem applies, we can safely terminate an optimization procedure at the first local minimum of $\mathcal{L}$ that we discover.

We first establish notation and then state our assumptions.

**Definition 4.** Let $\hat{\theta} := (X^TX)^{-1}X^TY$ be the least-squares estimate. Define the least-squares residuals $\hat{E} := Y - X\hat{\theta}$. Let $\hat{e}_n$ be the $n$th entry of $\hat{E}$.

Note that $\hat{\theta}$ is well-defined since we have assumed that $D < N$ and that $X$ is full-rank. For the tractability of our theory, all of our assumptions and conclusions will be asymptotic in $N$; in particular, our assumptions will use big-O and little-o statements, which are to be taken with respect to $N$ growing large. In our discussion of these assumptions, we assume that $D$ is fixed. There is nothing in our proofs or assumptions that requires $D$ to be fixed; however the validity of our assumptions is not clear if $D$ grows with $N$, so we do not consider this case here. Since LOOCV is useful precisely for finite $N$, we are careful to show in our experiments (Section 5.6) that these asymptotics take hold for small $N$.

Our first assumption concerns the magnitude of the residuals $\hat{E}$.

**Assumption 12.** $(1/N)\sum_{n=1}^{N} \hat{e}_n^2$ is $O(1)$ (i.e. it does not grow with $N$).

This assumption is fairly lax. For example, suppose our linear model is well-specified. In particular, suppose there exists some $\theta^* \in \mathbb{R}^D$ such that $y_n = \langle x_n, \theta^* \rangle + \varepsilon_n$.
where the $\varepsilon_n$ are i.i.d. $\mathcal{N}(0, \sigma^2)$ for some $\sigma > 0$. Stack the $\varepsilon_n$ into a vector $E \in \mathbb{R}^N$. Then $\|\hat{E}\|^2 = \|(I_N - UU^T)E\|^2 < \|E\|^2$. Since $(1/N)\|E\|^2$ is $O(1)$ with high probability, it follows that Assumption 12 holds with high probability in this well-specified linear model. We emphasize that Assumption 12 depends on the residuals of the least squares estimate, not (directly) on the noise in the observations.

Our next assumption governs the size of the least squares estimate $\hat{\theta}$.

**Assumption 13.** $\|\hat{\theta}\|$ is $O(1)$ (i.e. it does not grow with $N$).

Again, this is a lax assumption. For example, given any statistical model for which $\hat{\theta}$ is a consistent estimator for some quantity, Assumption 13 holds.

Our next assumption constrains the uniformity of the left-singular value matrix $U$ with rows $u_n \in \mathbb{R}^D$.

**Assumption 14.** We have $\max_n \|u_n\|^2 := \|u_{\max}\|^2 = O(N^{-p})$ for some $p > 1/2$.

Assumption 14 is an assumption about the coherence of the $U$ matrix, a quantity of importance in compressed sensing and matrix completion [Candès and Recht, 2009]. In particular, Assumption 14 requires that the coherence of $U$ decay sufficiently fast as a function of $N$. Suppose we remove the condition that $U$ have zero-mean columns (see Condition 2 and the discussion after Remark 1) and assume a uniform distribution over valid $U$ (i.e. matrices with orthonormal columns); then Assumption 14 is known to hold with high probability for any $p \in (1/2, 1)$ [Candès and Recht, 2009, Lemma 2.2].

There do exist matrices $U$ with orthonormal zero-mean columns that do not satisfy Assumption 14. For instance, take some small $N_0$ (say $N_0 = 5$) and a valid $U'$ for this $N_0$. Then, for $N > N_0$, form $U$ by appending $0 \in \mathbb{R}^{(N-N_0)\times D}$ to the bottom of $U'$. This construction yields an $N \times D$ matrix $U$ with orthonormal and zero-mean columns for which $\|u_{\max}\|^2$ is constant as $N$ grows. Still, in our experiments in Section 5.6 and Appendix 2.4 we confirm that, for a uniform distribution over orthonormal $U$ with zero-mean columns, Assumption 14 holds with high probability.

Our final assumption is a technical assumption relating $\|u_n\|^2$, $\hat{\varepsilon}_n$, and $\hat{\theta}$.

**Assumption 15.** The following quantity is positive and $\Theta(1)$ (i.e. is bounded away from zero and does not grow with $N$): $\|\hat{\theta}\|^2 - \sum_{n=1}^N \|u_n\|^2 \left(\langle u_n, \hat{\theta} \rangle^2 + 2\hat{\varepsilon}_n^2\right)$

Roughly, this assumption means that the largest $\|u_n\|^2$ and $\hat{\varepsilon}_n^2$ values do not occur for the same values of $n$. To see this relation, note that Assumption 14 implies $\|\hat{\theta}\|^2 - \sum_{n} \|u_n\|^2 \langle u_n, \hat{\theta} \rangle^2 \geq (1 - O(N^{-p}))\|\hat{\theta}\|^2$. If we assume that $\|\hat{\theta}\|^2 = \Theta(1)$ (i.e. Assumption 13 holds and $\hat{\theta}$ does not converge to $0 \in \mathbb{R}^D$), then we find $\|\hat{\theta}\|^2 - \sum_{n} \|u_n\|^2 \langle u_n, \hat{\theta} \rangle^2 = \Theta(1)$. So, we need only that $\sum_{n} \|u_n\|^2 \hat{\varepsilon}_n^2 = o(1)$ for Assumption 15 to hold; e.g. we need that the largest values of $\|u_n\|^2$ and the largest values of $\hat{\varepsilon}_n^2$ typically do not occur for the same values of $n$.

With our assumptions in hand, we can now state our main theorem. Our theorem relates the uniformity of the spectrum of $X$ to the quasiconvexity of $\mathcal{L}$. As we have shown in Proposition 9, the scaling of the singular values does not matter for the quasiconvexity of $\mathcal{L}$. We therefore take the spectrum to be nearly uniform around $1 \in \mathbb{R}^D$. 

68
Theorem 5. Consider a series of regression problems with \( N \) growing to infinity, where the \( N \)th problem uses data \((X^{(N)}, Y^{(N)})\). Assume this sequence satisfies Assumptions [12] to [15]. Let the covariate matrix of the \( N \)th regression problem have SVD \( X^{(N)} = U^{(N)} \text{diag}(S^{(N)}) V^{(N)} \). There is a \( N_0 > 0 \) and neighborhood \( \Delta \) of \( 1 \in \mathbb{R}^D \) such that if \( N \geq N_0 \) and the spectrum \( S^{(N)} \in \Delta \), then \( \mathcal{L} \) is quasiconvex.

Proof sketch: For one-dimensional functions \( \mathcal{L} \), a sufficient condition for quasiconvexity is that for all \( \lambda \) such that \( \mathcal{L}'(\lambda) = 0 \), we have \( \mathcal{L}''(\lambda) > 0 \) [Boyd and Vandenberghe, 2009, Chapter 3.4]. We first show \( \mathcal{L}' \) can be zero only for a bounded set of \( \lambda \). We then show that for any \( \lambda \) within this set with \( \mathcal{L}'(\lambda) = 0 \), we have \( \mathcal{L}''(\lambda) > 0 \). See Appendix D.5 for a full proof.

In Section 5.4, we showed it can be difficult to guess when \( \mathcal{L} \) is quasiconvex. But Theorem 5 yields one condition that guarantees \( \mathcal{L} \) is quasiconvex: when \( X \) has a nearly uniform spectrum. A natural question then is: when is the spectrum of \( X \) nearly uniform? As it happens, a uniform spectrum occurs under standard assumptions, for example, when the \( x_{nd} \) are i.i.d. sub-Gaussian random variables.

Definition 5 (e.g. Vershynin, 2018). A random variable \( Q \) is sub-Gaussian if there exists a constant \( c > 0 \) such that \( \mathbb{E}[\exp(\frac{Q^2}{c^2})] \leq 2 \).

Corollary 3. Take any series of regression problems satisfying Assumptions [14] and [15]. Assume the series of regression problems are drawn from a well-specified linear model for some \( \theta^* \in \mathbb{R}^D \): \( y^{(N)}_n = \langle x^{(N)}_n, \theta^* \rangle + \varepsilon_n \), where \( \varepsilon_n \text{ i.i.d. } \mathcal{N}(0, \sigma^2) \). If \( \sigma \) is sufficiently small, \( \hat{\theta} \) is consistent for \( \theta^* \), and the entries of the covariate matrices \( x^{(N)}_{nd} \) are i.i.d. sub-Gaussian random variables, then \( \mathcal{L} \) is quasiconvex with probability tending to 1 as \( N \to \infty \).

Proof sketch: Assumptions [12] and [13] hold for a well-specified linear model. If the entries of \( X \) are i.i.d. sub-Gaussian random variables, standard concentration inequalities imply that its spectrum is nearly uniform with high probability; hence the result of Theorem 5 applies. See Appendix D.6 for a full proof.

5.6 Theorem 5 in practice

In Section 5.5, we established a number of assumptions that we then required in Theorem 5 to prove that \( \mathcal{L} \) is quasiconvex. A few questions remain about our theorem in practice: (1) how large is the neighborhood \( \Delta \), (2) how necessary are our assumptions, and (3) how large do we require \( N \) to be? We explicitly answer (1) and (2) with experiments below. (3) is particularly concerning, as regularization has minimal impact when \( N \gg D \). That is, there is little performance gain by using a regularizer, which removes the need for hyperparameter tuning. To show that our theorem holds when \( N \sim D \), the majority of our experiments validating Theorem 5 use \( N \) that is at most an order of magnitude larger than \( D \).

Throughout our experiments, we check for non-quasiconvexity numerically and use a shortcut formula to compute \( \mathcal{L} \) that takes advantage of the fact that the right
Figure 5-3: (Upper left): We generate many datasets and plot the fraction that are non-quasiconvex ("non-QVX"), varying $N$ and the distance of the spectrum from uniformity ($\|S - 1\|_1$). (Upper right): We generate two sets (orange, blue) of left-singular vector matrices $U$. In the blue case, we check that the maximum of $\log \|u_{\max}\|^2$ across all $U$ for a particular $N$ decreases roughly linearly on a log-log plot (i.e. the blue set satisfies Assumption 14). In the orange case, we check that the minimum of $\log \|u_{\max}\|^2$ across all $U$ for a particular $N$ is roughly constant (i.e. the orange set does not satisfy Assumption 14). (Lower): For all the $U$ matrices from the upper right plot, we generate many datasets and plot the fraction that are not quasiconvex.
singular vectors of $X$ are $V = I_D$; see Appendix D.9 for details. The only software dependency for our experiments is NumPy [Harris et al., 2020], which uses the BSD 3-Clause “New” or “Revised” License.

**How do we know $S$ is in the neighborhood $\Delta$ in Theorem 5?** While our theorem does not give an explicit size of the neighborhood $\Delta$, we can show empirically that $\Delta$ is substantial, even for small to moderate $N$. We fix $D = 5$. To generate various spectra of $X$, we set $S_d = e^{\alpha_d} e^{\alpha D}$. For $\alpha \to 0$, we get $S \to 1$; we vary $\alpha$ from zero to one to generate spectra of varying distances from uniformity. For each $\alpha$, we sample 100 left-singular-value matrices $U$ from the uniform distribution over orthonormal $U$ with column means equal to 0; see Appendix D.7 for how to generate such matrices. We fix a unit-norm $\theta^* \in \mathbb{R}^D$ and for each $U$, we generate data from a well-specified linear model, $y_n = \langle x_n, \theta^* \rangle + \varepsilon_n$, where the $\varepsilon_n$ are drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$ with variance $\sigma^2 = 0.5$.

5. In particular, for each setting of $U$, we generate 100 vectors $Y$. For each setting of $U$ and $Y$, we compute $L$ and check whether it is quasiconvex. In the top left panel of Fig. 5-3, we report the fraction of problems (out of the 100 $\times$ 100 = 10,000 datasets for the corresponding $\alpha$ value) with a non-quasiconvex $L$ versus the distance from uniformity, $\|S - 1\|_1$. We see that, even for $N = 20$, the fraction of non-quasiconvex problems quickly hits zero as $\|S - 1\|_1$ shrinks.

To provide a rough practical heuristic, we observe from Fig. 5-3 that when $N = 50$ and $\|S - 1\|_1 \sim 2$, non-quasiconvexity occurs less than 1% of the time. In Fig. 5-3, $\Delta$ appears to grow slightly with $N$, so Fig. 5-3 seems to suggest that we should expect to see little quasiconvexity if $\|S - 1\|_1 \leq 2$ and $N \geq 50$. In practice, how should we access the spectrum to check this condition? When $N$ and $D$ are small enough we can directly compute $S$ via the singular value decomposition; however, in practice, $N$ or $D$ may be large. If $N$ is large and $D$ is small, we can access the spectrum as the square root of the eigenvalues of $X^T X$. If $N$ is *very* large, formation of $X^T X$ can be expensive; in this case, we suggest the use of randomized sketching to obtain a randomized approximation to the spectrum of $X$ [Woodruff et al., 2014]. Finally, when both $N$ and $D$ are large, we can use spectral density estimation, which gives an estimate of the density of a matrix’s eigenvalues [Lin et al., 2016], and has been shown to successfully scale to large problems [Ghorbani et al., 2019, Yao et al., 2020].

**Importance of Assumption 14.** We now establish the necessity of Assumption 14 on the decay of $\|u_{\text{max}}\|^2$ with $N$. To do so, we generate two sets of matrices $U$ as $N$ grows. We generate the first set to satisfy Assumption 14 and we generate the second to violate Assumption 14. In both cases, we will take $D = 5$ and ten settings of $N$ between $N = 10$ and $N = 300$.

To generate the assumption-satisfying matrices $U$, we proceed as follows. For each $N$, we draw 500 matrices $U$ from the uniform distribution over orthonormal $U$ matrices with column means equal to 0. For each $N$, we plot the maximum value of $\|u_{\text{max}}\|^2$ across these 500 $U$ matrices in Fig. 5-3 (top-right) as a blue dot. We fit a line to these values on a log-log plot, and find the slope is -0.74. This confirms that these matrices satisfy Assumption 14.

To generate assumption-violating matrices $U$, we proceed as follows. Recall that the smallest $N$ is 10 and $D = 5$. 100 times, we randomly draw a $U_{\text{small}} \in \mathbb{R}^{8 \times 5}$. We
Figure 5-4: Checking Assumption 12. (Left): For each $N$ and each $\nu = \|\hat{E}\|$, we generate many datasets and check if there is any non-quasiconvex (“non-QVX”) $\mathcal{L}$. We plot the largest $\nu$ for which we find only quasiconvex $\mathcal{L}$. The growth is roughly linear, which suggests Assumption 12 cannot be loosened. (Right): For each $N$ and each $\nu = \|\hat{E}\|$, we generate many data sets and plot the fraction of $\mathcal{L}$ that are non-quasiconvex. Vertical lines show $\nu_{\text{max}}$ for each $N$.

then construct each $U$ by appending $N - 8 \times D$ zeros to $U_{\text{small}}$. For each $N$, we plot the minimum value of $\|u_{\text{max}}\|^2$ across these 100 $U$ matrices in Fig. 5-3 (top-right) as an orange dot. Since the minimum of $\|u_{\text{max}}\|^2$ is constant with $N$, Assumption 14 is violated.

Now we check quasi-convexity. To that end, we randomly select a fixed unit-norm vector $\theta^* \in \mathbb{R}^D$. For each $N$, we generate 100 noise vectors $E \in \mathbb{R}^N$, where the entries $E_n$ are drawn i.i.d. from $\mathcal{N}(0, 0.5)$. For each $U$ and $E$, we construct $Y = US\theta^* + E$, where $S = 1 \in \mathbb{R}^D$. We then compute the fraction of these $(100 \times 100 = 10,000)$ losses $\mathcal{L}$ that are non-quasiconvex. In the lower panel of Fig. 5-3, we plot this fraction against $N$ for both for the assumption-satisfying case (blue) and the assumption-violating case (orange) in Assumption 14. When the assumption is satisfied (blue), we see that the conclusion of Theorem 5 holds: beyond a certain $N$, there are no settings of $U$ or $Y$ that generate a non-quasiconvex $\mathcal{L}$. We see that in practice, the boundary $N$ is small or moderate (below 50). When the assumption is violated (orange dots), we see that the conclusion of Theorem 5 fails to hold: as $N$ grows, there are still settings of $U$ and $Y$ for which quasiconvexity fails to hold. Finally, we call attention to the vertical axis. Even in the assumption-violating case, the fraction of non-quasiconvex losses is small. It follows that, even for our degenerate $U$’s, nearly every combination of noise and $U$ leads to a quasiconvex $\mathcal{L}$. An interesting challenge for future work is to provide a precise characterization of this effect.

**Do the $\hat{e}_n$ need to be small (Assumption 12)?** Finally, we demonstrate the necessity of Assumption 12 which can be restated as requiring that $\|\hat{E}\|^2$ grows at most linearly in $N$. But we also find a suggestion that there may be even more permissive assumptions of interest. To this end, we vary $N$ from 10 to 30. For each $N$, we generate 4,000 settings of $U$, each uniform over orthonormal $U$ with column means equal to 0. For each $U$, we generate 250 unit vectors $R$ such that $U^TR = 0$ (each generated uniformly over such $R$; see Appendix D.7). Separately, we consider 60 different norms $\nu$ for the vector $\hat{E}$ equally spaced between $\nu = 0$ and $\nu = 2$; these
are the same across \( N \). We generate a single unit-norm \( \theta^* \in \mathbb{R}^D \).

For each setting of \( U, R, \) and \( \nu \), we consider the regression problem with covariate matrix \( U_1 \) and responses \( Y = U_1 \theta^* + \hat{E} \), where \( \hat{E} := \nu R \). We record whether \( \mathcal{L} \) is quasiconvex or not for this problem. For a particular \( N \) and particular error-norm \( \nu \), we check whether any of the \( \mathcal{L} \) (across \( 4,000 \times 250 = 1,000,000 \) problems) were non-quasiconvex. Finally, for each \( N \), we find the maximum error-norm \( \nu_{\text{max}, N} \) such that for all \( \nu < \nu_{\text{max}, N} \) every regression problem is quasiconvex. We plot a dot at \( (N, \nu_{\text{max}, N}) \) in the left panel of Fig. 5-4. We see that in fact the boundary of allowable \( \hat{E} \) norms does grow about linearly in \( N \).

Our next plot lends additional insight into how the boundary \( \nu_{\text{max}, N} \) varies with \( N \) and is also suggestive of other potential variations on Assumption 12 that might be of interest. In particular, in the right panel of Fig. 5-4, we consider two particular values of \( N \): \( N = 10 \) (blue) and \( N = 20 \) (red). For each setting of \( \nu \) on the horizontal axis, we compute the fraction of non-quasiconvex losses \( \mathcal{L} \) over all settings of \( U \) and \( R \) (\( 4,000 \times 250 = 1,000,000 \) problems for each \( \nu \)). We see that, as expected from the left panel of Fig. 5-4, the boundary \( \nu_{\text{max}, N} \) is higher for \( N = 20 \) than for \( N = 10 \). Surprisingly, we also see that at high values of \( \nu \), the fraction of non-quasiconvex cases decreases again. We conjecture that in general (i.e. beyond these two particular \( N \)), large amounts of noise leads to little or no non-quasiconvexity. Finally, we note that, as in the bottom panel of Fig. 5-3, the fraction of non-quasiconvex cases across all \( \nu \) is low. Again, this small fraction suggests a direction for future work.

### 5.7 Discussion

We have shown that the LOOCV loss \( \mathcal{L} \) for ridge regression can be non-quasiconvex in real-data problems. Local optima need not be global optima. These multiple local optima may pose a practical problem for common hyperparameter tuning methods like gradient-based optimizers, which may get stuck in a local optimum, and grid search, for which upper and lower bounds need to be set.

We proved that the quasiconvexity of \( \mathcal{L} \) is determined by only a few aspects of a linear regression problem. But we also showed that the quasiconvexity of \( \mathcal{L} \) is still a complicated function of the remaining quantities, and as of this writing the nature of this function is far from fully understood. Nonetheless, we have provided theory that guarantees at least some useful cases when \( \mathcal{L} \) is quasiconvex: when the spectrum of the covariate matrix is sufficiently flat, the least-squares fit \( \hat{\theta} \) fits the data reasonably well, and the left singular vectors of the covariate matrix are regular. In our experiments, we have confirmed that these assumptions are necessary to some extent: when they are not satisfied, \( \mathcal{L} \) can be non-quasiconvex. Still, our empirical results make it clear there is more to be explored. We describe some of the directions we believe are most interesting for future work below.

**Sharper characterization of when \( \mathcal{L} \) is quasiconvex.** Fig. 5-2 shows that non-quasiconvexity disappears as the spectrum of \( X \) becomes uniform; however, it is clear that there is very regular behavior to the pattern of quasiconvexity even when the singular values of \( X \) are non-uniform. We are not able to characterize these
patterns at this time but believe these patterns pose a fascinating challenge for future work. Relatedly, our experiments (Section 5.6) show that when our assumptions are violated, quasiconvexity of $L$ is not guaranteed. However, we have observed that even when $L$ is not guaranteed to be quasiconvex, many settings of $U$ and $Y$ still give quasiconvexity. In many of our experiments, the fraction of non-quasiconvex losses $L$ was extremely small.

**How many local optima and how bad are they?** Without the guarantee of a single, global optimum, it is not clear that we can ever know that we have globally (near-)optimized $L$. However, notice that our examples in Fig. 5-1 all have at most two local optima. In simulated experiments, we also typically encountered two local optima in non-quasiconvex losses, although we have not studied this behavior systematically. If $L$ were guaranteed to have only two or some small number of optima, optimization might again be straightforward, even in the case of non-quasiconvexity; an algorithm could search until it finds the requisite number of optima and then report the one with the smallest value of $L$. Alternatively, one might hope that all local optima have CV loss (and ideally out-of-sample error) close in value to that of the global optimum. Indeed, Kawaguchi [2016] argue that this property holds for certain losses in deep learning. Presumably it is not universally the case that local optima exhibit similar loss since the right panel of Fig. 5-1 seems to give a counterexample. But it might be widely true, or true under mild conditions. Meanwhile, in the absence of such guarantees, optimization of $L$ should proceed with caution.

**Beyond ridge regression.** We have shown – in our opinion – surprising non-quasiconvexity for the LOOCV loss for ridge regression. Do similar results hold for simple models outside ridge regression? The regularization parameter in other $\ell_2$ or $\ell_1$-regularized generalized linear models is often tuned by minimizing a cross-validation loss. In preliminary experiments, we have found non-quasiconvexity in $\ell_2$-regularized logistic regression. To what extent do empirical results like those in Fig. 5-2 or theoretical results like those in Theorem 5 hold for other models and regularizers?
Chapter 6

Measuring the sensitivity of Gaussian processes to kernel choice

6.1 Introduction

Gaussian processes (GPs) enable practitioners to estimate flexible functional relationships between predictors and outcomes. GPs have been used to monitor physiological vital signs in hospital patients [e.g. Cheng et al., 2020; Colopy et al., 2016; Futoma et al., 2017a,b], to estimate the health effects of exposure to airborne pollutants [e.g. Ferrari and Dunson, 2020; Lee et al., 2017; Ren et al., 2021], and in many other medical and scientific settings. To use a GP for any application, a practitioner must choose a covariance kernel. The kernel determines the shape, smoothness, and other properties of the latent function of interest [Duvenaud, 2014, Chap. 2]. Ideally a user would specify a kernel that exactly encodes all of their prior beliefs about the latent function. In practice, a user often has only vague qualitative prior information and typically selects a kernel from a relatively small set of commonly used families. It seems plausible that other kernels could have been equally compatible with the user’s beliefs. When a user has no reason to prefer one kernel over another given their prior beliefs, we call the kernels qualitatively interchangeable. We would worry if substantive medical or scientific decisions changed when using a qualitatively interchangeable kernel. In this chapter, we propose a workflow to assess the sensitivity of the GP posterior under qualitatively interchangeable choices of the kernel. Fig. 6-1 situates our work, an example of model criticism, in the modeling workflow.

Related work. Robustness and sensitivity of data analyses to data and model choice have been studied for decades [Andrews et al., 1972; Huber and Ronchetti, 2009; Goodfellow et al., 2015]. In the context of Bayesian methods, sensitivity to the choice of prior has been studied as well [Berger et al., 1994; Gustafson, 1996; Berger, 2000; Giordano et al., 2021]. These works assess sensitivity by varying the prior within a small epsilon-ball around the user-specified prior with the intuition that a small ball will mostly contain priors that are acceptable alternatives to the user-specified prior. In contrast, we note that the class of qualitatively interchangeable kernels is actually the class of alternative priors of interest; we explicitly define and study sensitivity
Our focus on robustness to kernel choice stands in contrast to existing work on robustness in GPs, which focus on robustness to data perturbations [Kim and Ghahramani, 2008, Hernández-Lobato et al., 2011, Jylänki et al., 2011, Ranjan et al., 2016, Bogunovic et al., 2018, Cardelli et al., 2019]. Our focus is also distinct from that of works studying convergence rates [van der Vaart and van Zanten, 2011, Teckentrup, 2020, Wang and Jing, 2021, Wynne et al., 2021] and asymptotic predictive equivalence [Stein, 1993, Bevilacqua et al., 2019, Kirchner and Bolin, 2021] for GP regression with misspecified kernels. These works do not examine how kernel choice affects non-linear functionals like posterior variances, do not study what happens in finite samples, and do not consider kernel choice as an issue of prior specification as we do. Automatic kernel discovery procedures (e.g. Benton et al. [2019], Duvenaud et al. [2013], Wilson and Adams [2013], Wilson et al. [2016]) ostensibly obviate the need for careful kernel specification. But careful users will ensure discovered kernels match their prior beliefs about the data at hand, so prior sensitivity is still relevant in such settings.

Our contributions. We propose the first workflow to discover whether applied decisions based on a GP posterior are sensitive to the user-specified prior (i.e. the kernel). Our workflow proceeds as follows. (A) Keep expanding a class of appropriate kernels around the original kernel until some kernel in this class yields a substantially different decision. (B) Assess if this decision-changing kernel is qualitatively interchangeable with the original kernel. If the two kernels are interchangeable, we conclude the decision is not robust; a different decision could be reached with the same prior information. If the two kernels are not interchangeable, we cannot conclude non-robustness. We provide a practically viable implementation of steps (A) and (B). We demonstrate the practical utility of our workflow by discovering non-robustness in various applied applications of Gaussian processes: (1) predicting whether a hospital patient’s heart rate is alarmingly high, (2) predicting future carbon dioxide levels, and (3) classifying MNIST handwritten digits.

6.2 Our Workflow

Setup and notation. Consider data $D = \{(x_n, y_n)\}_{n=1}^{N}$, with covariates $x_n \in \mathbb{R}^D$ and outcomes $y_n \in \mathbb{R}$. Take a regression model $y_n \sim \mathcal{N}(f(x_n), \sigma^2)$ with $\sigma^2 > 0$ and a Gaussian process (GP) prior on $f$; we parameterize the GP with mean zero and covariance kernel $k$: $f \sim \mathcal{GP}(0, k)$. Typical $k$ depend on hyperparameters $\theta$ (including $\sigma^2$); unless stated otherwise, we assume that $\theta$ is estimated using maximal marginal likelihood estimation (MMLE). That is, $\hat{\theta} = \arg\max_{\bar{\theta}} p(y_1, \ldots, y_N \mid x_1, \ldots, x_N, \bar{\theta})$. Let $k_0$ be the practitioner-chosen kernel with MMLE hyperparameters. Let $F^*(k)$ be any scalar functional of the posterior $f \mid D$ that is a differentiable function of the kernel $k$. For example, $F^*$ might be the variance at a test point $x^*$ or an order statistic of the means at a set of test points $\{x^*_t\}_{t=1}^{T}$ (Section 6.4). Let the level $L \in \mathbb{R}$ represent a decision threshold in $F^*(k)$. That is, we make one decision when $F^*(k) \geq L$ and a different one when $F^*(k) < L$. For example, let time be a single covariate, and let outcome be the resting heart rate of a hospital patient. $F^*(k)$ could...
be the 95th percentile of the GP posterior at a particular time; an alarm might trigger if \( F^*(k) \) is greater than \( L = 130 \) bpm but not otherwise [Fidler et al., 2017].

We want to assess whether our decision would change if we used a different qualitatively interchangeable kernel. Without loss of generality, we assume that \( F^*(k_0) < L \). Then we can define non-robustness.

**Definition 6.** For original kernel \( k_0 \), we say that our decision \( F^*(k_0) < L \) is non-robust to the choice of kernel if there exists a kernel \( k_1 \) that is qualitatively interchangeable with \( k_0 \) and \( F^*(k_1) \geq L \).

**Workflow overview.** Our workflow is summarized in Algorithm 4. We start by defining a set \( K_\varepsilon \) of kernels that are “\( \varepsilon \)-near” \( k_0 \). We then solve the optimization problem:

\[
\begin{align*}
    k_1(\varepsilon) &:= \arg \max_{k \in K_\varepsilon} F^*(k) \\
    \varepsilon^* &\text{ is smallest } \varepsilon \text{ s.t. } F^*(k_1(\varepsilon)) \geq L
\end{align*}
\]  

(6.1)

To find \( \varepsilon^* \), we slowly increase \( \varepsilon \) until \( k_1(\varepsilon) \) changes our decision. We then check whether the decision-changing kernel, \( k_1(\varepsilon^*) \), is qualitatively interchangeable with \( k_0 \). It remains to precisely define a set of “\( \varepsilon \)-near” kernels and show that we can efficiently solve Eq. (6.1) (Section 6.2.1), and to provide ways to assess qualitative interchangeability (Section 6.2.2).

Note that although Algorithm 4 can detect non-robustness, it cannot certify robustness; it is possible, even though it may be unlikely, that there exists a qualitatively
interchangeable kernel that the methodology has not detected but that still changes
the decision. This point is generally true of sensitivity analyses, and the present
workflow is no exception. This observation is similar in spirit to classical hypothesis
tests: a user can reject – but not accept – a null hypothesis.

**Algorithm 4** Workflow for assessing robustness of GP inferences to kernel choice

1. Choose initial kernel \( k_0 \) using prior information.
2. Choose posterior quantity of interest \( F^* \). \( \triangleright \) E.g. Posterior mean at test point \( x^* \)
3. Define decision threshold \( L \). \( \triangleright \) E.g. 130 bpm is an alarming resting heart rate
4. Define “\( \varepsilon \)-near” kernels \( \mathcal{K}_\varepsilon \), for \( \varepsilon > 0 \) \( \triangleright \) Section 6.2.1
5. Solve Eq. (6.1) to get \( k_1(\varepsilon^*) \) \( \triangleright \) Section 6.2.1
6. Assess if \( k_0 \) and \( k_1(\varepsilon^*) \) are qualitatively interchangeable. \( \triangleright \) Section 6.2.2
7. **if** \( k_0 \) and \( k_1(\varepsilon^*) \) qualitatively interchangeable **then return** “\( F^* \) is non-robust to
the choice of kernel.”
8. **else return** “Did not find that \( F^* \) is non-robust to the choice of kernel.”
9. **end if**

### 6.2.1 Nearby kernels and efficient optimization

We give two practical examples of how to choose \( \mathcal{K}_\varepsilon \) in the present work and detail
how to solve Eq. (6.1) in each case. First, we consider the case where we assume
\( k \in \mathcal{K}_\varepsilon \) should be stationary. Second, we allow non-stationary kernels \( k \in \mathcal{K}_\varepsilon \).

**Stationary kernels.** By Bochner’s theorem, every stationary kernel can be rep-
resented by a positive measure \([\text{Rasmussen and Williams, 2006}, \text{Thm. 4.1}]\). In the
kernel discovery literature, it is common to make the additional assumption that this
measure has a density \( S(\omega) = \int e^{-2\pi i \tau \omega} k(\tau) d\tau \), where \( \tau = x - x' \)[\text{Wilson and Adams, 2013, Benton et al., 2019, Wilson et al., 2016}]. These authors show that the class of
stationary kernels with a spectral density is a rich, flexible class of kernels. So, we
optimize over spectral densities \( S(\omega) \) – which are positive integrable functions on the
real line – to recover stationary kernels. To make this optimization problem finite
dimensional, we optimize the spectral density over a finite grid of frequencies \( \omega \) and
use the trapezoidal rule to recover \( k \). To guarantee that our recovered kernels are not
overly dissimilar from \( k_0 \), we constrain ourselves to an \( \varepsilon \) ball in the \( \ell_\infty \) norm around
the spectral density of \( k_0 \) for some \( \varepsilon > 0 \). We summarize this constraint set and the
resulting optimization objective in Algorithm 5; see Appendix E.1 for more details,
including selection of \( \omega_1, \ldots, \omega_G \).

**Non-stationary kernels.** In many modeling problems, stationarity may be a
choice of convenience rather than prior conviction, or one may believe non-stationarity
is probable. In either case, we wish to allow non-stationary kernels in the neighbor-
hood \( \mathcal{K}_\varepsilon \). A convenient technique for constructing non-stationary kernels relies on
input warping \([\text{Rasmussen and Williams, 2006}, \text{Sec 4.2.3}]\). Given a kernel \( k_0 \) and a
non-linear mapping \( g \), we define a perturbed kernel \( k(x, x') = k_0(g(x), g(x')) \). This
construction guarantees that the perturbed kernel \( k \) is a kernel function as long as \( k_0 \) is
a valid kernel. We let the function \( g \) have parameters \( w \) and set \( g(x; w) := x + h(x; w) \),

78
**Algorithm 5** Objective and $K_\varepsilon$ for stationary kernels

**Objective**
1: **Input**: Frequencies $\omega_1, \ldots, \omega_G$, and density values $S(\omega_1), \ldots, S(\omega_G)$.
2: Use trapezoidal rule to approximate the integral $k(\tau) = \int e^{2\pi i \tau \omega} S(\omega) d\omega$.
3: **return** $F^* (k)$.

**Spectral constraint defining $K_\varepsilon$**
1: **Input**: Frequencies $\omega_1, \ldots, \omega_G$, density values $S(\omega_1), \ldots, S(\omega_G)$, constraint set size $\varepsilon > 0$.
2: Compute spectral density of $k_0$, $S_0(\omega_1), \ldots, S_0(\omega_G)$ via trapezoidal rule or exact formula.
3: Constrain spectral density $S$ of $k$ as:

$$\max \left( 0, (1 - \varepsilon) S_0(\omega_g) \right) \leq S(\omega_g) \leq (1 + \varepsilon) S_0(\omega_g),$$

$g = 1, \ldots, G$.

where $h : \mathbb{R}^D \to \mathbb{R}^D$ is a small neural network with weights $w$. By controlling the magnitude of $h$, we can control the deviations from $k_0$.

We could optimize the weights $w$ under the constraint $\|h(x; w)\|_2^2 \leq \varepsilon$. However, it is unclear how to enforce this constraint. Instead, we select a grid of points $\tilde{x}_1, \ldots, \tilde{x}_M \in \mathbb{R}^D$ and add a regularizer to our objective, $\frac{1}{\varepsilon M} \sum_{m=1}^M \|h(\tilde{x}_m, w)\|_2^2$, where $\varepsilon$ controls the regularization strength. We find using a grid of points to be a computationally cheap, mathematically simple, and empirically successful approximation to regularizing the entire function. We summarize our objective as a function of the network weights $w$ in Algorithm 6. Note that we have also changed our objective to include a generic loss $\ell$; some care needs to be taken to ensure that the optimal $k_1(\varepsilon)$ is finite. See Sections 6.4 and 6.5 for specific implementations of $\ell$. Given the $\hat{w}$ minimizing the objective in Algorithm 6, we set $k_1(\varepsilon)(x, x') = k_0(g(x; \hat{w}), g(x'; \hat{w}))$.

**Algorithm 6** Objective for non-stationary kernels
1: **Input**: Regularizer grid points $\tilde{x}_1, \ldots, \tilde{x}_M$, regularizer strength $\varepsilon > 0$, neural network weights $w \in \mathbb{R}^D$.
2: Define neural network $h(x; w)$ with weights $w$.
3: Define $k(x, x') := k_0(x + h(x; w), x' + h(x'; w))$.
4: **return** $\ell(k; F^*, L) + \frac{1}{\varepsilon M} \sum_{m=1}^M \|h(\tilde{x}_m; w)\|_2^2$.

### 6.2.2 Assessing qualitative interchangeability

We introduce two assessments, similar to prior predictive checks [Gabry et al., 2019], to assess qualitative interchangeability between two kernels $k_0$ and $k_1$.

**Visual comparison of prior predictive draws.** When the covariates $x$ are low-dimensional, we can plot a small collection of functions drawn from each of the
two distributions \( \mathcal{GP}(0, k_0) \) and \( \mathcal{GP}(0, k_1) \). To ensure that visual differences between the priors are due to actual differences in the kernels and not randomness in the draws, we use noise-matched prior draws. To define noise-matched draws, recall that one can draw from an \( N \)-dimensional Gaussian distribution \( \mathcal{N}(0, \Sigma) \) by computing the Cholesky decomposition \( LL^\top = \Sigma \); we then have \( Lz \sim \mathcal{N}(0, \Sigma) \), where \( z \sim \mathcal{N}(0, I_N) \).

We say that draws from two multivariate Gaussians are noise-matched if they use the same \( z \sim \mathcal{N}(0, I_N) \).

If the user believes the two plots express the same qualitative information, the kernels are qualitatively interchangeable under this test. A potential drawback to this method is that when covariates are high-dimensional, it may be difficult to effectively visualize prior draws. We address this concern next.

Comparison through Wasserstein distances. No matter the dimension of the covariates, we can numerically compare the \( \mathcal{GP}(0, k_0) \) and \( \mathcal{GP}(0, k_1) \) priors. Although directly computing distances between Gaussian processes is difficult, we can compare the \( \mathcal{GP}(0, k_0) \) and \( \mathcal{GP}(0, k_1) \) priors on \( (f(x_1), \ldots, f(x_N))^\top \). On this set, these priors are just multivariate normal distributions with covariance matrices equal to the Gram matrices \( k_0(X, X) \) and \( k_1(X, X) \) whose \((i, j)\) entries are, respectively, \( k_0(x_i, x_j) \) and \( k_1(x_i, x_j) \). Going forward, we denote by \( d(k_0, k_1) \) the 2-Wasserstein distance between these multivariate normals. We note that other statistical distances or matrix norms could have been chosen for \( d \). In Appendix E.6 we show that none of the results from any of our experiments are sensitive to this choice.

Looking only at \( d(k_0, k_1) \), it is difficult to judge whether \( k_1 \) is far from \( k_0 \) because we have no sense of scale for \( d \). To calibrate this distance, we compare it against a particular form of uncertainty about \( k_0 \). Since we learn the hyperparameter \( \hat{\theta} \) of \( k_0 \) from finite data, there remains uncertainty about \( \hat{\theta} \), which we quantify with a distribution \( q(\hat{\theta}) \). We make \( R \) i.i.d. draws \( \{\theta^{(r)}\}_{r=1}^R \) from \( q(\hat{\theta}) \) (or an approximation of \( q \)). For each \( r \), we compute \( d(k_0, k^{(r)}) \), where \( k^{(r)} \) has the same functional form as \( k_0 \) but with hyperparameters \( \theta^{(r)} \) instead of \( \hat{\theta} \). If \( d(k_0, k_1) \) is small relative to the \( d(k_0, k^{(r)}) \)'s, we say that \( k_0 \) and \( k_1 \) are qualitatively interchangeable.

In our experiments, we make the following choices. Unless otherwise stated, we approximately sample from \( q(\hat{\theta}) \) using the bootstrap. We construct a histogram of the 2-Wasserstein distance between \( k^{(r)} \) and \( k_0 \) across \( r \), with a marker indicating the position of the 2-Wasserstein between \( k_1 \) and \( k_0 \). If the marker lies to the right of the mass of the histogram, we conclude that \( k_1 \) and \( k_0 \) are not qualitatively interchangeable; otherwise, we conclude that they are.

### 6.2.3 Workflow illustration on synthetic data

Data and decision. Before turning to real data, we illustrate our workflow with a synthetic-data example. We consider \( N = 35 \) data points with a single covariate; see the leftmost panel of Fig. 6-2. We assume we have qualitative prior beliefs that (i) \( f \) is smooth and (ii) our beliefs about \( f \) are invariant to translation along the single covariate (stationarity). In this case a standard kernel choice is squared exponential: \( k_0(x, x') = \exp[-(1/2)(x - x')^2/\theta^2] \). We estimate \( \theta \) via maximum marginal likelihood estimation (MMLE). Four draws from the resulting prior are shown in the second
Figure 6-2: (Far left): Synthetic data. (Center left): Draws from the original prior $\mathcal{GP}(0, k_0)$. (Center right): Draws from $\mathcal{GP}(0, k_{\text{ex}})$. (Far right): Draws from $\mathcal{GP}(0, k_{\text{in}})$. The prior draws are noise-matched to the draws from $k_0$ (Section 6.2.2). The gray area in the right three plots is the 99.7% uncertainty interval for the original prior ($\mathcal{GP}(0, k_0)$).

The panel of Fig. 6-2

For the purposes of this illustration, we will look at two separate decisions. One is at $x^* = 2.00$, which is within the range of the training data (interpolation). And one is at $x^* = 5.29$, which is outside the range of the training data (extrapolation). Our functional of interest at either point will be the relative change in posterior mean, $F^*(k) := (\mu(x^*, k_0) - \mu(x^*, k))/\sigma(x^*, k_0)$, where $\mu(x, k)$ and $\sigma(x, k)$ are the posterior mean and standard deviation at test point $x$ under kernel $k$. We suppose that we would make a different decision if the posterior mean were two posterior standard deviations away from its current value: $F^* \geq L = 2$.

Nearby kernels. Since we assume stationarity, we choose Algorithm 5 at line 4 of Algorithm 4. Fig. 6-3 (first panel) shows what happens as we increase $\varepsilon$ to solve Eq. (6.1). The black dots (extrapolation) quickly cross the decision threshold line, so we have solved Eq. (6.1). The orange triangles (interpolation) do not cross the decision threshold line, even for very large $\varepsilon$.

Figure 6-3: (Left): Maximal value of the function $F^*$ as a function of constraint set $\varepsilon$. Comparison of the 2-Wasserstein distance between $k_0(X, X)$ and $k_1(X, X)$ to the posterior variation due to hyperparameter uncertainty for extrapolation (middle) and interpolation (right). The red line corresponds to our decision-changing kernel $k_1$.

Qualitative interchangeability: Visual comparison of prior predictive draws. We now demonstrate our first test for qualitative interchangeability. For our extrapolation example ($x^* = 5.29$), let $k_{\text{ex}}$ be the solution to Eq. (6.1). The third
The panel of Fig. 6-2 shows prior predictive draws with \( k^{(ex)} \); the draws are noise-matched with the second panel. Visually, the two sets of prior predictive draws (second and third panels) are qualitatively similar. Both are smooth and stationary by constraint; the length scale (cf. the number of “wiggles”) and amplitudes seem unchanged. We say that \( k^{(ex)} \) is qualitatively interchangeable with \( k_0 \).

Since even the largest constraint size value considered (\( \varepsilon = 10 \)) is not enough to induce the required change, we define \( k^{(in)} \) for our interpolation example \((x^* = 2.00)\) to be \( k_1(10) \) from Eq. (6.1). The fourth panel of Fig. 6-2 shows prior predictive draws with \( k^{(in)} \); the draws are noise-matched with the second panel. Again, by design, both sets of draws (second and fourth panels) are stationary and smooth. However, the magnitudes of peaks and troughs with \( k^{(in)} \) are much larger than those with \( k_0 \); cf. the gray uncertainty bands, which depict the 99.7\% uncertainty interval from the marginal of the \( k_0 \) GP. Thus, we say that \( k^{(in)} \) is not qualitatively interchangeable with \( k_0 \) under this test. We expect the difference would be even more pronounced at higher \( \varepsilon \).

**Qualitative interchangeability: 2-Wasserstein comparison.** Histograms of the 2-Wasserstein distance between \( k_0 \) and \( k^{(r)} \) appear in Fig. 6-3. \( k^{(ex)} \) sits within the histogram of alternative kernels generated via hyperparameter uncertainty (center), whereas \( k^{(in)} \) sits far outside of this uncertainty region (right). As in our prior predictive comparison, we say that \( k^{(in)} \) is not qualitatively interchangeable with \( k_0 \) under our 2-Wasserstein comparison, whereas \( k^{(ex)} \) is.

Finally, following our workflow, we conclude that our extrapolation example is non-robust to the choice of kernel in the sense of Definition 6. On the other hand, in our interpolation example, we do not find non-robustness.

### 6.3 Stationary perturbations to a model of heart rates

We now provide an example of using our workflow to assess the sensitivity of GP predictions of hospital patient deterioration. Colopy et al. [2016] use a GP to model individual patients’ heart rates and predict potentially troubling behavior at a future time \( x^* \). We check whether this prediction can be sensitive to the choice of kernel.

**Data, model, and decision.** Colopy et al. [2016] observe an outcome, heart-rate data measured in beats per minute (bpm), as a function of one covariate, time. The authors choose their GP model to have mean equal to zero and a kernel equal to the sum of a squared exponential and Matern-5/2 kernel; see Appendix E.3. We fit the overall kernel’s hyperparameters via MMLE and refer to the resulting kernel as \( k_0 \). Some standard hospital alarm systems activate at 130 bpm [Fidler et al., 2017], a threshold describing a worryingly-high resting heart rate. So we consider the task of predicting whether the 95th percentile of the GP posterior is above \( L = 130 \) bpm.

Most predictions in Colopy et al. [2016] take place 1.5 hours in the future, so we set \( F^* \) to be the 95th quantile at 1.5 hours hours after the last observed data point. Since the data from Colopy et al. [2016] is confidential, we use heart-rate data from
Figure 6-4: Sensitivity of heart rate analysis in Section 6.3.  

(Top row):  (left) Observed data.  (middle) Noise-matched draws from original prior \( \mathcal{GP}(0, k_0) \).  (right) Alternative prior \( \mathcal{GP}(0, k_1) \).  

(Bottom row):  (left) Comparison of the difference between \( k_0 \) and \( k_1 \) (red line) to bootstrapped hyperparameter uncertainty (histogram).  (middle) Once we expand the constraint set to \( \epsilon = 0.24 \) the predicted 95% quantile of heart rate at \( x^* \) exceeds 130 bpm (red line).  (right) Comparison of posterior distributions of \( f^* \) computed using \( k_0 \) (blue) and \( k_1(\epsilon^*) \) (red).

the 2019 Computing in Cardiology Challenge [Reyna et al., 2019, Goldberger et al., 2000].

Prior beliefs.  Colopy et al. [2016] note that \( k_0 \) encodes the belief that “longer trends (on the order of hours) are governed by the smooth RBF kernel, while minutely variations in [heart-rate] are governed by a twice-differentiable Matérn(5/2) kernel.” Although Colopy et al. [2016] are not explicit about assuming stationarity, we presume it is a reasonable prior belief here; while we expect that a patient’s heart rate may change while in the hospital, our prior beliefs about the timing of any changes may be roughly uniform. We thus choose \( \mathcal{K}_\epsilon \) according to the stationary specification in Section 6.2.1.

Robustness.  Fig. 6-4 depicts our workflow (Algorithm 4) in action. We solve Eq. (6.1) to obtain \( k_1(\epsilon^*) \) such that \( F^*(k_1(\epsilon^*)) \geq L \). We then compare noise-matched samples from the priors using \( k_0 \) and \( k_1(\epsilon^*) \). The noise-matched samples do not clearly represent different pieces of prior information; both prior plots display functions that are fairly rough with similar length scales. Finally, we see that the 2-Wasserstein distance between \( k_0 \) and \( k_1(\epsilon^*) \) is substantially smaller than the 2-Wasserstein distance between \( k_0 \) and kernels from the sampling uncertainty in the MMLE hyperparameters. Our tests suggest \( k_0 \) and \( k_1(\epsilon^*) \) are qualitatively interchangeable; we conclude that the prediction that \( F^* \) will breach the alarm threshold is non-robust in the sense of Definition 6.

While this outcome may be surprising, it is not entirely unintuitive. The patient’s heart rate is trending up toward the end of the observed data. In Appendix 2.3, we show an example where the observed data is trending downward at the end of the observed data. In the latter case, the resulting kernel \( k_1(\epsilon^*) \) fails both of our tests of qualitative interchangeability and so we cannot conclude non-robustness.
6.4 Non-stationary perturbations to a model of carbon dioxide emissions

In a now-classic analysis of carbon dioxide (CO\textsubscript{2}) levels at Mauna Loa, Rasmussen and Williams [2006] predicted future CO\textsubscript{2} levels based on data up to 2003. With data up to 2021, we can now see that the Rasmussen and Williams [2006] analysis substantially underestimates present-day CO\textsubscript{2} levels; compare the gray region (99.7% quantile of the original predictions) to the green (true levels) in Fig. 6-5. In this section, we show that this prediction of modern CO\textsubscript{2} levels is non-robust to kernel choice.

Data, model, and decision. At the present day, monthly data for CO\textsubscript{2} emissions is available from the year 1958 through 2021. But Rasmussen and Williams [2006] use training data up to 2003. Rasmussen and Williams [2006] use a kernel that is a sum of four basic kernels, where each term plays a specific role; e.g. a periodic term models the periodic seasonal trend in CO\textsubscript{2} levels. See Appendix E.4 for a full description of the kernel. We take this kernel with hyperparameters fit via MMLE as our \( k_0 \). Actual CO\textsubscript{2} levels breached 415 ppm for the first time in human history [Solly, 2019] in 2019. Under \( k_0 \), this level lies more than three standard deviations away from the predicted means in all of 2019. To see whether a qualitatively interchangeable \( k_1 \) would better predict modern CO\textsubscript{2} levels, we let \( F^* \) be the smooth-max of all posterior means in 2019. We will say the posterior has substantively changed if \( F^* \geq L = 415 \) ppm.

Prior beliefs. While \( k_0 \) is a stationary kernel we might also have non-stationary prior information such as known historical or expected future developments in climate policy or technology. We therefore choose \( \mathcal{K}_c \) according to the non-stationary input-warping specification in Section 6.2.1. For our regularizer grid, we use 600 evenly spaced points \( \tilde{x}_1, \ldots, \tilde{x}_{600} \) between 1958 and 2021 to control the behavior of \( h \) throughout our time period of interest. Input warping the entire kernel as \( k = k_0(g(\mathbf{x}), g(\mathbf{x}')) \) would violate an important piece of prior information that we have about CO\textsubscript{2} levels: we know CO\textsubscript{2} has a regular seasonality, with minimal levels in the winter and maximal levels in the summer. The original \( k_0 \) accounts for this feature of the data with a periodic term; as shown in Fig. 6-5 (top), this periodicity lines up very well with the training data. To produce an alternative kernel that accounts for this piece of prior knowledge, we leave the periodic portion of the kernel unwarped. To parameterize \( g \), we use a fully connected network with two hidden layers, 50 units, and ReLU nonlinearities. To ensure the optimal \( k_1 \) is finite, we take the loss in Algorithm 6 to be \( \ell(k; F^*, L) = (F^*(k) - L)^2 \), which guarantees our objective is bounded below.

Robustness. We now use our workflow to ask whether qualitatively interchangeable kernels might have better predicted the record-breaking CO\textsubscript{2} levels in 2019; see Fig. 6-5 for our results. We lower the regularization strength (i.e. increase \( \varepsilon \) in Algorithm 6) until \( F^* \geq L \). In the bottom of Fig. 6-5, we plot noise-matched prior draws from \( k_0 \) alongside draws from the resulting \( k_1(\varepsilon^*) \). Differences between draws from \( k_0 \) and \( k_1(\varepsilon^*) \) are almost visually indistinguishable on the scale of all prior draws. A
closer inspection in Appendix E.4 confirms that the two priors capture the same yearly periodic trend. These same zoomed-in plots show that the priors are not completely indistinguishable; however, in our opinion, the draws display the same prior beliefs. The 2-Wasserstein comparison in Fig. E-8 of Appendix E.4 further confirms that the perturbed kernel sits well within the histogram of alternate kernels stemming from hyperparameter uncertainty. We conclude that future predictions of CO₂ levels using the original $k_0$ are non-robust to the choice of kernel in the sense of Definition 6.

### 6.5 Non-stationary perturbations in classifying MNIST digits

So far we have restricted our attention to low-dimensional covariates. To evaluate our approach in a high-dimensional setting, we reproduce the MNIST image classification experiment of Lee et al. [2018]. It is rare to have concrete prior beliefs about high-dimensional functions, so in this case we do not consider different kernels as arising from prior beliefs. Rather we imagine $k_0$ is used purely for convenience and predictive quality – but that a malicious actor is interested in changing the kernel to achieve different test predictions without detection.

**Data, model, and decision.** Similar to Lee et al. [2018], we use 1000 randomly sampled MNIST images for a training set, and a separate 1000 images for a test set. Given a test image $\mathbf{x}^*$, Lee et al. [2018] predict the class label $c \in \{1, \ldots, C\}$ by using a $C$-output GP with compositional structure, considered as the infinite-width limit of a sequence of Bayesian neural networks Lee et al. [2018], de G. Matthews et al. [2018]. The authors classify any image $\mathbf{x}^*$ by picking the class $c$ that has the posterior mean
Figure 6-6: Sensitivity of MNIST analysis in Section 6.5. (Left): $F^*$ as a function of regularizer strength. (Right): Histogram of the 2-Wasserstein distances between the 1000 (one for each test image) input-warped kernel Gram matrices (in red) plotted with those arising from kernel hyperparameter uncertainty around $k_0$ (in black).

of $f_c(x^*)$, i.e. $\mu_c(x^*)$, closest to 0.9; see Appendix E.5 for details. We use the kernel and hyperparameters from [Lee et al. 2018] for $k_0$. We imagine that the malicious actor wants to change the label of a single test image $x^*$ from its current label $c_0$ to a different label $c_1$. For concreteness, we set $c_1 := |c_0 - 1|$. We consider 1000 separate iterations of this exercise, once for each of the 1000 test images. For a particular $x^*$, we set our posterior quantity of interest to be $F^* = |\mu_{c_0}(x^*) - 0.9| - |\mu_{c_1}(x^*) - 0.9|$. Since $F^* \geq 0$ implies that we have changed the prediction for $x^*$, we set our decision threshold $L = 0$.

**Malicious actor.** Instead of considering a range of priors that match prior beliefs, we here consider a range of priors that will allow a malicious actor to avoid detection. Since we have no prior belief of stationarity, we use the non-stationary construction from Section 6.2.1. We find that optimizing $F^*$ directly leads to kernels where for $c \neq c_1$, $\mu_c(x^*)$ takes on values at least an order of magnitude higher than for the original kernel. This change could be easily detected by an automated system. For the purposes of the malicious actor, we therefore consider these kernels to not be qualitatively interchangeable. We instead optimize a surrogate loss that maximizes the log probability of $c_1$ being correct and all other classes being incorrect; see Appendix E.5. We find that optimizing this surrogate loss leads to more benign-looking $\mu_c(x^*)$ and achieves $F^* \geq L$.

**Robustness.** Fig. 6-6 shows the results of our workflow applied to this problem. We find a sufficiently large setting of $\varepsilon$ that allows us, across all 1000 test-image problems, to change every decision. In particular, for $\varepsilon = 10^{-4}$, we are able to find perturbed kernels that change the predicted class label in every case. It is not clear how to visualize our priors in this application. So, of the approaches in Section 6.2.2, we use only the hyperparameter uncertainty visualization to assess qualitative interchangeability. [Lee et al. 2018] optimize the hyperparameters of $k_0$ over a grid. Instead of bootstrapping this procedure, we note that the size of the grid defines a natural variability in the hyperparameters $\hat{\theta}$. To be conservative, we sample $\theta^{(r)}$ from an area around the hyperparameters selected by [Lee et al. 2018] that is over 10 times smaller than the full grid. (Using the full original grid would find more extreme non-robustness.) The Gram matrices corresponding to our perturbed kernels
are much closer to \( k_0(X, X) \) than are the Gram matrices corresponding to each \( \theta^{(r)} \). We conclude that classification of handwritten digits using \( k_0 \) is non-robust to the choice of kernel in the sense of Definition 6.

6.6 Discussion

In this chapter, we proposed and implemented a workflow for measuring the sensitivity of GP inferences to the choice of the kernel function. We used our workflow to discover substantial non-robustness in a variety of practical examples, but also showed that many analyses are not flagged as non-robust by our method. There are many exciting directions for expanding on the present work – both within our existing workflow and beyond. Some of our choices in the present chapter were made for mathematical convenience. For instance, the constraint in our stationary objective in Algorithm 5 and the regularizer in our non-stationary objective in Algorithm 6 might be replaced by other notions of “nearby” spectral densities or “small” input warpings, respectively. Additionally, our framework flags robustness but does not show how to make an analysis more robust. The instances of non-robustness we have found suggest it might be worthwhile to develop methods to robustify GP inferences to the choice of kernel. One challenge would be understanding how to best balance robustness against the ability to adapt to the prior assumptions at hand: if a method is completely robust to any change in prior assumptions, there is no point in specifying a prior at all!
Chapter 7

Conclusions

7.1 Summary

In this thesis, we developed new model assessment tools that are easier and faster to use in modern, complex data analyses involving large datasets. The first model assessment method we focused on was cross-validation (CV), which usually becomes computationally expensive when used with complex models trained on large datasets. Previous work addressed this computational expense by approximating CV. In Chapters 2 and 3, we noted that these pre-existing approximations became both slow and inaccurate when used with high-dimensional models. We then showed that CV can be approximated quickly and accurately despite the presence of high dimensions when there is hidden low-dimensional structure in the data. In particular, we developed fast and accurate approximations to CV for high-dimensional sparse and low-rank generalized linear models in Chapters 2 and 3, respectively. In Chapter 4, we further extended the reach of approximate CV by developing approximate CV algorithms that apply to more complex, structured forms of CV in structured models. In particular, we showed how to approximate CV for models with latent structure following a Markov random field and forms of CV that drop out parts of that latent structure. In Chapter 5, we recalled that our focus on approximating CV assumed that exact CV has desirable and well-understood behavior. However, we showed that even exact CV can fail to be an effective tool for automatically and efficiently tuning model hyperparameters; that is, we showed that there can be local optima in the CV loss, posing challenges for tuning hyperparameters via optimizing the CV loss.

Finally, we noted that robustness checks can be expensive in terms of user time, as they often require the user to specify and re-fit multiple alternative models. In Chapter 6, we considered the case of prior robustness checks for Gaussian processes, for which a user would typically have to specify multiple alternative kernel functions. To save user time, we developed automated tools to search for alternative kernel functions that change an analysis’s outcome and help a user understand if this kernel function represents a valid alternative model.
7.2 Future work

We see many interesting directions for future work building on and around the work in this thesis. These directions are organized below into three main categories: CV in neural networks, local optima in exact CV, and robustness checks.

**CV in neural networks.** We have noted the especial need for CV in large and complex models; so, in this thesis, we focused on approximating CV in high-dimensional generalized linear models and models with latent structure following a Markov random field. This thesis does not cover neural networks, one of the most popular classes of large, complex models. Musgrave et al. [2020] show that CV is an important and underused tool for neural networks, so can we approximate CV for neural networks? Certainly modern neural networks are high-dimensional, so many of the challenges in Chapters 2 and 3 will apply; we hope that the insights in these chapters will help resolve these challenges. A broader challenge is to resolve how exact CV should be performed for neural networks. Neural networks have a highly non-convex loss landscape typically minimized with stochastic optimization methods using random initialization and early stopping. How should this procedure be replicated across the folds of CV? Should each fold of CV use a new random initialization? If so, how should each fold be initialized? Or should each fold of CV start from the full data fit? If so, how should we define our early stopping criteria? Our ability to approximate CV for neural networks using the methods developed in this thesis will depend on the answers to these questions. In particular, the methods developed in this thesis depend on local approximations to the objective function and how it changes across the folds of CV. If each fold of our desired exact CV scheme makes substantial, global changes to the fitted parameters (e.g. by using a new random restart), then it may not be possible to accurately approximate CV using the methods developed in this thesis. So, it will be interesting to consider not only what the optimal exact CV scheme is for neural networks, but also what the best CV scheme is that we can accurately and efficiently approximate.

**Local optima in exact CV.** In Chapter 5 we showed that the leave-one-out CV (LOOCV) loss for tuning the regularization parameter in ridge regression can have (non-global) local optima. We showed that the presence of these local optima is a complex function of the problem inputs and then proved conditions under which there are guaranteed to be no local optima. In Section 5.7 we discussed a number of directions for future work in this space, which we briefly summarize here. First, our experiments in Chapter 5 did not sharply characterize when the LOOCV does or does not have local optima; giving a more precise characterization of these local optima will help guide practitioners on when they can trust CV for hyperparameter tuning. Second, we did not attempt to understand how many non-global local optima can appear in the LOOCV loss. In all of our experiments the number was either zero or one, but precisely understanding this behavior would be both theoretically interesting and practically meaningful. Finally, extending the findings of Chapter 5 to models beyond ridge regression would be of interest to practitioners. E.g. if a model as
simple as ridge regression can have fairly severe local optima issues, how bad should a practitioner expect things to get for models with more complicated structure or more than one hyperparameter? And can we establish conditions similar to those of Theorem 5 under which the CV loss is guaranteed to have only global optima so we know when to trust the use of CV for hyperparameter tuning?

**Robustness checks.** In Chapter 6 we aimed to find GP kernels that are qualitatively interchangeable with the user’s specified kernel and yet reach a different decision. We assumed the user specified their prior beliefs via a single chosen kernel. While this is a common occurrence, users also often specify a distribution over kernels, for example by specifying a prior distribution over kernel hyperparameters. Extending our methodology to this case will require answers to new questions. First, how should we search over distributions over kernels that are “nearby” the user-specified distribution to find a decision-changing distribution? Second, what sorts of visualizations and statistics should we report to help a user decide if two distributions over kernels are qualitatively interchangeable?

Another direction for future work is to develop generic tests for qualitative interchangeability beyond GPs. Bayesian prior robustness checks are certainly relevant beyond GPs, and there is a great deal of work on finding “worst-case” alternative priors that substantially change the posterior [Berger et al., 1994, Gustafson, 1996, Berger, 2000, Owhadi et al., 2015, Giordano et al., 2021]. However, there seems to be limited work on helping users qualitatively understand a given prior (e.g. see Gabry et al. [2019]), and even less work on helping users compare two priors to see if they represent the same prior beliefs. So it is hard for users to understand whether these worst-case priors represent reasonable alternative specifications of their beliefs. Developing general tools and ideas for helping users assess qualitative interchangeability will increase the trustworthiness of such prior robustness checks, hopefully broadening the range of applications they are used in.
Appendix A

Appendix for: Approximate cross-validation for sparse models in high dimensions (Chapter 2)

A.1 Cross-validation methods

In this appendix, we review standard cross-validation (CV) for optimization problems of the form:

$$\arg\min_{\theta \in \Theta} \sum_{n=1}^{N} f_n(\theta) + \lambda R(\theta),$$

where $\Theta \subseteq \mathbb{R}^D$. By leave-one-out cross-validation (LOOCV), we mean the process of repeatedly computing:

$$\hat{\theta}_{\setminus n} := \sum_{m: m \neq n} f_m(\theta) + \lambda R(\theta).$$

The parameter estimates $\{\hat{\theta}_{\setminus n}\}_{n=1}^{N}$ can then be used to produce an estimate of variability or out-of-sample error; e.g., to estimate the out-of-sample error, one computes $(1/N) \sum_{n} f_n(\hat{\theta}_{\setminus n})$. By $K$-fold cross-validation, we mean the process of splitting up the dataset into $K$ disjoint folds, $S_1, \ldots, S_K$ with $S_1 \cup \cdots \cup S_K = [N]$. One then estimates the parameters:

$$\hat{\theta}_{\setminus S_k} := \arg\min_{\theta \in \Theta} \sum_{n: n \notin S_k} f_n(\theta) + \lambda R(\theta).$$

The parameter estimates $\{\hat{\theta}_{\setminus S_k}\}_{k=1}^{K}$ can then be used to produce an estimate of variability or out-of-sample error.
A.2 Scaling of the leave-one-out objective

We defined \( \hat{\theta}^n \) as the solution to the following optimization problem:

\[
\hat{\theta}^n := \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{m: m \neq n} f_m(\theta) + \lambda R(\theta).
\]

An alternative would be to use the objective \( \frac{1}{(N-1)} \sum_{m: m \neq n} f_m + \lambda R \) in order to keep the scaling between the regularizer and the objective the same as in the full-data problem. Indeed, all existing theory that we are aware of for CV applied to \( \ell_1 \) regularized problems seems to follow the \( \frac{1}{N} \) scaling [Homriglhausen and McDonald, 2014, 2013, Miolane and Montanari, 2021, Chetverikov et al., 2021]. On the other hand, all existing approaches to approximate LOOCV for regularized problems have used the \( \frac{1}{N} \) scaling that we have given [Beirami et al., 2017, Rad and Maleki, 2020, Wang et al., 2018, Xu et al., 2021, Obuchi and Kabashima, 2016, 2018]. Note that the scaling is not relevant in Giordano et al. [2019b], as they do not consider the regularized case. As our work is aimed at identifying when existing approximations work well in high dimensions, we have followed the \( \frac{1}{N} \) choice from the literature on approximate LOOCV. The different results from using the two scalings may be insignificant when leaving only one datapoint out. But one might expect the difference to be substantial for, e.g., \( K \)-fold CV. We leave an understanding of what the effect of this scaling is (if any) to future work.

A.3 Approximately solving \( \tilde{J}_n(R) \) and \( \tilde{N}_S_n(R) \)

We have seen \( \tilde{J}_n(R) \) and \( \tilde{N}_S_n(R) \) are in general not accurate for high-dimensional problems. Even worse, they can become prohibitively costly to compute due to the \( O(D^3) \) cost required to solve the needed linear systems. One idea to at least alleviate this computational burden, proposed by Koh and Liang [2017] in a slightly different context, is to use a stochastic inverse Hessian-vector-product from Agarwal et al. [2017] to approximately compute \( \tilde{J}_n(R) \) and \( \tilde{N}_S_n(R) \). Although this method works well for the purposes of Koh and Liang [2017], we will see that in the context of approximate CV, it adds a large amount of extra error on top of the already inaccurate \( \tilde{N}_S_n(R) \) and \( \tilde{J}_n(R) \).

We first describe this stochastic inverse Hessian-vector-product technique and argue that it is not suitable for approximating cross-validation. The main idea from Agarwal et al. [2017] is to use the series:

\[
H^{-1} = \sum_{k=0}^{\infty} (I - H)^k,
\]

which holds for any positive definite \( H \) with \( \|H\|_{op} \leq 1 \). Now, we can both truncate
Figure A-1: Stochastic Hessian experiments from Appendix A.3. We show percent error of approximation versus compute time for two different dataset sizes. We show three techniques for computing CV: exactly computing CV (black dot, which naturally has 0% error), $\tilde{I}_{\eta n}(R)$ with exactly computing the needed linear systems (red dot), and $\tilde{I}_{\eta n}(R)$ with the stochastic solves described in Appendix A.3 (blue dots, one for each setting of the parameters $K$ and $M$). Values of $M$ and $K$ used are described in Appendix A.3; we use an extended range of settings for the smaller dataset to more extensively illustrate the approximation’s behavior. Settings of $K$ and $M$ for which the stochastic solves are roughly as fast as exactly computing $\tilde{I}_{\eta n}(R)$ result in a significantly less accurate approximation of CV.

this series at some level $K$ and write it recursively as:

$$H^{-1} \approx H^{-1}_K := I + (I - H)H^{-1}_{K-1},$$

where $H_0^{-1} = I$. Next, to avoid computing $H$ explicitly, we can note that if $A_k$ is some random variable with $E[A_k] = H$, we can instead just sample a new $A_k$ at each iteration to define:

$$\tilde{H}^{-1}_k := I + (I - A_k)\tilde{H}^{-1}_{K-1}.$$ 

In our case, we pick a random $n_k \in [N]$ and set $A_k = \nabla^2 f(x^T_{n_k}, y_{n_k}) + (1/N)\lambda\nabla^2 R(\theta)$. Finally, Agarwal et al. [2017] suggest taking $M$ samples of $\tilde{H}^{-1}_K$ and averaging the results to lower the variance of the estimator. This leaves us with two parameters to tune: $M$ and $K$. Increasing either will make the estimate more accurate and more expensive to compute. Koh and Liang [2017] use this approximation to compute $\tilde{I}_{\eta n}(R)$ for high dimensional models such as neural networks; however, we remark that their interest lies in the qualitative properties of $\tilde{I}_{\eta n}(R)$, such as signs and relative magnitudes across various values of $n$. It remains to be seen whether this stochastic solver can be successfully used to approximate CV.

To test the application of this approximation to approximate CV, we generated a synthetic logistic regression dataset with covariates $x_{nd} \sim i.i.d. N(0,1)$. We use $R(\theta) = \|\theta\|_2^2$. In Fig. A-1 we show that for two settings of $N$ and $D$ there are no settings of
and $K$ for which using $\bar{H}_K^{-1}$ to compute $\tilde{I}_{J\setminus n}(R)$ provides a both fast and accurate approximation to $CV$. Specifically, we range $K \in \{1, 20, 30, 50, 60, 80, 100, 120\}$ and $M \in \{2, 25\}$ and see that when the stochastic approximation is faster, it provides only a marginal speedup while providing a significantly worse approximation error.

### A.4 Further details of Eq. (2.2) and Eq. (2.3)

In Section 2.2, we briefly outlined the approximations $\tilde{N}_{S\setminus n}(R)$ and $\tilde{I}_{J\setminus n}(R)$ to $\hat{\theta}^n$; we give more details about these approximations and their derivations here. Recall that we defined $H(\hat{\theta}) := (1/N) \sum_{n=1}^{N} \nabla^2_{\theta} f(x_n^T \hat{\theta}, y_n) + \lambda \nabla^2_{R} R(\hat{\theta})$. We first restate the “infinitesimal jackknife” approximation from the main text, which was derived by the same approach taken by Giordano et al. [2019b]:

$$
\hat{\theta}^n \approx \tilde{I}_{J\setminus n}(R) := \hat{\theta} + \frac{1}{N} H(\hat{\theta})^{-1} \nabla_{\theta} f(x_n^T \hat{\theta}, y_n). \quad (A.1)
$$

The “Newton step” approximation, similar to the approach in Beirami et al. [2017] and identical to the approximation in Rad and Maleki [2020], Wang et al. [2018], is:

$$
\hat{\theta}^n \approx \tilde{N}_{S\setminus n}(R) := \hat{\theta} + \frac{1}{N} \left( H(\hat{\theta}) - \frac{1}{N} \nabla^2_{\theta} f(x_n^T \hat{\theta}, y_n) \right)^{-1} \nabla_{\theta} f(x_n^T \hat{\theta}, y_n). \quad (A.2)
$$

### A.4.1 Derivation of $\tilde{I}_{J\setminus n}(R)$

We will see in Appendix A.4.3 that, after some creative algebra, $\tilde{I}_{J\setminus n}(R)$ is an instance of $\hat{\theta}_{IJ}$ from Definition 2 of Giordano et al. [2019b]. However, this somewhat obscures the motivation for considering Eq. (A.1). As an alternative to jamming our problem setup into that considered by Giordano et al. [2019b], we can more directly obtain the approximation in Eq. (A.1) by a derivation only slightly different from that in Giordano et al. [2019b]. We begin by defining $\theta^w$ as the solution to a weighted optimization problem with weights $w_n \in \mathbb{R}$:

$$
\theta^w := \arg \min_{\theta \in \Theta} G(w, \theta) := \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} w_n f(x_n^T \theta, y_n) + \lambda R(\theta), \quad (A.3)
$$

where we assume $G$ to be twice continuously differentiable with an invertible Hessian at $\theta^1$ (where $\theta^1$ is the solution in Eq. (A.3) with all $w_n = 1$). For example, we have that $\hat{\theta}^n = \theta^w$ if $w$ is the $N$-dimensional vector of all ones but with a zero in the $n$th coordinate. We will form a linear approximation to $\theta^w$ as a function of $w$. To do so, we will need to compute the derivatives $d\theta^w/dw_n$ for each $n$. To compute these derivatives, we begin with the first order optimality condition of Eq. (A.3) and take
a total derivative with respect to \( w_n \):

\[
\frac{\partial G}{\partial \theta} \bigg|_{w=1, \theta=\theta^1} = 0
\]

\[
\implies \frac{d}{dw_n} \frac{\partial G}{\partial \theta} \bigg|_{w=1, \theta=\theta^1} = \frac{\partial^2 G}{\partial \theta \partial w_n} \bigg|_{w=1, \theta=\theta^1} dw_n + \frac{\partial^2 G}{\partial \theta^2} \bigg|_{w=1, \theta=\theta^1} d\theta^w = 0.
\]

Re-arranging, defining \( H(\theta^1) := \nabla_\theta^2 G(w, \theta^1) \), and using the assumed invertibility of \( H(\theta^1) \) gives:

\[
\frac{d}{dw_n} \bigg|_{w=1} \frac{\partial G}{\partial \theta} \bigg|_{w=1, \theta=\theta^1} = -\left( \frac{\partial^2 G}{\partial \theta^2} \bigg|_{w=1, \theta=\theta^1} \right)^{-1} \frac{\partial^2 G}{\partial w_n \partial \theta} \bigg|_{w=1, \theta=\theta^1} = -\frac{1}{N} H(\hat{\theta})^{-1} \nabla_\theta f(x_n^T \hat{\theta}, y_n). \tag{A.4}
\]

In the final equality, we used the fact that \( \theta^1 = \hat{\theta} \). Now, by a first order Taylor expansion around \( w = (1, 1, \ldots, 1) \), we can write:

\[
\theta^w \approx \hat{\theta} + \sum_{n=1}^{N} \frac{d}{dw_n} \bigg|_{w=1} \frac{\partial G}{\partial \theta} \bigg|_{w=1, \theta=\theta^1} (w_n - 1) \tag{A.5}
\]

\[
= \hat{\theta} - \frac{1}{N} \sum_{n=1}^{N} H(\hat{\theta})^{-1} \nabla_\theta f(x_n^T \hat{\theta}, y_n)(w_n - 1). \tag{A.6}
\]

For the special case of \( w \) being the vector of all ones with a zero in the \( n \)th coordinate (i.e., the weighting for LOOCV), we recover Eq. (A.1).

### A.4.2 Invertibility in the definition of \( \tilde{S}_{n}(R) \) and \( \tilde{J}_{n}(R) \)

In writing Eqs. (2.2) and (2.3) we have assumed the invertibility of \( H(\hat{\theta}) \) and \( H(\hat{\theta}) - (1/N) \nabla_\theta f(x_n^T \hat{\theta}, y_n) \). We here note a number of common cases where this invertibility holds. First, if \( \nabla^2 R \) is positive definite for all \( \theta \) (as in the case of \( R = \| \cdot \|^2_2 \)), then these matrices are always invertible. If \( R \) is merely convex, \( H(\hat{\theta}) - (1/N) \nabla_\theta f(x_n^T \hat{\theta}, y_n) \) is invertible if \( \text{Span} \{ x_m \}_{m: m \neq n} = \mathbb{R}^D \). This condition on the span holds almost surely if the \( x_n \) are sampled from a continuous distribution and \( D \leq N \).

### A.4.3 Accuracy of \( \tilde{J}_{n}(R) \) for regularized problems

As noted in the main text, Giordano et al. [2019b] show that the error of \( \tilde{J}_{n}(R) \) is bounded by \( C/N \) for some \( C \) that is constant in \( N \). However, their results apply only to the unregularized case (i.e., \( \lambda = 0 \)). We show here that their results can be extended to the case of \( \lambda > 0 \) with mild additional assumptions; the proof of Proposition 10 appears below.

**Proposition 10.** Assume that the conditions for Corollary 1 of Giordano et al. [2019b] are satisfied by \( F(\theta) \). Furthermore, assume that we are restricted to \( \theta \) in some compact subset \( \Theta \) of \( \mathbb{R}^D \), \( \lambda = O(1/\sqrt{N}) \), \( F + \lambda R \) is twice continuously differ-
entiable for all \( \theta \), and that \( \nabla^2 R(\theta) \) is positive definite for all \( \theta \in \Theta \). Then \( \tilde{I}_{\lfloor n \rfloor}(R) \) can be seen as an application of the approximation in Definition 2 of [Giordano et al. 2019]. Furthermore, the assumptions of their Corollary 1 are met, which implies:

\[
\max_{n \in [N]} \| \tilde{\psi}_{IJ} - \hat{\theta}^n \|_2 \leq \frac{C'}{N^2} \sup_{\theta \in \Theta} \max_{n \in [N]} \| \nabla \theta f(x_n^T \theta, y_n) \|_\infty \leq \frac{C'}{N},
\]

(A.7)

where \( C \) and \( C' \) are problem-specific constants independent of \( N \) that may depend on \( D \).

Proposition 10 provides two bounds on the error \( \| \tilde{I}_{\lfloor n \rfloor}(R) - \hat{\theta}^n \|_2 \): either \( C'/N^2 \) times the maximum of the gradient or just \( C/N \). One bound or the other may be easier to use, depending on the specific problem. It is worth discussing the conditions of Proposition 10 before going into its proof. The first major assumption is that \( \theta \) is restricted to some compact set \( \Theta \). Although this assumption may not be satisfied by problems of interest, one may be willing to assume that \( \theta \) lives in some bounded set in practice. In any case, such an assumption seems necessary to apply the results of [Giordano et al. 2019] to most unregularized problems, as they, for example, require \( \sup_{\theta \in \Theta} F(\theta) \) to be bounded. We will require the compactness of \( \Theta \) to show that \( \sup_{\theta \in \Theta} F(\theta) + \lambda R(\theta) \) is bounded.

The second major assumption of Proposition 10 is that \( \lambda = O(1/\sqrt{N}) \). We need this assumption to ensure that the term \( \lambda R(\theta) \) is sufficiently well behaved. In practice this assumption may be somewhat limiting; however, we note that for fixed \( D \), such a scaling is usually assumed – and in some situations is necessary – to obtain standard theoretical results for \( \ell_1 \) regularization (e.g., [Wainwright 2009] gives the standard scaling for linear regression, \( \lambda = \Omega(\sqrt{\log(D)/N}) \)). Our Theorems 2 and 3 also satisfy such a scaling when \( D \) is fixed. In any case, we stress that this assumption – as well as the assumption on compactness – are needed only to prove Proposition 10 and not any of our other results. We prove Proposition 10 to demonstrate the baseline results that exist in the literature so that we can then show how our results build on these baselines.

Proof. We proceed by showing that the regularized optimization problem in our Eq. (2.1) can be written in the framework of Eq. (1) of [Giordano et al. 2019] and then showing that the re-written problem satisfies the assumptions of their Corollary 1. First, the framework of [Giordano et al. 2019] applies to weighted optimization problems of the form:

\[
\theta^w := \theta \in \Theta \text{ s.t. } \frac{1}{N} \sum_{n=1}^{N} w_n g_n(\theta) = 0.
\]

(A.8)

In order to match this form, we will rewrite the gradient of the objective in Eq. (2.1) as a weighted sum with \( N + 1 \) terms, where the first term, with weight \( w_0 = 1 \), will
correspond to $R(\theta)$:

$$
\frac{1}{N+1} w_0 (N+1) \lambda \nabla R(\theta) + \frac{1}{N+1} \sum_{n=1}^{N} w_n \frac{N+1}{N} \nabla f(x_n^T \theta, y_n). \tag{A.9}
$$

We will also need a set of weight vectors $W \subseteq \mathbb{R}^{N+1}$ for which we are interested in evaluating $\theta^w$. We choose this set as follows. In the set, we include each weight vector that is equal to one everywhere except $w_n = 0$ for exactly one of $n \in \{1, \ldots, N\}$. Thus, for each $n$, there is a $w \in W$ such that $\theta^w = \hat{\theta}^n$. Finally, then, we can apply Definition 2 of [Giordano et al. 2019b] to find the approximation $\theta_{1,J}(w)$ for the $w$ that corresponds to leaving out $n$. We see that $\theta_{1,J}(w)$ in this case is exactly equal to $\widetilde{J}_{\setminus n}(R)$ in our notation here.

Now that we know our approximation is actually an instance of $\theta_{1,J}(w)$, we need to check that Eq. (A.9) meets the assumptions of Corollary 1 of [Giordano et al. 2019b] to apply their theoretical analysis to our problem. We check these below, first stating the assumption from [Giordano et al. 2019b] and then covering why it holds for our problem.

1. (Assumption 1): for all $\theta \in \Theta$, each $g_n$ is continuously differentiable in $\theta$.
   For our problem, by assumption, $R(\theta)$ and $f(x_n^T \theta, y_n)$ are twice continuously differentiable functions of $\theta$, so this assumption holds.

2. (Assumption 2): for all $\theta \in \Theta$, the Hessian matrix, $H(\theta,1) := (1/N) \sum_n \partial g_n(\theta)/\partial \theta^T$ is invertible and satisfies $\sup_{\theta \in \Theta} \|H^{-1}(\theta,1)\|_{op} \leq C_{op} < \infty$ for some constant $C_{op}$, where $\| \cdot \|_{op}$ denotes the operator norm on matrices with respect to the $\ell_2$ norm (i.e., the maximum eigenvalue of the matrix).
   For our problem, by assumption, the inverse matrix $(\nabla^2 F(\theta))^{-1}$ exists and has bounded maximum eigenvalue for all $\theta \in \Theta$. Also by assumption, $R$ has a positive semidefinite Hessian for all $\theta$, which implies:

$$
\sup_{\theta \in \Theta} \|H^{-1}(\theta,1)\|_{op} = \sup_{\theta \in \Theta} \left\| \left( \nabla^2 F(\theta) + \lambda \nabla^2 R(\theta) \right)^{-1} \right\|_{op} \leq \sup_{\theta \in \Theta} \left\| \left( \nabla^2 F(\theta) \right)^{-1} \right\|_{op}.
$$

To see that the inequality holds, first note that for a positive semi-definite (PSD) matrix $A$, $\|A^{-1}\|_{op} = 1/\lambda_{min}(A)$. The inequality would then follow if $\lambda_{min}(\nabla^2 F(\theta) + \lambda \nabla^2 R(\theta)) \geq \lambda_{min}(\nabla^2 F(\theta))$. To see that this holds, take any two $D \times D$ PSD matrices $A$ and $B$. Let $\lambda_d(\cdot)$ be the $d$th eigenvalue of a matrix with $\lambda_1 = \lambda_{min}$. Then:

$$
\lambda_d(A + B) = \min_{E \subseteq \mathbb{R}^B} \max_{x \in E} x^T(A + B)x \geq \min_{E \subseteq \mathbb{R}^A} \max_{x \in E} x^TAx = \lambda_d(A),
$$

where the inequality holds because $B$ is PSD. So, $\lambda_{min}(A+B) \geq \lambda_{min}(A)$, which finishes the proof. We have thus showed that the operator norm of $H^{-1}(\theta,1)$ is bounded by that of $\nabla^2 F(\theta)^{-1}$ for all $\theta \in \Theta$.

3. (Assumption 3): Let $g(\theta)$ and $h(\theta)$ be the $(N+1) \times D$ stack of gradients and $(N+
1) \times D \times D \text{ stack of Hessians, respectively. That is, } g(\theta)_{nd} := (\nabla_{\theta} f(x_n^T \theta, y_n))_d \text{ for } n = 1, \ldots, N \text{ and } g(\theta)_{N+1,d} := (\nabla_{\theta} R(\theta))_d, \text{ with } h \text{ defined similarly. Let } \|g(\theta)\|_2 \text{ be the } \ell_2 \text{ norm of } g \text{ flattened into a vector with } \|h(\theta)\|_2 \text{ defined similarly. Then assume that there exist constants } C_g \text{ and } C_h \text{ such that:}

\begin{align*}
\sup_{\theta \in \Theta} \frac{1}{\sqrt{N+1}} \|g(\theta)\|_2 &\leq C_g < \infty \\
\sup_{\theta \in \Theta} \frac{1}{\sqrt{N+1}} \|h(\theta)\|_2 &\leq C_h < \infty
\end{align*}

To see that this holds for our problem, we have that:

\[ \|g(\theta)\|_2 := \left[ \sum_{d=1}^{D} \left( (\lambda(N+1)\nabla R(\theta)_d)^2 + \sum_{n=1}^{N} \frac{N+1}{N} (\nabla f(x_n^T \theta, y_n)_d)^2 \right) \right]^{1/2} \]

\[ \leq \lambda(N+1) \|\nabla R(\theta)\|_2 + \frac{N+1}{N} \left[ \sum_{d=1}^{D} \sum_{n=1}^{N} (\nabla f(x_n^T \theta, y_n)_d)^2 \right]^{1/2} \]

We need to show this is bounded by \(\sqrt{N+1}C_g\) for some constant \(C_g\). By assumption in the statement of Proposition \[10\] we have \(\frac{1}{\sqrt{N+1}} \|\nabla F(\theta)\|_2 \leq \frac{1}{\sqrt{N}} \|\nabla F(\theta)\|_2 \leq C_g^{(1)}\) for some constant \(C_g^{(1)}\). Because \(\lambda\) is \(O(1/\sqrt{N})\), the first term is equal to \(O(\sqrt{N})\|\nabla R(\theta)\|_2\). The compactness of \(\Theta\) and the continuity of \(\nabla R(\theta)\) imply that \(\|\nabla R(\theta)\|_2\) is bounded by a constant for all \(\theta \in \Theta\). So, we know that \(\frac{O(\sqrt{N})}{\sqrt{N+1}} \|\nabla R(\theta)\|_2 \leq C_g^{(2)}\) for some constant \(C_g^{(2)}\). Thus, we have that the assumption on \(\|g(\theta)\|_2\) holds with \(C_g = \frac{(N+1)}{N}C_g^{(1)} + C_g^{(2)}\). That the condition on \(\|h(\theta)\|_2\) holds follows by the same reasoning.

4. (Assumption 4): There exists some \(\Delta_\theta > 0\) and \(L_h < \infty\) such that if \(\|\theta - \hat{\theta}\|_2 \leq \Delta_\theta\), then \(\frac{1}{\sqrt{N+1}} \|h(\theta) - h(\hat{\theta})\|_2 \leq L_h \|\theta - \hat{\theta}\|_2\).

We can show this holds for our problem by:

\[ \|h(\theta) - h(\hat{\theta})\|_2 := \| \frac{N+1}{N} \nabla^2 F(\theta) + \lambda(N+1)\nabla^2 R(\theta) - \frac{N+1}{N} \nabla^2 F(\hat{\theta}) - \lambda(N+1)\nabla^2 R(\hat{\theta}) \|_2 \]

\[ \leq (N+1)\lambda \left\| \nabla^2 R(\theta) - \nabla^2 R(\hat{\theta}) \right\|_2 + \frac{N+1}{N} \left\| \nabla^2 F(\theta) - \nabla^2 F(\hat{\theta}) \right\|_2, \]

where we have abused notation to denote \(\|\nabla^2 F(\theta)\|_2 := \sqrt{\sum_{i,j=1}^{D} \sum_{n=1}^{N} \nabla^2_{ij}(f(x_n^T \theta, y_n)_{ij})^2}\). Now, we want to show that this quantity divided by \(\sqrt{N+1}\) is bounded by \(L_h \|\theta - \hat{\theta}\|_2\) for some constant \(L_h\). By assumption in the statement of Proposition \[10\] we have that Assumption 4 holds for \(F\); this
implies that
\[
\frac{N + 1}{(\sqrt{N} + 1)(N)} \left\| \nabla^2 F(\theta) - \nabla^2 F(\hat{\theta}) \right\|_2 \leq L_h^{(1)} \| \theta - \hat{\theta} \|_2
\]
for some constant $L_h^{(1)}$. As $R$ is twice continuously differentiable and the condition of Assumption 4 needs only to hold over a compact set of $\theta$’s, we know that $\nabla^2 R(\theta)$ is Lipschitz over this domain. Using this along with the assumption that $\lambda$ is $O(1/\sqrt{N})$, we have that:
\[
\frac{\lambda(N + 1)}{\sqrt{N} + 1} \left\| \nabla^2 R(\theta) - \nabla^2 R(\hat{\theta}) \right\|_2 = O(1) \left\| \nabla^2 R(\theta) - \nabla^2 R(\hat{\theta}) \right\|_2
\]
\[
\leq L_h^{(2)} \left\| \theta - \hat{\theta} \right\|_2,
\]
for some constant $L_h^{(2)}$. So, Assumption 4 holds with constant $L_h = L_h^{(1)} + L_h^{(2)}$.

5. (Assumption 5): For all $w \in W$, we have $\frac{1}{\sqrt{N}+1} \|w\|_2 \leq C_w$ for some constant $C_w$. This is immediately true for our definition of $W$, which, for all $w \in W$, has $\|w\|_2 = \sqrt{N}$.

\[\square\]

### A.4.4 Derivation of $\widehat{\text{NS}}_\setminus n(R)$

Wang et al. [2018] and Rad and Maleki [2020] derive $\widehat{\text{NS}}_\setminus n(R)$ in Eq. (A.2) by taking a single Newton step on the objective $F\setminus n + \lambda R$ starting at the point $\hat{\theta}$. For completeness, we include a derivation here. Recall that the objective with one datapoint left out is:

\[
F\setminus n(\theta) + \lambda R(\theta) := \frac{1}{N} \sum_{m=1}^{N} f(x_m^T \theta, y_m) - \frac{1}{N} f(x_n^T \theta, y_n) + \lambda R(\theta), \tag{A.10}
\]

which has $H(\theta) - (1/N)\nabla^2 f(x_n^T \theta, y_n)$ as its Hessian. Now consider approximating $\hat{\theta}\setminus n$ by performing a single Newton step on $F\setminus n$ starting from $\hat{\theta}$:

\[
\hat{\theta}\setminus n \approx \hat{\theta} - \left( H(\hat{\theta}) - \frac{1}{N} \nabla^2 f(x_n^T \hat{\theta}, y_n) \right)^{-1}
\]
\[
\times \left( \frac{1}{N} \sum_{m=1}^{N} \nabla f(x_m^T \hat{\theta}, y_m) - \frac{1}{N} \nabla f(x_n^T \hat{\theta}, y_n) + \lambda \nabla R(\hat{\theta}) \right). \tag{A.11}
\]

Using the fact that, by definition of $\hat{\theta}$, $(1/N) \sum_{n=1}^{N} \nabla f(x_n^T \hat{\theta}, y_n) + \lambda \nabla R(\hat{\theta}) = 0$, we
have that this simplifies to:

$$\hat{\theta}^n \approx \hat{\theta} + \frac{1}{N} \left( H(\hat{\theta}) - \frac{1}{N} \nabla_{\hat{\theta}}^2 f(x_n^T \hat{\theta}, y_n) \right)^{-1} \nabla_{\theta} f(x_n^T \hat{\theta}, y_n),$$

(A.13)

which is exactly $\tilde{\text{NS}}_n(R)$.

As $\tilde{\text{NS}}_n(R)$ can be interpreted as a single Newton step on the objective $F^n + \lambda R$, it follows that $\tilde{\text{NS}}_n(R)$ is exactly equal to $\hat{\theta}^n$ in the case that $F^n + \lambda R$ is a quadratic, as noted by Beirami et al. [2017]. For example, $\ell_2$ regularized linear regression has $\tilde{\text{NS}}_n(R) = \hat{\theta}^n$ for all $n$. We further note that somewhat similar behavior can hold for $\ell_1$ regularized linear regression. Specifically, when $\text{sign} \hat{\theta} = \text{sign} \hat{\theta}^n$, we have that the objective $F^n + \lambda \| \cdot \|_1$ is a quadratic when restricted to the dimensions in $\hat{S}$. In this case, $\text{NS}_n$ can be interpreted as taking a Newton step on $F^n + \lambda \| \cdot \|_1$ restricted to the dimensions in $\hat{S}$. It follows that $\text{NS}_n = \hat{\theta}^n$ when $\text{sign} \hat{\theta} = \text{sign} \hat{\theta}^n$ for $\ell_1$ regularized linear regression.

### A.4.5 Computation time of approximations

There is a major computational difference between Eq. (A.2) and Eq. (A.1): the former requires the inversion of a $D \times D$ matrix for each $\hat{\theta}^n$ approximated, while the latter requires a single $D \times D$ matrix inversion for all $\hat{\theta}^n$ inverted, which incurs a cost of $O(ND^3)$ versus a cost of $O(D^3)$. Even for small $D$, this is a significant additional expense.

However, as noted by Rad and Maleki [2020], Wang et al. [2018], Eq. (A.2) is much cheaper when considering the special case of generalized linear models. In this case, $\nabla_{\hat{\theta}}^2 f_n$ is some scalar times $x_n x_n^T$ – a rank one matrix. The Sherman-Morrison formula then allows us to cheaply compute the needed inverse in Eq. (A.2) given only $H^{-1}$; this is how Equation 8 in Rad and Maleki [2020] and Equation 21 in Wang et al. [2018] are derived. Even though we only consider GLMs in this work, we still study Eq. (A.1) with the hope of retaining scalability in more general problems.

### A.5 Derivation of $\text{IJ}_n$ and $\text{NS}_n$ via smoothed approximations

As noted in Section 2.2 Rad and Maleki [2020], Wang et al. [2018] derive the $\text{NS}_n$ approximation by considering $\tilde{\text{NS}}_n(R^\eta)$ with $R^\eta$ being some smoothed approximation to the $\ell_1$ norm, and then taking the limit of $\tilde{\text{NS}}_n(R^\eta)$ as the amount of smoothness goes to zero. We review this approach and then state our Proposition 12, which says that the same technique can be used to derive $\text{IJ}_n$.

We first give two possible ways to smooth the $\ell_1$ norm. The first is given by Rad
and Maleki [2020]:

\[ \|\theta\|_1 \approx R^\eta(\theta) := \sum_{d=1}^D \frac{1}{\eta} \left( \log(1 + e^{\eta \theta_d}) + \log(1 + e^{-\eta \theta_d}) \right). \quad (A.14) \]

The second option is to use the more general smoothing framework described by Wang et al. [2018]. They allow selection of a function \( q : \mathbb{R} \to \mathbb{R} \) satisfying: (1) \( q \) has compact support, (2) \( \int q(u) \, du = 1 \), \( q(0) > 0 \), and \( q \geq 0 \), and (3) \( q \) is symmetric around 0 and twice continuously differentiable on its domain, and then define a smoothed approximation:

\[ R^\eta(\theta) := \eta \sum_{d=1}^D \int_{-\infty}^{\infty} \abs{u} q(\eta (\theta_d - u)) \, du, \quad (A.15) \]

In both Eqs. (A.14) and (A.15), we have \( \lim_{\eta \to \infty} \|\theta\|_1 = \|\theta\|_1 \). Notice that either choice of \( R^\eta \) is twice differentiable for any \( \eta < \infty \), so one can consider the approximations \( \tilde{\text{NS}}_n(R^\eta), \tilde{\text{IJ}}_n(R^\eta) \). We now state two assumptions, both of which are given by Rad and Maleki [2020], Wang et al. [2018], under which one can show the limits of these approximations as \( \eta \to \infty \) are equal to \( \text{NS}_n \) and \( \text{IJ}_n \).

**Assumption 16.** For any element \( \hat{z} \in \mathbb{R}^D \) of the subdifferential \( \partial \|\theta\|_1 \) evaluated at \( \hat{\theta} \) such that \( \nabla F(\hat{\theta}) + \lambda \hat{z} = 0 \), we have \( \|\hat{z}\|_\infty < 1 \).

**Assumption 17.** For any \( y_n \in \mathbb{R} \), \( f(z, y_n) \) is a twice continuously differentiable function as a function of \( z \in \mathbb{R} \).

**Proposition 11** (Theorem 1 of Rad and Maleki [2020]; Theorem 4.2 of Wang et al. [2018]). Take Assumptions 16 and 17. Suppose \( H^\hat{\theta} \) has strictly positive eigenvalues. Let \( H^\hat{\theta} := H^\hat{\theta} - [\nabla^2_{\theta} f(x_n^T \hat{\theta}, y_n)]_{\hat{\theta}} \), and suppose that, for all \( n \), \( H^\hat{\theta} \) is invertible. Then, for \( R^\eta \) as in Eq. (A.14) or Eq. (A.15),

\[ \text{NS}_n := \lim_{\eta \to \infty} \tilde{\text{NS}}_n(R^\eta) = \left( \hat{\theta} + (H^\hat{\theta})^{-1} \left[ \nabla_{\theta} f(x_n^T \hat{\theta}, y_n) \right]_{\hat{\theta}} \right)_S. \quad (A.16) \]

As noted in the main text, we show that a very similar result holds for the limit of \( \tilde{\text{IJ}}_n(R^\eta) \):

**Proposition 12.** Take Assumptions 16 and 17. Suppose \( H^\hat{\theta} \) is invertible. Then for \( R^\eta \) as in Eq. (A.14) or Eq. (A.15):

\[ \text{IJ}_n := \lim_{\eta \to \infty} \tilde{\text{IJ}}_n(R^\eta) = \left( \hat{\theta} + H^\hat{\theta}^{-1} \left[ \nabla_{\theta} f(x_n^T \hat{\theta}, y_n) \right]_{\hat{\theta}} \right)_S. \quad (A.17) \]

The proof of Proposition 12 is a straightforward adaptation of the proof of Proposition 11. We prove it separately for the two different forms of \( R^\eta \) in the next two subsections.
A.5.1 Proof of Proposition 12 using Eq. (A.14)

This proof is almost identical to the proof of Theorem 1 from [Rad and Maleki 2020]. First we will need some notation. Let $\hat{\theta}^\eta$ be the solution to Eq. (2.1) using $R^\eta$ from Eq. (A.14) as the regularizer. Let $\hat{S}^\eta := \{ i : |\hat{\theta}^\eta| > c/\eta \}$ for some constant $c$. We know from the arguments in Appendix A.2 of [Rad and Maleki 2020] that for an appropriately chosen $c$ and $\eta > C$ for some large constant $C > 0$, we have $S^\eta = \hat{S} =: \text{supp } \hat{\theta}$. Next, define the scalars $\hat{D}^{(1,\eta)}_n, \hat{D}^{(2,\eta)}_n$ as the derivatives of $f$ evaluated at $\hat{\theta}^\eta$:

$$\hat{D}^{(1,\eta)}_n := \left. \frac{df(z,y_n)}{dz} \right|_{x=\hat{x}^T \hat{\theta}^\eta}, \quad \hat{D}^{(2,\eta)}_n := \left. \frac{d^2 f(z,y_n)}{dz^2} \right|_{x=\hat{x}^T \hat{\theta}^\eta}.$$

Finally, divide the Hessian of the smoothed problem up into blocks by defining:

$$A := X^T \hat{S} \text{diag } \{ \hat{D}^{(2,\eta)}_n \} X \hat{S} + \lambda \nabla^2 R^\eta(\hat{\theta}^\eta),$$

$$B := X^T \hat{S} \text{diag } \{ \hat{D}^{(2,\eta)}_n \} X \hat{S} + \lambda \nabla^2 R^\eta(\hat{\theta}^\eta),$$

$$C := X^T \hat{S} \text{diag } \{ \hat{D}^{(2,\eta)}_n \} X \hat{S} + \lambda \nabla^2 R^\eta(\hat{\theta}^\eta),$$

$$D := (A - BC^{-1}B^T)^{-1}.$$

We can then compute the block inverse of the Hessian of the smoothed problem, $H^{-1}_\eta$, as:

$$H^{-1}_\eta = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}^{-1} = \begin{pmatrix} D & -DBC^{-1} \\ -C^{-1}B^TD & A^{-1} + A^{-1}BDB^T A^{-1} \end{pmatrix}. \quad (A.19)$$

[Rad and Maleki 2020] show that all blocks of $H^{-1}_\eta$ converge to zero as $\eta \to \infty$ except for the upper left, which has $D \to X^T \hat{S} \text{diag } \{ \hat{D}^{(2)}_n \} X \hat{S}$. So, we have that the limit of $\tilde{J}_{n}(R^\eta)$ is:

$$\lim_{\eta \to \infty} \tilde{J}_{n}(R^\eta) = \lim_{\eta \to \infty} H^{-1}_\eta \hat{D}^{(1,\eta)}_n x_n = \hat{D}^{(1)}_n \begin{pmatrix} (X^T \hat{S} \text{diag } \{ \hat{D}^{(2)}_n \} X \hat{S})^{-1} & 0 \\ 0 & (x_n \hat{S}) \end{pmatrix}, \quad (A.20)$$

where we used that $\hat{\theta}^\eta \to \hat{\theta}$ by Lemma 15 of [Rad and Maleki 2020], which gives that $\hat{D}^{(1,\eta)}_n \to \hat{D}^{(1)}_n$ by Assumption 17. The resulting approximation is exactly that given in the statement of Proposition 12 by noting that $\hat{D}^{(1)}_n x_n \hat{S} = [\nabla f(x_n \hat{\theta}, y_n)] \hat{S}$.

A.5.2 Proposition 12 using Eq. (A.15)

This proof proceeds along the exact same direction as when using Eq. (A.14). In their proof of their Theorem 4.2, [Wang et al. 2018] provide essentially all the same
ingredients that Rad and Maleki [2020] do, except for the general class of smoothed approximations given by Eq. (A.15). This allows the same argument of taking the limit of each block of the Hessian individually and finishing by taking the limit as in Eq. (A.20).

### A.6 The importance of correct support recovery

Theorem 1 shows that each $\hat{\theta}^n$ having correct support (i.e., $\text{supp} \hat{\theta}^n = \text{supp} \theta^*$) is a sufficient condition for obtaining the fixed-dimensional error scaling shown in blue in Fig. 2-1. Here, we give some brief empirical evidence that this condition is necessary in the case of linear regression when using $\text{IJ}_n$ as an approximation. For values of $N$ ranging from 1,000 to 8,000, we set $D = N/10$ and generate a design matrix with i.i.d. $N(0,1)$ entries. The true $\theta^*$ is supported on its first five entries, with the rest set to zero. We then generate observations $y_n = x_n^T \theta^* + \varepsilon_n$, for $\varepsilon_n \sim i.i.d. N(0,1)$.

To examine what happens when the recovered supports are and are not correct, we use slightly different values of the regularization parameter $\lambda$. Specifically, the results of Wainwright [2009] (especially their Theorem 1) tell us that the support recovery of $\ell_1$ regularized linear regression will change sharply around $\lambda \approx 4 \sqrt{\log(D)/N}$, where lower values of $\lambda$ will fail to correctly recover the support. With this in mind, we choose two settings of $\lambda$: 1. $\lambda = 0.0 \sqrt{\log(D)/N}$ and 10. $\lambda = 10.0 \sqrt{\log(D)/N}$. As expected, the righthand side of Fig. A-2 shows that the accuracy of $\text{IJ}_n$ is drastically different in these two situations. The lefthand plot of Fig. A-2 offers an explanation for this observation: the support of $\text{supp} \hat{\theta}^n$ grows with $N$ under the lower value of $\lambda$, whereas the larger value of $\lambda$ ensures that $|\text{supp} \hat{\theta}^n| = |\text{supp} \theta^*| = \text{const}$. Empirically, these results suggest that, for high-dimensional problems, approximate CV methods are accurate estimates of exact CV only when taking advantage of some kind of low “effective dimensional” structure.

That the approximation quality relies so heavily on the exact setting of $\lambda$ is somewhat concerning. However, we emphasize that sensitivity exists for $\ell_1$ regularization in general; as previously noted, Wainwright [2009] demonstrated similarly drastic behavior of $\text{supp} \hat{\theta}$ in the same exact linear regression setup that we use here. On the other hand, Homrighausen and McDonald [2014] do show that using exact LOOCV to select $\lambda$ for $\ell_1$ regularized linear regression gives reasonable results. In Appendix A.8, we empirically show this is sometimes, but not always, the case for our and other approximate CV methods.

**Accuracy of approximate CV by optimization error.** In early experiments, we used the Python bindings for the glmnet package [Friedman et al., 2009] to solve our $\ell_1$ regularized problems. However, we found that both $\text{IJ}_n$ and $\text{NS}_n$ failed to recover the roughly $1/N^2$ scaling present in fixed-dimensional problems (e.g. as shown in Fig. 2-1 of Section 2.1) that we would expect given our theoretical results. We found that this was due to the relatively loose convergence tolerance with which glmnet is implemented (e.g. parameter changes of $\leq 1 \times 10^{-4}$ between iterations), which seems to be an issue for approximate CV methods and related approximations [Giordano].
Figure A-2: Illustration of the role of support recovery in the accuracy of IJ\_n in the case of linear regression. *Left*: Points show the average of |supp \( \hat{\theta} \|^n | \) over random values of \( n \). Error bars show the min and max |supp \( \hat{\theta} \|^n | \) over these \( n \). For \( \lambda = 10.0 \sqrt{\log(D)/N} \) (blue), the mean recovered support is constant with \( N \). For \( \lambda = 1.0 \sqrt{\log(D)/N} \) (red), |supp \( \hat{\theta} \|^n | \) grows with \( N \), and varies dramatically for different values of \( n \). *Right*: Percent error (Eq. (2.10)) as \( D \) scales with \( N \). When the support recovery is constant, we recover an error scaling of roughly \( 1/N^2 \), whereas a growing support results in a much slower decay.

We implemented our own \( \ell_1 \) solver in Python using many of the speed-ups proposed in Friedman et al. [2009] and set a convergence threshold of \( 1 \times 10^{-10} \) for the initial fit of \( \hat{\theta} \). This solver was used to produce all of our results, including Fig. A-2 which shows the expected roughly \( 1/N^2 \) accuracy of IJ\_n in blue.

### A.7 Details of real experiments

We use three publicly available datasets for our real-data experiments in Section 2.5:

1. The “Gisette” dataset [Guyon et al., 2004] is available from the UCI repository at [https://archive.ics.uci.edu/ml/datasets/Gisette](https://archive.ics.uci.edu/ml/datasets/Gisette). The dataset is constructed from the MNIST handwritten digits dataset. Specifically, the task is to differentiate between handwritten images of either “4” or “9.” There are \( N = 6,000 \) training examples, each of which has \( D = 5,000 \) features, some of which are junk “distractor features” added to make the problem more difficult.

2. The “bcTCGA” [bcTCGA, 2018] is a dataset of breast cancer samples from The Cancer Genome Atlas, which we downloaded from [http://myweb.uiowa.edu/pbreheny/data/bcTCGA.html](http://myweb.uiowa.edu/pbreheny/data/bcTCGA.html). The dataset consists of \( N = 536 \) samples of tumors, each of which has the real-valued expression levels of \( D = 17,322 \) genes. The task is to predict the real-valued expression level of the BRCA1 gene, which is known to correlate with breast cancer.

3. The “RCV1” dataset [Lewis et al., 2004] is a dataset of Reuters’ news articles given one of four categorical labels according to their subject: “Corporate/Industrial,” “Economics,” “Government/Social,” and “Markets.” We use a pre-processed binarized version from [https://www.csie.ntu.edu.tw/~cjlin/](https://www.csie.ntu.edu.tw/~cjlin/).
which combines the first two categories into a “positive” label and the latter two into a “negative” label. The full dataset contains $N = 20,242$ articles, each of which has $D = 47,236$ features. Running exact CV on this dataset would have been prohibitively slow, so we created a smaller dataset. First, the covariate matrix $X$ is extremely sparse (i.e., most entries are zero), so we selected the top 10,000 most common features and threw away the rest. We then randomly chose 5,000 documents to keep as our training set. After throwing away any of the 10,000 features that were not observed in this subset, we were left with a dataset of size $N = 5,000$ and $D = 9,836$.

In order to run $\ell_1$ regularized regression on each of these datasets, we first needed to select a value of $\lambda$. Since all of these datasets are fairly high dimensional, our experiments in Appendix A.8 suggests our approximation will be inaccurate for values of $\lambda$ that are “too small.” In an attempt to get the order of magnitude for $\lambda$ correct, we used the theoretically motivated value of $\lambda = C \sqrt{\log(D)/N}$ for some constant $C$ (e.g., Li et al. [2015] shows this scaling of $\lambda$ will recover the correct support for both linear and logistic regression). Section 2.5 suggests that the constant $C$ can be very important for the accuracy of our approximation, and our experiments there suggest that inaccuracy is caused by too large a recovered support size $|\text{supp} \hat{\theta}|$. For the RCV1 and Gisette datasets, both run with logistic regression, we guessed a value of $C = 1.5$, as this sits roughly in the range of values that give support recovery for logistic regression on synthetic datasets. After confirming that $|\text{supp} \hat{\theta}|$ was not too large (i.e., of size ten or twenty), we proceeded with these experiments. Although we found linear regression on synthetic data typically needed a larger value of $C$ than logistic regression on synthetic data, we found that $C = 1.5$ also produced reasonable results for the bcTCGA dataset.

A.8 Selection of $\lambda$

Our work in this chapter is almost exclusively focused on approximating CV for model assessment. However, this is not the only use-case of CV. CV is also commonly used for model selection, which, as a special case, contains hyperparameter tuning. Previous authors have used approximate CV methods for hyperparameter tuning in the way one might expect: for various values of $\lambda$, compute $\hat{\theta}$ and then use approximate CV to compute the out-of-sample error of each $\hat{\theta}$; the $\lambda$ leading to the lowest out-of-sample error is then selected [Obuchi and Kabashima 2016, 2018, Beirami et al. 2017, Rad and Maleki 2020, Wang et al. 2018, Giordano et al. 2019b]. While many of these authors theoretically study the accuracy of approximate CV, we note that they only do so in the context of model assessment and only empirically study approximate CV for hyperparameter tuning. In this appendix, we add to these experiments by showing that approximate CV can exhibit previously undemonstrated complex behavior when used for hyperparameter tuning.

We generate two synthetic $\ell_1$ regularized logistic regression problems with $N = 300$ observations and $D = \{75,150\}$ dimensions. The matrix of covariates $X$ has i.i.d.
entries, and the true $\theta^*$ has its first five entries drawn i.i.d. as $N(0, 1)$ with the rest set to zero. As a measure of the true out of sample error, we construct a test set with ten thousand observations. For a range of values of $\lambda$, we find $\hat{\theta}$, and measure the train, test, exact LOOCV, and approximate LOOCV errors via both $\text{NS}_n$ and $\text{IJ}_n$; the results are plotted in Fig. A-3. $\text{NS}_n$ (blue dashed curve) is an extremely close approximation to exact CV (red curve) in both datasets and selects a $\lambda$ that gives a test error very close to the $\lambda$ selected by exact CV. On the other hand, $\text{IJ}_n$ (solid blue curve) performs very differently on the two datasets. For $D = 75$, it selects a somewhat reasonable value for $\lambda$; however, for $D = 150$, $\text{IJ}_n$ goes disastrously wrong by selecting the obviously incorrect value of $\lambda = 0$. While the results in Fig. A-3 come from using our $\text{IJ}_n$ to approximate CV for an $\ell_1$ regularized problem, we note that this issue is not specific to the current work; we observed similar behavior when using $\ell_2$ regularization and the pre-existing $\tilde{\text{IJ}}_n$ ($\ell_2$).

While $\text{NS}_n$ performs far better than $\text{IJ}_n$ in the experiments here, it too has a limitation when $D > N$. In particular, when $\lambda$ is small enough, we will eventually recover $|\hat{S}| = N$. At this point, the matrix we need to invert in the definition of $\text{NS}_n$ in Eq. (2.5) will be a $N \times N$ matrix that is the sum of $N - 1$ rank-one matrices. As such, it will not be invertible, meaning that we cannot compute $\text{NS}_n$ for small $\lambda$ when $D > N$. Even when $D$ is less than – but still close to – $N$, we have observed numerical issues in computing $\text{NS}_n$ when $\lambda$ is sufficiently small; typically, these issues show up as enormously large values for ALOO for small values of $\lambda$.

Given the above discussion, we believe that an understanding of the behavior of $\text{IJ}_n$ and $\text{NS}_n$ for the purposes of hyperparameter tuning is a very important direction for future work.

### A.9 Proofs from Section 2.4

As mentioned in the main text, there exist somewhat general assumptions in the $\ell_1$ literature under which $\text{supp} \hat{\theta} = S$ [Lee et al., 2014; Li et al., 2015]. By taking these assumptions for all leave-one-out problems, we immediately get that $\text{supp} \hat{\theta}^n = S$ for all $n$. Our method for proving Theorems 2 and 3 will be to show that the assumptions of those theorems imply those from the $\ell_1$ literature for all leave-one-out problems.

#### A.9.1 Assumptions from Li et al. [2015]

We choose to use the conditions from Li et al. [2015], as we find them easier to work with for our problem. Li et al. [2015] gives conditions on $F = (1/N) \sum_n f(x_n^T \theta, y_n)$ under which $\text{supp} \hat{\theta} = S$. We are interested in $\text{supp} \hat{\theta}^n$, so we state versions of these conditions for $F^n := (1/N) \sum_{m: m \neq n} f(x_m^T \theta, y_n)$.

**Assumption 18 (LSSC).** $\forall n, F^n$ satisfies the $(\theta^*, \mathbb{R}^D)$ locally structured smoothness condition (LSSC) with constant $K$. We recall this condition, due to Li et al. [2015].

---

1 Readers familiar with the LSSC may see choosing the neighborhood of $\theta^*$ as $\mathbb{R}^D$ to be too restrictive. This choice is not necessary for our results; we state Assumption 18 this way only for...
Figure A-3: Experiment for selecting $\lambda$ from Appendix A.8. (Top:) Despite being very accurate for higher values of $\lambda$, the degradation of the accuracy of IJ\(_n\) for lower values of $\lambda$ (which corresponds to a larger $\hat{S}$) causes the selection of a $\lambda$ that is far from optimal in terms of test loss. (Bottom:) For a lower dimensional problem, the curve constructed by IJ\(_n\) much more closely mirrors that of exact CV for all values of $\lambda$. In both cases, NS\(_n\) performs well.

**Assumption 19** (Strong convexity). For a matrix $A$, let $\lambda_{\text{min}}(A)$ be the smallest eigenvalue of $A$. Then, $\forall n$ and for some constant $L_{\text{min}}$, the Hessian of $F^{\alpha}$ is positive definite at $\theta^*$ when restricted to the dimensions in $S$: $\lambda_{\text{min}}(\nabla^2 F^{\alpha}(\theta^*)_{SS}) \geq L_{\text{min}} > 0$.

**Assumption 20** (Incoherence). $\forall n$ and for some $\gamma > 0$,

$$\left\| \nabla F^{\alpha}(\theta^*)_{S^c,S} \left( \nabla^2 F^{\alpha}(\theta^*)_{SS} \right)^{-1} \right\|_{\infty} < 1 - \gamma.$$ (A.21)

**Assumption 21** (Bounded gradient). For $\gamma$ from Assumption 20, $\forall n$, the gradient of $F^{\alpha}$ evaluated at the true parameters $\theta^*$ is small relative to the amount of regularization: $\left\| \nabla F^{\alpha}(\theta^*) \right\|_{\infty} \leq (\gamma/4)\lambda$.

**Assumption 22** ($\lambda$ sufficiently small). For $K, L_{\text{min}}$ and $\gamma$ as in Assumptions 18 to 20, the regularization parameter is sufficiently small: $\lambda < L_{\text{min}}^2 \gamma/(4(\gamma + 4)^2 D_{\text{eff}}K)$, where there is no constraint on $\lambda$ if $K = 0$. 

in Appendix A.9.2
We see in Appendix A.9.3 that a minor adaptation of Theorem 5.1 from Li et al. [2015] tells us that Assumptions 18 to 22 imply $\forall n, \text{supp } \hat{\theta}_n \subseteq S$. To prove the accuracy of $\text{NS}_n$ and $\text{IJ}_n$, though, we further need that $\text{supp } \hat{\theta}_n \subseteq \hat{S}$ so that all LOOCV problems run over the same low-dimensional space as the full-data problem. It will be easier to state conditions for a stronger result, that $\text{supp } \hat{\theta}_n = \hat{S} = S$. This will follow from an assumption on the smallest entry of $\hat{\theta}_n$, which we stated as Assumption 4 in the main text. We stated Assumption 4 using the quantity $T_{\text{min}}$ to avoid stating Assumptions 19 and 20 in the main text. We can now state its full version.

**Assumption 23** (full version of Assumption 4). For $L_{\text{min}}$ and $\gamma$ from Assumptions 19 and 20, \[ \min_{s \in S} |\theta^*_s| > \left( \sqrt{D_{\text{eff}}(\gamma + 4)/L_{\text{min}}} \right) \lambda. \] 

**Proposition 13.** If Assumptions 18 to 23 hold, then $\forall n, \text{supp } \hat{\theta}_n = \hat{S} = S$.

**Proof.** This is immediate from Theorem 5.1 of Li et al. [2015]. □

### A.9.2 Local structured smoothness condition (LSSC)

We now define the local structured smoothness condition (LSSC). The LSSC was introduced by Li et al. [2015] for the purpose of extending proof techniques for the support recovery of $\ell_1$ regularized linear regression to more general $\ell_1$ regularized $M$-estimators. Essentially, it provides a condition on the smoothness of the third derivatives of the objective $F(\theta)$ near the true sparse $\theta^*$. One can then analyze a second order Taylor expansion of the loss and use the LSSC to show that the remainder in this expansion is not too large. To formalize the LSSC, we need to define the third order derivative of $F$ evaluated along a direction $u \in \mathbb{R}^D$:

$$D^3 F(\theta)[u] := \lim_{t \to 0} \frac{\nabla^2 F(\theta + tu) - \nabla^2 F(\theta)}{t}.$$ 

In the cases considered in this chapter, this is just a $D \times D$ matrix. We can then naturally define the scalar $D^3 F(\theta)[u, v, w]$ as an outer product on this matrix:

$$D^3[u, v, w] := v^T (D^3 F(\theta)[u]) w$$

**Definition 7** (LSSC). Let $F : \mathbb{R}^D \to \mathbb{R}$ be a continuously three-times differentiable function. For $\theta^* \in \mathbb{R}^D$ and $N_{\theta^*} \subseteq \mathbb{R}^D$, the function $F$ satisfies the $(\theta^*, N_{\theta^*})$ LSSC with constant $K \geq 0$ if for any $u \in \mathbb{R}^D$:

$$|D^3 f(\theta^* + \delta)[u, v, e_j]| \leq K \|u\|^2,$$  \hspace{1cm} (A.22)

where $e_j \in \mathbb{R}^D$ is the $j$th coordinate vector, and $\delta \in \mathbb{R}^D$ is any vector such that $\theta^* + \delta \in N_{\theta^*}$.

We note that this definition is actually different from the original definition given in Li et al. [2015], who prove the two to be equivalent in their Proposition 3.1. Li
et al. [2015] go on to prove bounds on the LSSC constants for linear and logistic regression, which we state as Proposition 19 and Proposition 21 below.

Note that Assumption 18 in the main text states that the LSSC holds with $N_{\theta^*} = R_D$. We state Assumption 18 in this form purely for conciseness; we will only consider checking Assumption 18 for linear and logistic regression, both of which satisfy the LSSC with $N_{\theta^*} = R_D$. Going beyond these cases, it is easily possible to state a version of our results with $N_{\theta^*} \neq R_D$; however, this will require an extra assumption along the lines of Condition 7 of Theorem 5.1 in Li et al. [2015], which is trivially satisfied when $N_{\theta^*} = R_D$. In order to avoid stating an extra assumption that is trivially satisfied in the cases we consider, we chose to simply state the LSSC with $N_{\theta^*} = R_D$.

A.9.3 Assumptions 18 to 22 imply $\text{supp} \hat{\theta} \setminus n \subseteq S$ for all $n$

Theorem 5.1 of Li et al. [2015] gives conditions on $F$ under which $\text{supp} \hat{\theta} = S$. So, if these conditions hold for all $F \setminus n$, then we have $\text{supp} \hat{\theta} \setminus n = S$ for all $n$. Their Theorem 5.1 actually has two extra assumptions beyond Assumptions 18 to 22. The first is their Assumption 7; however, this is immediately implied by the fact that we assume the LSSC holds with $N_{\theta^*} = R_D$. The second is their analogue of our Assumption 23; however, they use this condition to imply that $\hat{\theta} = S$ after having shown that $\hat{\theta} \subseteq S$.

A.9.4 Useful results for proving Theorems 2 and 3

Before going on to Theorems 2 and 3, we will give a few useful results. We first define a sub-Exponential random variable:

**Definition 8 (Vershynin [2018]).** A random variable $V$ is $c_x$-sub-Exponential if

$$E[\exp(V/c_x)] \leq 2.$$  \hfill (A.23)

We will frequently use the fact that if $X$ is $c_x$-sub-Gaussian, then $X^2$ is $c_x^2$-sub-Exponential. Now we state a few existing results about the maxima of sub-Gaussian and sub-Exponential random variables that will be useful in our proofs.

**Lemma 3 (Lemma 5.2 from van Handel [2016]).** Suppose that we have real valued random variables $Z_1, \ldots, Z_N$ that satisfy $\log E[e^{\lambda Z_n}] \leq \psi(\lambda)$ for all $n = 1, \ldots, N$ and all $\lambda \geq 0$ for some convex function $\psi : \mathbb{R} \rightarrow \mathbb{R}$ with $\psi(0) = \psi'(0) = 0$. Then for any $u \geq 0$:

$$\Pr \left[ \max_{n=1,\ldots,N} Z_n \geq \psi^{-1}(\log N + u) \right] \leq e^{-u}.$$  

where $\psi^{-1}$ is the inverse of the Legendre dual of $\psi$.

Remembering the definition of a sub-Gaussian random variable from Definition 1, Lemma 3 can be used to show the following:
**Corollary 4.** Let $Z_1, \ldots, Z_N$ be i.i.d. sub-Gaussian random variables with parameter $c_x$. Then:

\[
\Pr \left[ \max_{n=1,\ldots,N} Z_n \geq E[Z_n] + \sqrt{2Cc_x^2 \log N} + u \right] \leq e^{-\frac{u^2}{2C^2}} \quad (A.24)
\]

\[
\Pr \left[ \max_{n=1,\ldots,N} Z_n^2 \geq E[Z_n^2] + Cc_x^2 \log (N+1) + u \right] \leq e^{-u} \quad (A.25)
\]

**Proof.** For the first inequality, the definition of a sub-Gaussian random variable is that \( \log \mathbb{E} e^{\lambda Z_n} \leq \lambda^2 c_x^2 / 2 =: \psi(\lambda) \), which has \( \psi^*(-y) = y^2 / (2Cc_x^2) \) and \( \psi^*(x) = \sqrt{2Cc_x^2} x \).

We use the upper bound:

\[
\psi^*(-(\log N + u)) = \sqrt{2Cc_x^2 (\log N + u)} \leq \sqrt{2Cc_x^2 \log N} + \sqrt{2Cc_x^2 u},
\]

Using this upper bound with Lemma 3 and changing variables \( u \mapsto u^2 / (2Cc_x^2) \) gives the first inequality.

For the second inequality, use the fact that \( Z_n^2 \) is sub-Exponential with parameter \( c_x^2 \) so that it satisfies \( \log \mathbb{E} e^{\lambda Z_n^2} \leq \psi(\lambda) \), where:

\[
\psi(\lambda) := \begin{cases} 
\lambda Cc_x^2, & 0 \leq t \leq 1/c_x^2 \\
\infty, & \text{o.w.}
\end{cases}
\]

For \( x \geq 0 \), this \( \psi \) has inverse Legendre dual \( \psi^*(x) = Cc_x^2 (x+1) \). Plugging into Lemma 3 gives the result. \( \square \)

**Proposition 14.** Let \( x_1, \ldots, x_N \) be random vectors in \( \mathbb{R}^D \) with i.i.d. \( c_x \)-sub-Gaussian components and \( E[x_n^2] = 1 \). Then:

\[
\Pr \left[ \max_{n=1,\ldots,N} \|x_n\|_2 \geq \sqrt{D} + \sqrt{2Cc_x^2 \log N} + u \right] \leq e^{-\frac{u^2}{2C^2}}, \quad (A.26)
\]

where \( C > 0 \) is some global constant, independent of \( c_x, D, \) and \( N \).

**Proof.** From Theorem 3.1.1 of Vershynin [2018], we have that \( \|x_n\|_2 - \sqrt{D} \) is sub-Gaussian with parameter \( CC_x^2 \), where \( C \) is some constant. Using the first part of Corollary 4 gives the result. \( \square \)

### A.9.5 Proof of Theorem 2 (Linear Regression)

Recall Assumptions 1 and 5, we assume a linear regression model \( y_n = x_n^T \theta^* + \varepsilon_n \), where \( x_n \in \mathbb{R}^D \) has i.i.d. \( c_x \)-sub-Gaussian components with \( E[x_n^2] = 1 \) and \( \varepsilon_n \) is \( c_\varepsilon \)-sub-Gaussian. For notation throughout this section, we will let \( C \) denote an absolute constant independent of any aspect of the problem \( (N, D, D_{\text{eff}}, c_x, \text{ or } c_\varepsilon) \) that will change from line to line (e.g. we may write \( 5C^2 = C \)). We will frequently use \( X_S \) to denote the \( N \times D_{\text{eff}} \) matrix formed by taking the columns of \( X \) that are in \( S \), \( x_{nS} \) to denote the coordinates of the \( n \)th vector of covariates \( x_n \) that are in the set \( S \),
and \( X_{\setminus n,S} \) to denote the matrix \( X_S \) with the \( n \)th row removed. We will show the following theorem, stated more concisely as Theorem 2 in the main text:

**Theorem 6** (Restated version of Theorem 2 from main text). Take Assumptions 1 to 3, 5 and 23. Suppose the regularization parameter \( \lambda \) satisfies:

\[
\lambda \geq \frac{1}{\alpha - M_{\text{lin}}} \sqrt{\frac{c_x^2 c_{\varepsilon}^2 \log D}{NC} + \frac{25 c_x^2 c_{\varepsilon}^2}{NC} + \frac{4 c_x c_{\varepsilon} (\log(ND) + 26)}{N(\alpha - M_{\text{lin}})}},
\]

\( (A.27) \)

where \( C \) is a constant in \( N, D, D_{\text{eff}}, c_x \) and \( c_{\varepsilon} \), and \( M_{\text{lin}} \) is defined as:

\[
M_{\text{lin}} = CD_{\text{eff}} \left( \sqrt{50 c_x^2} + \sqrt{2 c_x^2 \log(N(D - D_{\text{eff}}))} \right) \left( \sqrt{D_{\text{eff}} + \sqrt{50 c_x^4} + \sqrt{2 c_x^4 \log(N)}} \right)
+ \frac{CD_{\text{eff}} (D_{\text{eff}} + D_{\text{eff}} c_x^2 (\log N + 26))}{(N - 3 c_x^2 \sqrt{N(\sqrt{D_{\text{eff}}} + 5)})^2}
\times \left( \sqrt{N} + \sqrt{50 c_x^4} + \sqrt{2 c_x^4 \log(D - D_{\text{eff}})} \right) \left( \sqrt{ND_{\text{eff}}} + \sqrt{50 c_x^4} \right)
\]

\( (A.28) \)

Then for \( N \) sufficiently large, Condition 1 holds with probability at least \( 1 - 26 e^{-25} \), where the probability is over the random data \( \{(x_n, y_n)\}_{n=1}^N \).

**Proof.** For a fixed regularization parameter \( \lambda \) and random data \( \{x_n, y_n\}_{n=1}^N \), we are interested in the probability that any of Assumptions 18 to 22 are violated, as Proposition 13 then proves the result. For convenience in writing the incoherence condition, define \( J_{nd} \in \mathbb{R}^D \), for \( d \in S^c \), as:

\[
J_{nd} := (X_{\setminus n,S}^T X_{\setminus n,S})^{-1} X_{\setminus n,S}^T X_{\setminus n,d}.
\]

\( (A.29) \)

It is easiest to show that each of Assumptions 18 to 22 hold with high probability separately, rather than all together, so we apply a union bound to get:

\[
\Pr[\text{any assumption violated}] \leq \\
\Pr \left[ \min_n \lambda_{\text{min}}(X_{\setminus n,S}^T X_{\setminus n,S}) = 0 \right] \\
+ \Pr \left[ \max_n \max_{d \in S^c} \|J_{nd}\|_1 \geq 1 \right] \\
+ \Pr \left[ \max_n \|\nabla F_{\setminus n}\|_\infty > \frac{\lambda (1 - \max_n \max_{d \in S^c} \|J_{nd}\|_1)}{4} \right] \\
+ \Pr \left[ \min_n \lambda_{\text{min}}^2(X_{\setminus n,S}^T X_{\setminus n,S}) \left( \frac{1 - \max_n \max_{d \in S^c} \|J_{nd}\|_1}{4 (1 - \max_n \max_{d \in S^c} \|J_{nd}\|_1) + 4} \right)^2 \frac{D_{\text{eff}} K}{4} \right] \leq \lambda
\]

We will bound each term by appealing to the Lemmas and Propositions proved below. Using Lemma 4 and Lemma 6, the first and third terms are bounded by \( 16^{-25} \). As
noted in Proposition 19, we have \( \Pr[K = 0] = 1 \), so the final probability is equal to zero (as the event reduces to \( \infty < \lambda \)). To bound the second probability, we have that Lemma 5 says that:

\[
\Pr\left[ \max_n \max_{d \in S^n} \| J_{nd} \|_1 \geq 1 - \alpha + M_{\text{lin}} \right] \leq 9e^{-25}.
\]

As \( \alpha > 0 \), if \( M_{\text{lin}} = o(1) \) as \( N \to \infty \), we will have that \( 1 - \alpha + M_{\text{lin}} < 1 \) for large enough \( N \). This would imply the third probability is \( \leq 9e^{-25} \) for \( N \) large enough.

Under our conditions on the growth of \( D_{\text{eff}} \) and \( D \), we can show that \( M_{\text{lin}} = o(1) \).

We have, hiding constants and lower order terms in \( N, D, \) and \( D_{\text{eff}} \):

\[
M_{\text{lin}} = O \left( \frac{D_{\text{eff}} \sqrt{\log(N)} + \log(D)}{N - \sqrt{ND_{\text{eff}}}} + \frac{D_{\text{eff}}^{5/2} \log(N) \left( \sqrt{N} + \log(D) \right) \sqrt{N}}{(N - \sqrt{ND_{\text{eff}}})^2} \right)
\]

\[
= O \left( \frac{D_{\text{eff}} \left( \sqrt{D_{\text{eff}} \log(N)} + \sqrt{D_{\text{eff}} \log(D)} \right)}{N - \sqrt{ND_{\text{eff}}}} \right)
\]

where the second statement follows from using \( \sqrt{\log(N)} + \log(D) \leq \sqrt{\log(N)} + \sqrt{\log(D)} \) and \( D = o(e^N) \). Now, given that \( D_{\text{eff}} = o([N/ \log(N)]^{2/5}) \), the second term in Eq. (A.30) is \( o(1) \). The first term is also \( o(1) \) by combining \( D_{\text{eff}} = o([N/ \log(N)]^{2/5}) \) with \( D_{\text{eff}}^{5/2} \sqrt{\log(D)} = o(N) \). Thus, \( M_{\text{lin}} = o(1) \), which completes the proof.

What remains is to prove Lemmas 4 to 6 and Proposition 19 needed to prove Theorem 6. We do this in the following four subsections.

### A.9.6 Linear regression: minimum eigenvalue

All we want to bound right now is the probability that the minimum eigenvalue is actually equal to zero; however, it will be useful later to show that it is \( \Omega(N) \) with high probability. The lemma we prove in this section shows exactly this. We will start with two propositions.

**Proposition 15.** If \( X_{:,S} \) is an \( N \times D_{\text{eff}} \) matrix with independent \( c_x \)-sub-Gaussian entries with unit second moments, then:

\[
\Pr\left[ \lambda_{\text{min}}(X_{:,S}^T X_{:,S}) \leq N - 2C c_x^2 \sqrt{N(D_{\text{eff}} + 5)} \right] \leq 2e^{-25},
\]

where \( C > 0 \) is a global constant.
Proof. Theorem 4.6.1 of [Vershynin 2018] gives a concentration inequality for the minimum singular value, $s_{\min}(X,S)$, of $X,S$:

$$\Pr \left[ s_{\min}(X,S) \leq \sqrt{N} - Cc_x^2(\sqrt{D_{\text{eff}}} + t) \right] \leq 2e^{-t^2}. \quad (A.32)$$

Using the fact that the minimum eigenvalue of $X^T S X$ is the square of the minimum singular value of $X,S$ and putting in $t = 5$:

$$\Pr \left[ \lambda_{\min}(X^T S X) \leq N - 2Cc_x^2\sqrt{N(\sqrt{D_{\text{eff}}} + 5)} + C^2c_x^4(\sqrt{D_{\text{eff}}} + 5)^2 \right] \leq 2e^{-25}. \quad (A.33)$$

Dropping the $C^2c_x^4(\sqrt{D_{\text{eff}}} + 5)^2$ gives the result. \qed

**Proposition 16.** If $X_{\setminus n,S}$ is the $N - 1 \times D_{\text{eff}}$ matrix formed by removing the $n$th row from $X_S$, we have:

$$\lambda_{\min}(X_{\setminus n,S}^T X_{\setminus n,S}) \geq \lambda_{\min}(X_S^T X_S) - \|x_nS\|_2^2, \quad (A.34)$$

where $x_n$ is the $n$th row of $X_S$.

Proof. Looking at the variational characterization of the minimum eigenvalue:

$$\lambda_{\min}(X_{\setminus n,S}^T X_{\setminus n,S}) = \min_{z \in \mathbb{R}^{D_{\text{eff}}}, \|z\|_2 = 1} \left[ z^T X_{\setminus n,S}^T X_{\setminus n,S} z - z^T x_n S x_n^T S z \right]$$

$$\geq \min_{z} z^T X_{\setminus n,S}^T X_{\setminus n,S} z - \max_{z} z^T x_n S x_n^T S z$$

$$= \lambda_{\min}(X_S^T X_S) - \|x_nS\|_2^2. \quad (A.35)$$

The above two propositions now allow us to prove the bound we want on $\min_n \lambda_{\min}(X_{\setminus n,S}^T X_{\setminus n,S})$. In the following lemma, we will assume that $D_{\text{eff}} = o(N \log(N))$. While we ultimately will have the more restrictive requirement that $D_{\text{eff}} = o([N \log(N)]^{2/5})$ in Assumption 3, the current result can be stated with the less restrictive requirement of $o(N \log(N))$.

**Lemma 4.** Suppose $X_S$ is a $N \times D_{\text{eff}}$ matrix with independent $c_x$-sub-Gaussian entries and $D_{\text{eff}}$ is $o(N \log(N))$ as function of $N$. Then we have for $N$ sufficiently large:

$$\Pr \left[ \min_{n=1,\ldots,N} \lambda_{\min}(X_{\setminus n,S}^T X_{\setminus n,S}) \leq N - 3Cc_x^2\sqrt{N(\sqrt{D_{\text{eff}}} + 5)} \right] \leq 3e^{-25} \quad (A.36)$$

Proof. In what follows, and repeatedly throughout the rest of our proofs, we will make use of the following generic inequality for any events $A$ and $B$:

$$\Pr[A] = \Pr[A \mid B] \Pr[B] + \Pr[A \mid B^c] \Pr[B^c] \leq \Pr[A \mid B] \Pr[B] + \Pr[B^c] \quad (A.37)$$

113
Calling the probability on the left hand side of Eq. (A.34) $P$, we can break $P$ down as, for some constant $L_{\text{min}}$:

\[ P \leq \]

\[ \Pr \left[ \min_{n=1,\ldots,N} \lambda_{\text{min}}(X_{n,S}^T X_{n,S}) \leq N - 3Cc_x^2 \sqrt{N} \left( \sqrt{D_{\text{eff}}} + 5 \right) \mid \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \geq L_{\text{min}} \right] \]

\[ \times \Pr \left[ \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \geq L_{\text{min}} \right] + \Pr \left[ \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \leq L_{\text{min}} \right] \]

\[ \leq \Pr \left[ \min_{n=1,\ldots,N} \|x_{nS}\|_2 \leq N - 3Cc_x^2 \sqrt{N} \left( \sqrt{D_{\text{eff}}} + 5 \right) \mid \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \geq L_{\text{min}} \right] \]

\[ \times \Pr \left[ \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \geq L_{\text{min}} \right] + \Pr \left[ \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \leq L_{\text{min}} \right] \]

\[ \leq \Pr \left[ \max_n \|x_{nS}\|_2 \geq L_{\text{min}} - N + 3Cc_x^2 \sqrt{N} \left( \sqrt{D_{\text{eff}}} + 5 \right) \right] \]

\[ + \Pr \left[ \lambda_{\text{min}}(X_{S,S}^T X_{S,S}) \leq L_{\text{min}} \right] \]

Picking $L_{\text{min}} = N - 2Cc_x^2 \sqrt{N} \left( \sqrt{D_{\text{eff}}} + 5 \right)$, we have that the second probability at most $2e^{-25}$ by Proposition 16. Now to control the $\max_n \|x_{nS}\|_2$, note that $\|x_{nS}\|_2$ is $D_{\text{eff}}c_x^2$-sub-Exponential, and choose $u = 25$ in the second statement of Corollary 4; this tells us that the first probability is at most $e^{-25}$ if $E[\|x_{nS}\|_2^2] + Cc_x^2 (\log N + 26) = D_{\text{eff}} + Cc_x^2 (\log N + 26)$ is less than $Cc_x^2 \sqrt{N} \left( \sqrt{D_{\text{eff}}} + 5 \right)$, which, for $D_{\text{eff}}$ being $o(\sqrt{\log(N)})$, is satisfied for $N$ large enough.

\[ \square \]

**A.9.7 Linear regression: incoherence**

The following proposition will be useful in proving Lemma 5 below:

**Proposition 17.** Let $z \in \mathbb{R}^N$ be any vector and $z_n \in \mathbb{R}^{N-1}$ the same vector with the $n$th coordinate removed. Also let $X_{S} \in \mathbb{R}^{N\times D_{\text{eff}}}$ be some matrix with $X_{n,S}$ the same matrix with the $n$th row removed. Define, for any vector $z \in \mathbb{R}^N$,

\[ J_{nz} := (X_{n,S}^T X_{n,S})^{-1} X_{n,S}^T z_n, \]  

(A.36)

and $J_z$ the same but with no row removed. Then:

\[ \|J_{nz} - J_z\|_1 \leq D_{\text{eff}} \frac{|z_n| \|x_{nS}\|_2}{\lambda_{\text{min}}(X_{n,S}^T X_{n,S})} \]

\[ + D_{\text{eff}} \frac{\|x_{nS}\|_2^2}{\lambda_{\text{min}}^2(X_{n,S}^T X_{n,S})} \|z\|_2 \|X_{S,S}\|_2, \]

where $\|X_{S,S}\|_2 := \sqrt{\sum_{n=1}^{N} \sum_{s \in S} X_{ns}^2}$.

**Proof.** We can rewrite $J_z = (X_{S,S}^T X_{S,S})^{-1} X_{S,S} z$ by noting that $X_{S,S}^T X_{S,S}$ and $X_{n,S}^T X_{n,S}$
differ by a rank one update and then applying the Sherman-Morrison formula:

\[ J_z = (X_{:,s}^T X_{:,s})^{-1} X_{:,s}^T z \]

\[ = \left( (X_{n,S}^T X_{n,S})^{-1} - \frac{(X_{n,S}^T X_{n,S})^{-1} x_{nS} x_{nS}^T (X_{n,S}^T X_{n,S})^{-1}}{1 + x_{nS}^T (X_{n,S}^T X_{n,S})^{-1} x_{nS}} \right) X_{:,s}^T z \]

\[ = (J_{nz} + (X_{n,S}^T X_{n,S})^{-1} x_{nS} z_n) - \frac{(X_{n,S}^T X_{n,S})^{-1} x_{nS} x_{nS}^T (X_{n,S}^T X_{n,S})^{-1}}{1 + x_{nS}^T (X_{n,S}^T X_{n,S})^{-1} x_{nS}} X_{:,s}^T z \]

To cleanup notation a bit, let \( B := X_{n,S}^T X_{n,S} \). We can continue to rewrite the above as:

\[ = (J_{nz} + B^{-1} x_{nS} z_n) - \frac{B^{-1} x_{nS}}{1 + x_{nS}^T B^{-1} x_{nS}} \sum_{m=1}^{N} z_m x_{mS}^T B^{-1} x_{mS} \]

(A.40)

Now, we are interested in \( \|J_{nz} - J_z\|_1 \), which we will bound by subtracting \( J_{nz} \) from both sides of the above equation and then examine each coordinate by multiplying by the \( i \)th unit vector \( e_i \):

\[ |e_i^T (J_{nz} - J_z)| \leq |e_i^T B^{-1} x_{nS}| |z_n| + \frac{|e_i^T B^{-1} x_{nS}|}{1 + x_{nS}^T B^{-1} x_{nS}} \sum_{m=1}^{N} z_m |x_{mS}^T B^{-1} x_{mS}| \]

\[ \leq |z_n| \lambda_{\text{max}}(B^{-1}) \|x_{nS}\|_2^2 + \frac{\lambda_{\text{max}}^2(B^{-1}) \|x_{nS}\|_2^2}{1 + \lambda_{\text{min}}(B^{-1}) \|x_{nS}\|_2^2} \sum_{m=1}^{N} |z_m| \|x_{mS}\|_2 \]

(A.42)

The \( \lambda_{\text{min}}(B^{-1}) \|x_{nS}\|_2^2 \) is strictly positive, so we can drop it from the denominator for a further upper bound. Using the fact that, for the positive semidefinite matrix \( B \) we have \( \lambda_{\text{min}}(B^{-1}) = 1/\lambda_{\text{max}}(B) \) and \( \lambda_{\text{max}}(B^{-1}) = 1/\lambda_{\text{min}}(B) \), we get:

\[ |e_i^T (J_{nz} - J_z)| \leq \frac{|z_n| \|x_{nS}\|_2^2}{\lambda_{\text{min}}(B)} + \frac{\|x_{nS}\|_2^2}{\lambda_{\text{min}}^2(B)} \sum_{m=1}^{N} |z_m| \|x_{mS}\|_2^2 \cdot \]

(A.43)

Finally, use Cauchy-Schwarz to get \( \sum_{m=1}^{N} |z_m| \|x_{mS}\|_2 \leq \|z\|_2 \|X_{:,s}\|_2 \), where \( \|X_{:,s}\|_2 := \left( \sum_{m=1}^{N} \sum_{s \in S} x_{ms}^2 \right)^{1/2} \). Notice that our upper bound is now independent of the index \( i \); this means we have a bound on any coordinate \( i \) of \( |(J_{nz} - J_z)| \). So, multiplying this bound by \( D_{\text{eff}} \) upper bounds \( \|J_{nz} - J_z\|_1 \), which gives the result.

To get a high probability upper bound on \( \|J_{nd}\|_1 \), the idea will be to use \( \|J_{nd}\|_1 \leq \|J_d\|_1 + \|J_{nd} - J_d\|_1 \), and then put high probability bounds on the bound given by Proposition 17.

\[ \text{Lemma 5. Take Assumptions 1, 2 and 5. Then, for the scalar } M_{\text{lin}} \text{ defined in} \]
Theorem 2, we have:

\[
\Pr \left[ \max_{n=1, \ldots, N} \|J_{nd}\|_1 \geq 1 - \alpha + M_{\text{lin}} \right] \leq 10e^{-25}, \tag{A.44}
\]

where \(J_{nd}\) is defined in Eq. (A.29) above.

Proof. First, for any \(n \text{ and } d\), we have \(\|J_{nd}\|_1 \leq \|J_d\|_1 + \|J_{nd} - J_d\|_1\). We can upper bound \(\|J_{nd} - J_d\|_1\) using Proposition 17 and then apply a high probability upper bound. Following the same idea of conditioning and peeling off terms as in the proof of Lemma 4, we can condition on the following events, the complement of each of which has a small constant probability:

\[
\begin{align*}
\left\{ \min_n \lambda_{\min}(X^T_{n,S}X_{n,S}) \geq N - 3Cc_2^2\sqrt{N}(\sqrt{D_{\text{eff}}} + 5) \right\} \\
\left\{ \|X_s\|_2 \leq \sqrt{ND_{\text{eff}}} + \sqrt{50Cc_2^2} \right\} \\
\left\{ \max_n \|x_{nS}\|_2 \leq \sqrt{D_{\text{eff}}} + \sqrt{50Cc_2^2} + \sqrt{2Cc_4^2\log N} \right\} \\
\left\{ \max_{d \in S^c} \|X_{-d}\|_2 \leq \sqrt{N} + \sqrt{50Cc_4^2} + \sqrt{2Cc_4^2\log(ND_{\text{eff}})} \right\} \\
\left\{ \max_n \max_{d \in S^c} |x_{n,d}| \leq 50Cc_2^2 + \sqrt{2Cc_4^2\log(ND_{\text{eff}})} \right\} \\
\left\{ \max_n \|x_{nS}\|^2 \leq D_{\text{eff}} + c_2^2D_{\text{eff}}(\log N + 26) \right\}
\end{align*}
\tag{A.45}
\tag{A.46}
\tag{A.47}
\tag{A.48}
\tag{A.49}
\tag{A.50}
\]

The probability of the complement of the first event is \(\leq 3e^{-25}\) by Lemma 4, the second is \(\leq e^{-25}\) by noting that \(\|X_s\|_2 - \sqrt{ND_{\text{eff}}}\) is a \(Cc_2^2\)-sub-Gaussian random variable and applying a standard sub-Gaussian bound, the third is \(\leq e^{-25}\) by applying Proposition 14, the fourth is \(\leq e^{-25}\) by the same reasoning as the third, and the fifth is \(\leq 2e^{-25}\) by the first part of Corollary 4. Finally, the sixth is \(\leq e^{-25}\) by noting that \(\|x_{nS}\|^2\) is a \(c_2^2D_{\text{eff}}\)-sub-Exponential random variable, to which we can apply Corollary 4. All in all, these probabilities sum up to \(9e^{-25}\). Conditioned on all these events, we can upper bound the upper bound on \(\|J_{nd} - J_d\|_1\) given by Proposition 17 to get:

\[
\|J_{nd} - J_d\|_1 \leq \frac{CD_{\text{eff}} \left( \sqrt{50Cc_2^2} + \sqrt{2Cc_4^2\log(ND_{\text{eff}})} \right) \left( \sqrt{D_{\text{eff}}} + \sqrt{50Cc_2^2} + \sqrt{2Cc_4^2\log N} \right)}{N - 3Cc_2^2\sqrt{N}(\sqrt{D_{\text{eff}}} + 5)} + \frac{CD_{\text{eff}} \left( D_{\text{eff}} + D_{\text{eff}}c_2^2(\log N + 26) \right)}{\left( N - 3Cc_2^2\sqrt{N}(\sqrt{D_{\text{eff}}} + 5) \right)^2} \cdot \left( \sqrt{N} + \sqrt{50Cc_4^2} + \sqrt{2Cc_4^2\log(D_{\text{eff}})} \right) \left( \sqrt{ND_{\text{eff}}} + \sqrt{50Cc_4^2} \right)
\]

Call the entire quantity on the right-hand side of this inequality \(M_{\text{lin}}\), and call the union of the above six events the event \(F\). Then by conditioning on \(F\) and the event
\{ \max_{d \in S^c} \| J_d \|_1 < 1 - \alpha \}, \text{ we get:} \\
\Pr \left[ \max_{n \in [N]} \max_{d \in S^c} \| J_{nd} \|_1 \geq 1 - \alpha + M_{\text{lin}} \right] \leq \tag{A.51} \\
\Pr \left[ \max_{n} \max_{d \in S^c} \| J_{nd} - J_n \|_1 \geq M_{\text{lin}} \mid F \right] + \Pr [F^c] + \Pr \left[ \max_{d \in S^c} \| J_d \|_1 \geq 1 - \alpha \right] \tag{A.52} \\
By the definition of \( M_{\text{lin}} \) above, we know that the first probability is zero, by the argument above and a union bound we know \( \Pr [F^c] \leq 9e^{-25} \), and the third is \( \leq e^{-25} \) by Assumption \[2\]. 

A.9.8 Linear regression: bounded gradient

We need to bound the probability

\[
\Pr \left[ \max_{n} \max_{d \in S^c} \| \nabla F(\theta^*) \|_\infty \geq \frac{\lambda (1 - \max_n \max_{d \in S^c} \| J_{nd} \|_1)}{4} \right] \\
\leq \Pr \left[ \max_{n \in [N]} \left( \| \nabla F(\theta^*) \|_\infty + \left\| \frac{1}{N} \nabla f(x_n^T \theta^*, y_n) \right\|_\infty \right) \geq \frac{\lambda (1 - \max_n \max_{d \in S^c} \| J_{nd} \|_1)}{4} \right] \\
\]

Conditioning on the event that \( \| \nabla F(\theta^*) \|_\infty \leq B_G \) for some number \( B_G \) and the event that \( \max_n \max_{d \in S^c} \| J_{nd} \|_1 \leq 1 - \alpha + M_{\text{lin}} \), we get that this probability is less than or equal to:

\[
\leq \Pr \left[ \max_{n=1, \ldots, N} \left\| \frac{1}{N} \nabla f(x_n^T \theta^*, y_n) \right\|_\infty \geq \frac{\lambda (\alpha - M_{\text{lin}})}{4} - B_G \right] \\
+ \Pr [\| \nabla F(\theta^*) \|_\infty \geq B_G] + \Pr \left[ \max_{n \in [N]} \max_{d \in S^c} \| J_{nd} \|_1 \geq 1 - \alpha + M_{\text{lin}} \right] \tag{A.53} \\
\]

The following proposition gives a reasonable value for \( B_G \):

**Proposition 18.** In the above setup for linear regression,

\[
\Pr \left[ \| \nabla F(\theta^*) \|_\infty \geq \left( \frac{c^2 \epsilon^2 \log D}{NC} + \frac{25 c^2 \epsilon^2}{NC} \right)^{1/2} \right] \leq e^{-25} \tag{A.54} \\
\]

**Proof.** The \( d \)th coordinate of the gradient is \( (\nabla F(\theta^*))_d = 1/N \sum_n \varepsilon_n x_{nd} \). First, we have that \( 1/N \sum_n \varepsilon_n x_{nd} \) is a \( c_x c_{\epsilon} \)-sub-Exponential random variable. By Bernstein’s inequality (see Theorem 2.8.1 from [Vershynin 2018]), we have:

\[
\Pr \left[ \left| \frac{1}{N} \sum_{n=1}^N \varepsilon_n x_{nd} \right| \geq \left( \frac{c^2 \epsilon^2 \log D}{NC} + \frac{25 c^2 \epsilon^2}{NC} \right)^{1/2} \right] \leq e^{-25 - \log D} \\
\]

If we union bound over the \( D \) dimensions of \( \nabla F(\theta^*) \), we get that the probability in
the proposition’s statement is \( \leq De^{-25 - \log D} = e^{-25} \), as claimed.

Now we can prove the lemma we need, which bounds the probability that any
\( \| \nabla F^n(\theta^*) \|_{\infty} \) is large:

**Lemma 6.** For the above setup for linear regression and the \( \lambda \) given in Theorem 2, we have:

\[
\Pr \left[ \max_{n=1,...,N} \| \nabla F^n(\theta^*) \|_{\infty} \geq \frac{\lambda(1 - \max_n \max_{d \in S_c} \| J_{nd} \|_1)}{4} \right] \leq 13e^{-25} \quad (A.55)
\]

**Proof.** We can first apply the bound worked out in Eq. (A.53). Picking \( B_G \) to be the
value given in Proposition 18, the second probability is \( \leq e^{-25} \) by Proposition 18 and
the third is \( \leq 10e^{-25} \) by Lemma 5. To analyze the first probability, note that we can
write the event as:

\[
\Pr \left[ \frac{1}{N} \max_n \max_{d} |\varepsilon_{nxd}| \geq \frac{\lambda(\alpha - M_{\text{lin}})}{4} - B_G \right].
\]

Looking at the form of \( \lambda \) given in Theorem 2 we get that this is equal to:

\[
= \Pr \left[ \frac{1}{N} \max_n \max_{d} |\varepsilon_{nxd}| \geq 4c_x c_\varepsilon (\log(ND) + 26) \right].
\]

The event we’re considering is just the absolute value of the max of \( ND \) sub-Exponential
variables with parameter \( c_x c_\varepsilon \). Plugging into Corollary 4 gives that this probability
is \( \leq 2e^{-25} \).

**A.9.9 Linear regression: \( \lambda \) small enough**

To check the bound in Assumption 22 we need to know the LSSC constant \( K \) for
linear regression:

**Proposition 19 ([Li et al., 2015]).** For the linear regression setup in Theorem 6, the
loss \( F(\theta) \) satisfies the \( (\theta^*, N_{\theta^*}) \) LSSC with constant \( K = 0 \) for any \( \theta^* \), \( N_{\theta^*} \), and any
data \( X,Y \).

**Proof.** This follows from the fact that \( F(\theta) = \frac{1}{2} \| X\theta - Y \|^2 \) has zero third derivatives,
implying that \( D^3F(\theta)|_{u,u,e_j} = 0 \) for any \( \theta, u \in \mathbb{R}^D \) and coordinate vector \( e_j \in \mathbb{R}^D \).

As linear regression has a LSSC constant \( K \) that is deterministically equal to zero,
the only constraint implied by the bound in Assumption 22 is that \( \lambda < \infty \), which is
always satisfied by the value of \( \lambda \) given in Theorem 6.
A.9.10 Proof of Theorem 3 (Logistic Regression)

Recall Assumptions 1 and 6: we assume a logistic regression model such that the responses \( y_n \in \{-1, 1\} \) with \( \Pr[y_n = 1] = 1/(1 + e^{x_n^T\theta^*}) \). The derivatives are slightly more complicated here than in the case of linear regression. In particular, defining:

\[
 D_n^{(1)} := \frac{-y_n}{1 + e^{y_n x_n^T\theta^*}}, \quad D_n^{(2)} := \frac{e^{x_n^T\theta^*}}{(1 + e^{x_n^T\theta^*})^2},
\]

the derivatives of \( F \) are:

\[
 \nabla_{\theta} F(\theta^*) = \frac{1}{N} \sum_{n=1}^{N} D_n^{(1)} x_n, \quad \nabla_{\theta}^2 F(\theta^*) = \frac{1}{N} \sum_{n=1}^{N} D_n^{(2)} x_n x_n^T.
\]

For comparison, things were easier for linear regression because \( D_n^{(2)} = 1 \) and \( D_n^{(1)} = \varepsilon_n \) for some sub-Gaussian noise \( \varepsilon_n \). Still, we will be able to extend basically all our proof techniques for linear regression by using the fact that \(|D_n^{(2)}|\) and \(|D_n^{(1)}|\) are both \( \leq 1 \), allowing us to drop them in many of our upper bounds. This will allow us to prove a very similar result to Theorem 2. Again, we will let \( C \) denote an absolute constant independent of any aspect of the problem \((N, D, D_{\text{eff}}, c_x, c)\) that will change from line to line (e.g. we may write \( 5C^2 = C \)).

**Theorem 7.** Take Assumptions 1 to 3, 6, 7 and 23. Suppose the regularization parameter is set as:

\[
 \lambda \geq \frac{C}{\alpha - M_{\text{logr}}} \left( \sqrt{c_x^2 25 + \log D} + \sqrt{2c_x^2 \log(ND)} + \sqrt{50c_x^2} \right),
\]

where \( C \) is a constant in \( N, D, c_x, \) and \( M_{\text{logr}} \) is defined similarly to \( M_{\text{lin}} \) from Theorem 6, but with different denominators:

\[
 M_{\text{logr}} = \frac{CD_{\text{eff}} \left( \sqrt{50c_x^2} + \sqrt{2c_x^2 \log(N(D - D_{\text{eff}}))} \right) \left( \sqrt{D_{\text{eff}}} + \sqrt{50c_x^4} + \sqrt{2c_x^4 \log N} \right)}{L_{\text{min}} - c_x^2 \sqrt{N} (\sqrt{D_{\text{eff}}} + 5)} + \frac{CD_{\text{eff}} (D_{\text{eff}} + D_{\text{eff}} c_x^2 (\log N + 26)) \left( \sqrt{N} + \sqrt{50c_x^4} + \sqrt{2c_x^4 \log(D - D_{\text{eff}})} \right)}{(L_{\text{min}} - c_x^2 \sqrt{N} (\sqrt{D_{\text{eff}}} + 5))^2} * \left( \sqrt{N D_{\text{eff}}} + \sqrt{50c_x^2} \right)
\]

Then for \( N \) sufficiently large, Condition 4 holds with probability at least \( 1 - 43e^{-25} \), where the probability is over the random data \( \{(x_n, y_n)\}_{n=1}^{N} \).

**Proof.** The proof is exactly the same as that of Theorem 6—we bound the probability that any of Assumptions 18 to 22 are violated by a union bound—except that we...
use Lemmas 7 to 9 and Proposition 21 below to bound each term. Note that we have $M_{\text{logr}} = o(1)$ by Assumptions 3 and 7.

A.9.11 Logistic regression: lambda min

**Lemma 7.** Take Assumption 7. Further suppose that $D_{\text{eff}}$ grows as $o(N/\log(N))$. Then for $N$ sufficiently large:

\[
\Pr \left[ \min_{n=1,\ldots,N} \lambda_{\min} (\nabla^2 F^{\lambda_n}(\theta)^{SS}) \leq L_{\min} - Cc^2_x \sqrt{N} (\sqrt{D_{\text{eff}}} + 5) \right] \leq 3e^{-25}, \tag{A.60}
\]

where $L_{\min}$ is the constant from Assumption 7.

**Proof.** We have by Proposition 16 and the fact that $|D_n^{(2)}| \leq 1$:

\[
\lambda_{\min} (\nabla^2 F^{\lambda_n}(\theta)^{SS}) \geq \lambda_{\min} (\nabla^2 F(\theta)^{SS}) - \|x_nS\|_2^2 |D_n^{(2)}| \\
\geq \lambda_{\min} (\nabla^2 F(\theta)^{SS}) - \|x_nS\|_2^2.
\]

The rest of the proof is now exactly the same as that of Lemma 4.

A.9.12 Logistic regression: incoherence

We can get exactly the same bound as in Lemma 5. To do so, we first note that Proposition 17 is only written to deal with Hessians of the form $X^T X$; however, if we rewrite our data as $\bar{x}_n := \sqrt{D_n^{(2)}} x_n$, the Hessian for logistic regression is equal to $\bar{X}^T \bar{X}$. We can further upper bound the upper bound in Proposition 17 by noting that $|D_n^{(2)}| \leq 1 \Rightarrow \|\bar{x}_n\|_2 \leq \|x_n\|_2$. Applying this reasoning, we get an identical lemma to Lemma 5.

**Lemma 8.** Take Assumptions 1, 2, 6 and 7. Then for the scalar $M_{\text{logr}}$ defined in Theorem 3, we have:

\[
\Pr \left[ \max_{n=1,\ldots,N} \max_{d \in S_c} \|J_{nd}\|_1 \geq 1 - \alpha + M_{\text{logr}} \right] \leq 10e^{-25}, \tag{A.61}
\]

where $J_{nd}$ is defined in Eq. (A.29).

**Proof.** The proof is very similar to that of Lemma 5. To prove Lemma 5, we wrote $\|J_{nd}\|_1 \leq \|J_d\|_1 + |J_{nd} - J_d|_1$. To bound $\|J_d\|_1$ with high probability, we applied Assumption 2. To bound $|J_{nd} - J_d|_1$, we used the bound from Proposition 17 and then conditioned on a number of high-probability events to give an overall bound. We can condition on all of the same events, except we replace the event in Eq. (A.45) by:

\[
\left\{ \min_{n=1,\ldots,N} \lambda_{\min} (\nabla^2 F^{\lambda_n}_{SS}) \geq L_{\min} - Cc_x^2 \sqrt{N} (\sqrt{D_{\text{eff}}} + 5) \right\}. \tag{A.62}
\]
By Lemma 7, the complement of this event has probability at most $3e^{-25}$. We condition on the rest of the events in the proof of Lemma 5 and finish the proof along the same lines.

### A.9.13 Logistic regression: bounded gradient

Again, we are interested in bounding:

$$
\Pr \left[ \max_{n=1,\ldots,N} \| \nabla F(\theta^*) \|_\infty \geq \lambda (1 - \max_n \max_{d \in S^c} \| J_{nd} \|_1) \right] \leq 2e^{-25}.
$$

The same reasoning that led to Eq. (A.53) gives us the same bound:

$$
\leq \Pr \left[ \max_{n=1,\ldots,N} \left( \frac{1}{N} \sum_{n=1}^N D_n^{(1)} x_{nd} \right) \geq \lambda \frac{(\alpha - M_{\log r})}{4} - B_G \right] + \Pr [\| \nabla F(\theta^*) \|_\infty \geq B_G] + \Pr \left[ \max_n \max_{d \in S^c} \| J_{nd} \|_1 \geq 1 - \alpha + M_{\log r} \right] \quad (A.63)
$$

Just as in the case of linear regression, we can first pick a reasonable value for $B_G$:

**Proposition 20.** For the logistic regression setup above, we have:

$$
\Pr \left[ \| \nabla F(\theta^*) \|_\infty \geq c_x \sqrt{\frac{25 + \log D}{CN}} \right] \leq 2e^{-25}.
$$

**Proof.** The $d$th coordinate of the gradient is $(\nabla F(\theta^*))_d = \frac{-y_n}{1 + e^{y_n x_n^T \theta^*}}$. Noting that this satisfies $|D_n^{(1)}| \leq 1$:

$$
\Pr \left[ \left| \frac{1}{N} \sum_{n=1}^N D_n^{(1)} x_{nd} \right| \geq c_x \sqrt{\frac{25 + \log D}{CN}} \right] \leq \Pr \left[ \sum_{n=1}^N |x_{nd}| \geq c_x \sqrt{\frac{25 + \log D}{C}} \right] \leq 2e^{-25 - \log D},
$$

where the final inequality comes from noting that $|x_{nd}|$ is also $c_x$-sub-Gaussian and using Hoeffding’s inequality (Theorem 2.6.2 from [Vershynin 2018]). Union bounding over all $D$ dimensions of $\nabla F(\theta^*)$ gives the result.

**Lemma 9.** For the above setup for logistic regression and the $\lambda$ given in Theorem 3.
we have:

\[
\Pr \left[ \max_{n \in [N]} \left\| \nabla F_n^\theta (\theta^*) \right\|_\infty \geq \frac{\lambda (1 - \max_n \max_{d \in S_c} \| J_{nd} \|_1)}{4} \right] \leq 14e^{-25} \quad (A.65)
\]

**Proof.** Just as in the proof of Lemma 6, we will apply the upper bound in Eq. (A.63) and then bound each term. The second probability in Eq. (A.63) is \( \leq 10e^{-25} \) by Lemma 8. The second term is \( \leq 2e^{-25} \) by Proposition 20. We now just need to analyze the first term:

\[
\Pr \left[ \max_{n \in [N]} \left\| \frac{1}{N} \nabla f(x_n^T \theta^*, y_n) \right\|_\infty \geq \frac{\lambda (\alpha - M \log r)}{4} - B_G \right].
\]

Plugging in the \( \lambda \) given in Theorem 7, and using \( \| \nabla f(x_n^T \theta^*, y_n) \|_\infty \leq \| x_n \|_\infty \), we can further upper bound this probability:

\[
\leq \Pr \left[ \max_{n \in [N]} \| x_n \|_\infty \geq \left( \sqrt{2C c_x^2 \log (N D)} + \sqrt{50C c_x^2} \right) \right].
\]

By part 1 of Corollary 4, this probability is \( \leq 2e^{-25} \). \( \square \)

**A.9.14 Logistic regression: \( \lambda \) small enough**

In the case of linear regression, the LSSC held with \( K = 0 \), so there was no work to be done in checking the bound in Assumption 22; this is not the case for logistic regression. Li et al. [2015] prove that the LSSC holds here:

**Proposition 21** ([Li et al., 2015]). The logistic regression model given above satisfies the \( (\theta^*, N_{\theta^*}) \) LSSC for any \( \theta^* \) and \( N_{\theta^*} = \mathbb{R}^D \) with a data-dependent constant \( K = 1/4(\max_n \| x_n \|_\infty)(\max_n \| x_n S \|_2^2) \).

**Proof.** This is proved in Section 6.2 of [Li et al., 2015]. \( \square \)

We first show that this random \( K \) is not too large with high probability under our random design:

**Proposition 22.** For \( x_n \in \mathbb{R}^D \) comprised of i.i.d. \( c_x \)-sub-Gaussian random variables, the random variable \( K = 1/4(\max_n \| x_n \|_\infty)(\max_n \| x_n S \|_2^2) \) satisfies:

\[
\Pr \left[ K \geq \frac{1}{4} \left( \sqrt{2C c_x^2 \log (N D)} + \sqrt{50C c_x^2} \right) \left( D_{\text{eff}} + c_x^2 D_{\text{eff}} (\log N + 26) \right) \right] \leq 3e^{-25} \quad (A.66)
\]

**Proof.** First, Corollary 4 implies that \( \max_n \| x_n \|_\infty \geq \sqrt{2C c_x^2 \log (N D)} + \sqrt{50C c_x^2} \) with probability at most \( 2e^{-25} \), so the probability we are interested in is bounded by:

\[
\leq \Pr \left[ \max_n \| x_n S \|_2^2 \geq D_{\text{eff}} + c_x^2 D_{\text{eff}} (\log N + 26) \right] + 2e^{-25}. \quad (A.67)
\]
Noting that $\|x_n s\|^2_2$ is the sum of $D_{\text{eff}} c_x^2$-sub-Exponential random variables, $\|x_n s\|^2_2$ is a $D_{\text{eff}} c_x^2$-sub-Exponential random variable. Corollary 4 then gives us that Eq. (A.67) is bounded above by $3 e^{-25}$.

We can now prove the result we need, which is that $\lambda$ satisfies the upper bound in Assumption 22 with high probability.

**Lemma 10.** Take Assumptions 1 to 3 and 7. Then, for the logistic regression setup in Assumption 6 and $\lambda$ as given in Theorem 3 and large enough $N$, we have:

$$\Pr \left[ \lambda \geq \frac{\min_n \alpha}{4 \left( 1 - \max_n \max_{d \in S^c} \| J_{nd} \|_1 \right)} \right] \leq 16 e^{-25}$$

(A.68)

**Proof.** Using Lemma 7, Lemma 8, and Proposition 22, the desired probability is $\leq 16 e^{-25}$ if the following deterministic inequality holds:

$$\lambda \leq \frac{4 \left( \alpha - M_{\log r} \right)}{4 \left( \alpha - M_{\log r} + 4 \right)^2} \left( L_{\min} - C c_x^2 D_{\text{eff}} N \sqrt{\log(ND)} + \sqrt{50 c_x^2} \right) \left( D_{\text{eff}} + c_x^2 D_{\text{eff}} (\log N + 26) \right)$$

(A.69)

We will lower bound the right hand side and show that $\lambda$ is less than this lower bound. Throughout, $C$ will be a generic constant that changes from line-to-line. First, as noted in the proof of Theorem 3, $M_{\log r} = o(1)$ as $N \to \infty$, so that for large enough $N$, we have $(\alpha - M_{\log r})/(\alpha - M_{\log r} + 4)^2 \geq (\alpha/2)^2/(\alpha/2 + 4)^2$. Next, for large enough $N$, Lemma 7 implies the denominator is greater than $CN$. Also for large enough $N$, the denominator is less than $CD_{\text{eff}} \log N \sqrt{\log(ND)}$. We are left with checking the condition:

$$\lambda \leq C \frac{\alpha/2}{\left( \alpha/2 + 4 \right)^2} \frac{N^2}{D_{\text{eff}} \log N \sqrt{\log(ND)}}$$

(A.70)

Under Assumption 3, we can upper bound the denominator to get a further lower bound on the right hand side:

$$\lambda \leq C \frac{\alpha/2}{\left( \alpha/2 + 4 \right)^2} \frac{\log^{2/5}(N) N^2}{D_{\text{eff}} \log N \sqrt{\log(ND)} + N}$$

(A.71)

Now, the right hand side goes to infinity as $N$ gets large, while the $\lambda$ given in Theorem 3 goes to 0 as $N$ gets large. Thus, for sufficiently large $N$, Eq. (A.69) holds. \[\square\]
Appendix B

Appendix for: Approximate cross-validation with low-rank data in high dimensions (Chapter 3)

B.1 Derivation of $x_n^T \text{NS}_n$ and $x_n^T \text{IJ}_n$

Here, we derive the expressions for $x_n^T \text{NS}_n$ and $x_n^T \text{IJ}_n$ given in Eqs. (3.3) and (3.4).

We recall from previous work (e.g., see Stephenson and Broderick [2020, Appendix C] for a summary) that the LOOCV parameter estimates given by the Newton step and infinitesimal jackknife approximations are given by

\[
\hat{\theta}^\text{NS}_n := \hat{\theta} + \frac{1}{N} \left( \sum_{m: m \neq n} \hat{D}_n^{(2)} x_m x_m^T + \lambda I_D \right)^{-1} \hat{D}_n^{(1)} x_n \tag{B.1}
\]

\[
\hat{\theta}^\text{IJ}_n := \hat{\theta} + \frac{1}{N} \left( \sum_{n=1}^N \hat{D}_n^{(2)} x_n x_n^T \lambda I_D \right)^{-1} \hat{D}_n^{(1)} x_n. \tag{B.2}
\]

Taking the inner product of $\text{IJ}_n$ with $x_n$ immediately gives Eq. (3.4). To derive Eq. (3.3), define $H := \sum_n \hat{D}_n^{(2)} x_n x_n^T + \lambda I_D$ and note that we can rewrite $\text{NS}_n$ using the Sherman-Morrison formula:

\[
\text{NS}_n = \hat{\theta} + \frac{1}{N} \left[ \hat{D}_n^{(1)} H^{-1} x_n + \hat{D}_n^{(1)} \hat{D}_n^{(2)} \frac{H^{-1} x_n x_n^T H^{-1}}{1 - \hat{D}_n^{(2)} Q_n} x_n \right].
\]

Taking the inner product with $x_n$ and reorganizing gives:

\[
x_n^T \text{NS}_n = x_n^T \hat{\theta} + \hat{D}_n^{(1)} \left[ \frac{Q_n - \hat{D}_n^{(2)} Q_n^2}{1 - \hat{D}_n^{(2)} Q_n} + \frac{\hat{D}_n^{(2)} Q_n^2}{1 - \hat{D}_n^{(2)} Q_n} \right] = x_n^T \hat{\theta} + \frac{\hat{D}_n^{(1)} Q_n}{N} \frac{1 - \hat{D}_n^{(2)} Q_n}{1 - \hat{D}_n^{(2)} Q_n}. \]

124
Figure B-1: Experiment from Appendix B.2. Across four different dataset sizes, using the Neumann series approximation (orange) does not show any noticeable improvement on the time scale of running our approximation (green) for all possible values of $K$.

B.2 Comparison to existing Hessian inverse approximation

We note that two previous works have used inverse Hessian approximations for applications similar to ACV. Koh and Liang [2017] use influence functions to estimate behavior of black box models, and Lorraine et al. [2020] use the implicit function theorem to optimize model hyperparameters. In both papers, the authors need to multiply an inverse Hessian by a gradient. To deal with the high dimensional expense associated with this matrix inverse, both sets of authors use the method of Agarwal et al. [2017], who propose a stochastic approximation to the Neumann series. The Neumann series writes the inverse of a matrix $H$ with operator norm $\|H\|_{op} < 1$ as:

$$H^{-1} = \sum_{k=0}^{\infty} (I - H)^k.$$

The observation of Agarwal et al. [2017] is that this series can be written recursively, as well as estimated stochastically if one has random variables $A_s$ with $\mathbb{E}[A_s] = H$. In the general case of empirical risk minimization with an objective of $(1/N) \sum_{n=1}^{N} f_n(\theta)$, Agarwal et al. [2017] propose using $A_s = \nabla^2 f_s(\theta)$ for some $s \in [N]$ chosen uniformly...
at random. In the GLM setting we are interested in here, we choose an index \( s \in [N] \) uniformly at random and set \( A_s = \hat{D}_n^{(2)} x_s a_s^T + (\lambda/N)D \). Then, for \( s = 1, \ldots, S \), we follow Agarwal et al. [2017] to recursively define:

\[
H^{-1} \approx \hat{H}_s^{-1} := I_D + (I - A_s) \hat{H}_{s-1}^{-1}.
\]

The final recommendation of Agarwal et al. [2017] is to repeat this process \( M \) times and average the results. We thus have two hyperparameters to choose: \( S \) and \( M \).

To test out the Agarwal et al. [2017] approximation against our approximation in Algorithm 1, we generate Poisson regression datasets of increasing sizes \( N \) and \( D \). We generate approximately low-rank covariates \( x_n \) by drawing \( x_{nd} \sim N(0, 1) \) for \( d = 1, \ldots, 1,000 \) and \( x_{nd} \sim N(0, 0.01) \) for \( d = 1,001, \ldots, D; \) for our dataset with \( D = 40 \), we follow the same procedure but with \( R = 20 \) instead. For each dataset, we compute \( IJ_n \), as well as our approximation \( \tilde{I}J_n \) from Algorithm 1. We run Algorithm 1 for \( K = 1, 100, 200, \ldots, D \) and run the stochastic Neumann series approximation with all combinations of \( M \in \{2, 5\} \) and \( S \in \{1, 5, 10, 15, \ldots, 200\} \). We measure the accuracy of all approximations as percent error to exact CV \((x_n^T \hat{\theta}^n)\). We show in Fig. B-1 that our approximation has far improved error in far less time. Notably, this phenomenon becomes more pronounced as the dimension gets higher; while spending more computation on the Neumann series approximation does noticeably decrease the error for the \( N = 80, D = 40 \) case, we see that as soon as we step into even moderate dimensions (\( D \) in the thousands), spending more computation on the Neumann approximation does not noticeably decrease the error. In fact, in the three lowest-dimensional experiments here, the dimension is so low that exactly computing \( H^{-1} \) via a Cholesky decomposition is the fastest method.

We also notice that in the \( N = 80, D = 40 \) experiment, \( \tilde{I}J_n \) is a better approximation of exact CV than is \( IJ_n \) for intermediate values of \( K \) (i.e. some orange dots sit below the large green dot). We note that we have observed this behavior in a variety of synthetic and real-data experiments. We do not currently have further insight into this phenomenon and leave its investigation for future work.

### B.3 Previous ACV theory

We briefly review pre-existing theoretical results on the accuracy of ACV. Theoretical results for the accuracy of \( IJ_n \) are given by Giordano et al. [2019b], Koh et al. [2019], Wilson et al. [2020]. Giordano et al. [2019b] give a \( O(1/N^2) \) error bound for unregularized problems, which Stephenson and Broderick [2020, Proposition 2] extends to regularized problems; however, in our GLM case here, both results require the covariates and parameter space to be bounded. Koh et al. [2019] give a similar bound, but require the Hessian to be Lipschitz, and their bounds rely on the inverse of the minimum singular value of \( H \), making them unsuited for describing the low rank case of interest here. The bounds of Wilson et al. [2020] are close to our bounds in Lemma 2. The difference to our work is that Wilson et al. [2020] consider generic (i.e. not just GLM) models, but also require a Lipschitz assumption on the Hessian.
We specialize to GLMs, avoid the Lipschitz assumption by noting that it only need hold locally, and provide fully computable bounds.

Various theoretical guarantees also exist for the quality of NS\(_n\) from Eq. (3.3). Rad and Maleki [2020] show that the error \(\|\text{NS}\_n - \hat{\theta}\_n\|_2\) is \(o(1/N)\) as \(N \to \infty\) and give conditions under which the error is a much slower \(O(1/\sqrt{N})\) as both \(N, D \to \infty\) with \(N/D\) converging to a constant. Beirami et al. [2017] show that the error is \(O(1/N^2)\), but require fairly strict assumptions (namely, boundedness of the covariates and parameter space). Koh et al. [2019], Wilson et al. [2020] provide what seem to be the most interpretable bounds, but, as is the case for IJ\(_n\), both require a Lipschitz assumption on the Hessian and the results of Koh et al. [2019] depend on the lowest singular value of the Hessian.

B.4 Proofs from Sections 3.3 and 3.4

B.4.1 Proving accuracy of NS\(_n\) and IJ\(_n\) under exact low-rank data (Lemma 1)

Here, we prove that, when the covariate matrix is exactly rank \(R \ll D\), the accuracy of NS\(_n\) and IJ\(_n\) behaves exactly as in a dimension \(R \ll D\) problem. Let \(X = U\Sigma V\) be the singular value decomposition of \(X\), where \(\Sigma\) is a diagonal matrix with only \(R\) non-zero entries; let \(V_R \in \mathbb{R}^{D \times R}\) be the right singular vectors of \(X\) corresponding to these \(R\) non-zero singular values. We define the restricted, \(R\)-dimensional problem with covariates \(\tilde{x}_n := V_R^T x_n\) as:

\[
\hat{\phi} := \arg \min_{\phi \in \mathbb{R}^R} \frac{1}{N} \sum_{n=1}^{N} f(\tilde{x}_n^T \phi) + \frac{\lambda}{2} \|\phi\|_2^2. \tag{B.3}
\]

Let \(\hat{\phi}_n\) be the solution to the leave-one-out version of this problem and RIJ\(_n\) and RNS\(_n\) the application of Eqs. (3.3) and (3.4) to this problem. We then have the following proposition, which implies the statement of Lemma 1.

**Proposition 23** (Generalization of Lemma 1). The following hold for all datapoints \(n\):

\[
x_n^T \hat{\theta}\_n = \tilde{x}_n^T \hat{\phi}_n
\]
\[
x_n^T \text{IJ}\_n = \tilde{x}_n^T \text{RIJ}\_n
\]
\[
x_n^T \text{NS}\_n = \tilde{x}_n^T \text{RNS}\_n.
\]

In particular, \(|x_n^T \text{NS}\_n - x_n^T \hat{\theta}\_n| = |\tilde{x}_n^T \text{RNS}\_n - \tilde{x}_n^T \hat{\phi}_n|\) and \(|x_n^T \text{IJ}\_n - x_n^T \hat{\theta}\_n| = |\tilde{x}_n^T \text{RIJ}\_n - \tilde{x}_n^T \hat{\phi}_n|\), as claimed in Lemma 1.

**Proof.** First, note that if \(\hat{\phi}\) is an optimum of Eq. (B.3), then \((1/N) \sum_n \hat{D}_n^{(l)} V_R^T x_n + \lambda \hat{\phi} = 0\). As \(V_R V_R^T x_n = x_n\), we have that \(\hat{\theta} = V_R \hat{\phi}\) is optimal for the full, \(D\)-
dimensional problem. This implies that \( \hat{\phi} = V^T_R \hat{\theta} \), and thus \( x_n^T \hat{\theta} = \tilde{x}_n^T \hat{\phi} \). The same reasoning shows that \( x_n^T \hat{\theta} = \tilde{x}_n^T \hat{\phi} \).

Now, notice that the Hessian of the restricted problem, \( H_R \), is given by \( H_R = (1/N) \sum_n V^T_R x_n x_n^T V_R \hat{D}_n^{(2)} + \lambda I \) \( \implies H_R^{-1} = V^T_R H^{-1} V_R \), where the \( \hat{D}_n^{(2)} \) are evaluated at \( \hat{x}_n^T \hat{\phi} = x_n^T \hat{\theta} \). Also, the gradients of the restricted problem are given by \( \nabla \phi f(x_n^T \hat{\theta}, y_n) = \hat{D}_n^{(1)} V^T_R x_n \). Thus the restricted IJ is:

\[
RIJ_n = \hat{\phi} + H_R^{-1} V_R x_n \hat{D}_n^{(1)} = V^T_R \left( \hat{\theta} + H^{-1} x_n \hat{D}_n^{(1)} \right) = V^T_R IJ_n.
\]

Thus, we have \( \tilde{x}_n^T RIJ_n = x_n^T V_R^T IJ_n \). Using \( V_R^T x_n = x_n \), we have that \( \tilde{x}_n^T RIJ_n = x_n^T IJ_n \). The proof that \( \tilde{x}_n^T RNS_n = x_n^T NS_n \) is identical. \( \square \)

### B.4.2 Proving accuracy of \( NS_n \) and \( IJ_n \) under ALR data (Lemma 2)

We will first need a few lemmas relating to how the exact solutions \( \hat{\theta} \) and \( \hat{\theta} \) vary as we leave datapoints out and move from exactly low-rank to ALR. We start by bounding \( \| \hat{\theta} - \hat{\theta} \|_2 \); this result and its proof are from [Wilson et al. 2020, Lemma 16] specialized to our GLM context.

**Lemma 11.** Assume that \( \lambda > 0 \). Then:

\[
\| \hat{\theta} - \hat{\theta} \|_2 \leq \frac{1}{N\lambda} |\hat{D}_n^{(1)}| \| x_n \|_2. \tag{B.4}
\]

**Proof.** Let \( F^{\setminus n} \) be the leave-one-out objective, \( F^{\setminus n}(\theta) = (1/N) \sum_{m: m \neq n} f(x_m^T \theta, y_m) + (\lambda/2) \| \theta \|_2^2 \). As \( F^{\setminus n} \) is strongly convex with parameter \( \lambda \), we have:

\[
\lambda \| \hat{\theta} - \hat{\theta} \|_2^2 \leq \langle \hat{\theta} - \hat{\theta} \| n \nabla F^{\setminus n}(\hat{\theta}) - \nabla F^{\setminus n}(\hat{\theta} \| n) \rangle.
\]

Now, use the fact that \( \nabla F^{\setminus n}(\hat{\theta}) = \nabla F(\hat{\theta}) \) is 0 and then that \( F^{\setminus n} - F = (1/N) f(x_n^T \theta) \) to get:

\[
= \langle \hat{\theta} - \hat{\theta} \| n, \nabla F^{\setminus n}(\hat{\theta}) - \nabla F(\hat{\theta}) \rangle = \langle \hat{\theta} - \hat{\theta} \| n, \nabla f(x_n^T \theta) \rangle \\
\leq \| \hat{\theta} - \hat{\theta} \|_2 |\hat{D}_n^{(1)}| \| x_n \|_2.
\]

\( \square \)

We will need a bit more notation to discuss the ALR and exactly low-rank versions of the same problem. Suppose we have a \( N \times D \) covariate matrix \( X \) that is exactly low-rank (ELR) with rows \( x_{n,ELR} \in \mathbb{R}^D \). Then, suppose we form some approximately low-rank (ARL) covariate matrix by adding \( \varepsilon_n \in \mathbb{R}^D \) to all \( x_n \) such that \( X \varepsilon_n = 0 \) for all \( \varepsilon_n \). Let \( x_{n,ALR} \) be the rows of this ALR matrix. Let \( \hat{\theta}_{ELR} \) be the fit with the ELR
data and $\hat{\theta}_{ALR}$ the fit with the ALR data. Finally, define the scalar derivatives:

$$
\hat{D}^{(1)}_{n,ELR}(\theta) := \frac{df(z, y_n)}{dz} \bigg|_{z = \langle x_n, ELR, \theta \rangle}
$$

$$
\hat{D}^{(1)}_{n,ALR}(\theta) := \frac{df(z, y_n)}{dz} \bigg|_{z = \langle x_n, ALR, \theta \rangle}
$$

We can now give an upper bound on the difference between the ELR and ALR fits $\|\hat{\theta}_{ELR} - \hat{\theta}_{ALR}\|_2$. Our bound will imply that the $\hat{\theta}_{ALR}$ is a continuous function of the $\varepsilon_n$, which in turn are continuous functions of the singular values of the ALR covariate matrix.

**Lemma 12.** Assume $\lambda > 0$. We have:

$$
\|\hat{\theta}_{ELR} - \hat{\theta}_{ALR}\|_2 \leq \frac{1}{N\lambda} \left\| \sum_{n=1}^{N} \hat{D}^{(1)}_{n,ELR}(\hat{\theta}_{ELR})\varepsilon_n \right\|_2
$$

In particular, $\hat{\theta}_{ALR}$ is a continuous function of the $\varepsilon_n$ around $\varepsilon_1, \ldots, \varepsilon_N = 0$.

**Proof.** Denote the ALR objective by $F_{ALR}(\theta) = (1/N) \sum_n f(x_n^{T, ALR} \theta) + \lambda \|\theta\|_2^2$. Then, via a Taylor expansion of its gradient around $\hat{\theta}_{ALR}$:

$$
\nabla_{\theta} F_{ALR}(\hat{\theta}_{ELR}) = \nabla_{\theta} F_{ALR}(\hat{\theta}_{ALR}) + \nabla_{\theta}^2 F_{ALR}(\tilde{\theta})(\hat{\theta}_{ELR} - \hat{\theta}_{ALR}),
$$

where $\tilde{\theta} \in \mathbb{R}^D$ satisfies $\tilde{\theta}_d = (1 - s_d)\theta_{ALR,d} + s_d\theta_{ELR,d}$ for some $s_d \in [0, 1]$ for each $d = 1, \ldots, D$. Via strong convexity and $\nabla_{\theta} F_{ALR}(\hat{\theta}_{ALR}) = 0$, we have:

$$
\|\hat{\theta}_{ELR} - \hat{\theta}_{ALR}\|_2 \leq \frac{1}{\lambda} \left\| \nabla_{\theta} F_{ALR}(\hat{\theta}_{ELR}) \right\|_2.
$$

Now, note that the gradient on the right hand side of this equation is equal to

$$
\nabla_{\theta} F_{ALR}(\hat{\theta}_{ELR}) = \frac{1}{N} \sum_{n=1}^{N} \hat{D}^{(1)}_{n,ALR}(\hat{\theta}_{ELR}) x_{n,ELR} + \frac{1}{N} \sum_{n=1}^{N} \hat{D}^{(1)}_{n,ALR}(\hat{\theta}_{ELR})\varepsilon_n + \lambda \hat{\theta}_{ELR}. \quad (B.5)
$$

By the optimality of $\hat{\theta}_{ELR}$ for the exactly low-rank problem, we must have that $\langle \varepsilon_n, \hat{\theta}_{ELR} \rangle = 0$ for all $n$; in particular, this implies that $\langle x_{n,ELR}, \hat{\theta}_{ELR} \rangle = \langle x_{n,ALR}, \hat{\theta}_{ELR} \rangle$, which in turn implies $\hat{D}^{(1)}_{n,ALR}(\hat{\theta}_{ELR}) = \hat{D}^{(1)}_{n,ELR}(\hat{\theta}_{ELR})$ for all $n$. Also by the optimality of $\hat{\theta}_{ELR}$, we have $(1/N) \sum_n \hat{D}^{(1)}_{n,ELR}(\hat{\theta}_{ELR}) x_n + \lambda \hat{\theta}_{ELR} = 0$. Thus we have that Eq. (B.5) reads:

$$
\nabla_{\theta} F_{ALR}(\hat{\theta}_{ELR}) = \frac{1}{N} \sum_{n=1}^{N} \hat{D}^{(1)}_{n,ELR}(\hat{\theta}_{ELR})\varepsilon_n,
$$

which completes the proof. \qed

We now restate and prove Lemma 2.
Lemma 2. Assume that $\lambda > 0$ and recall the definition of $L_n$ from Eq. (3.8). Then, for all $n$:

\[ |x_n^T \text{NS}_n - x_n^T \hat{\theta}^n| \leq \frac{L_n}{N^2 \lambda^3} |\tilde{D}_n^{(1)}|^2 \| x_n \|_2^3 \] (B.6)

\[ |x_n^T \text{IJ}_n - x_n^T \hat{\theta}^n| \leq \frac{L_n}{N^2 \lambda^3} |\tilde{D}_n^{(1)}|^2 \| x_n \|_2^3 + \frac{1}{N^2 \lambda^2} |\tilde{D}_n^{(1)}| \tilde{D}_n^{(2)} \| x_n \|_2^4. \] (B.7)

Furthermore, these bounds continuously decay as the data move from exactly to approximately low rank in that they are continuous in the singular values of $X$.

Proof. The proof of Eqs. (B.6) and (B.7) strongly resembles the proof of Wilson et al. [2020] Lemma 17] specialized to our current context. We first prove Eq. (B.6). We begin by applying the Cauchy-Schwarz inequality to get:

\[ |x_n^T \text{NS}_n - x_n^T \hat{\theta}^n| \leq \|x_n\|_2 \|\text{NS}_n - \hat{\theta}^n\|_2. \]

The remainder of our proof focuses on bounding $\|\text{NS}_n - \hat{\theta}^n\|_2$. Let $\tilde{F}^{\text{NS}}$ be the second order Taylor expansion of $F^{\text{NS}}$ around $\hat{\theta}$; then $\text{NS}_n$ is the minimizer of $\tilde{F}^{\text{NS}}$. By the strong convexity of $\tilde{F}^{\text{NS}}$:

\[ \lambda \|\hat{\theta}^n - \text{NS}_n\|_2^2 \leq \langle \text{NS}_n - \hat{\theta}^n, \nabla \tilde{F}^{\text{NS}}(\text{NS}_n) - \nabla \tilde{F}^{\text{NS}}(\hat{\theta}^n) \rangle \] (B.8)

\[ = \langle \text{NS}_n - \hat{\theta}^n, \nabla \tilde{F}^{\text{NS}}(\hat{\theta}^n) - \nabla \tilde{F}^{\text{NS}}(\hat{\theta}^n) \rangle \] (B.9)

Now the goal is to bound this quantity as the remainder in a Taylor expansion. To this end, define $r(\theta) := \langle \hat{\theta}^n - \text{NS}_n, \nabla F^{\text{NS}}(\theta) \rangle$. To apply Taylor’s theorem with integral remainder, define $g(t) := r((1 - t)\hat{\theta} + t\hat{\theta}^n)$ for $t \in [0, 1]$. Then, by a zeroth order Taylor expansion:

\[ g(1) = g(0) + g'(0) + \int_0^1 (g'(s) - g'(0)) \, ds. \]

Putting in the values of $g$ and its derivatives:

\[ \langle \hat{\theta}^n - \text{NS}_n, \nabla F^{\text{NS}}(\hat{\theta}) \rangle = \langle \hat{\theta}^n - \text{NS}_n, \nabla F^{\text{NS}}(\hat{\theta}) \rangle + \langle \hat{\theta}^n - \text{NS}_n, \nabla^2 F^{\text{NS}}(\hat{\theta})(\hat{\theta}^n - \hat{\theta}) \rangle \]

\[ + \int_0^1 \langle \hat{\theta}^n - \text{NS}_n, \left( \nabla^2 F^{\text{NS}}((1 - s)\hat{\theta} + s\hat{\theta}^n) - \nabla^2 F^{\text{NS}}(\hat{\theta}) \right)(\hat{\theta}^n - \hat{\theta}) \rangle ds \]

Now, subtracting the first two terms on the right hand side from the left, we get can identify the left with Eq. (B.9). Thus, Eq. (B.9) is equal to:

\[ = \int_0^1 \langle \hat{\theta}^n - \text{NS}_n, \left( \nabla^2 F^{\text{NS}}((1 - s)\hat{\theta} + s\hat{\theta}^n) - \nabla^2 F^{\text{NS}}(\hat{\theta}) \right)(\hat{\theta}^n - \hat{\theta}) \rangle ds. \]

We can upper bound this by taking an absolute value, then applying the triangle
inequality and Cauchy-Schwarz to get

$$
\leq \left\| \hat{\theta}^{\|n} - NS^{\|n} \right\|_2 \left\| \hat{\theta}^{\|n} - \hat{\theta} \right\| \int_0^1 \left\| \left( \nabla^2 F^{\|n}((1-s)\hat{\theta} + s\hat{\theta}^{\|n}) - \nabla^2 F^{\|n}(\hat{\theta}) \right) \right\|_{op} ds.
$$

(B.10)

Using the fact that, on the line segment $(1-s)\hat{\theta} + s\hat{\theta}^{\|n}$, the $\hat{D}_n^{(2)}$ are lipschitz with constant $C_n$:

$$
C_n := \max_{s=\eta[0,1]} \left\| \hat{D}_n^{(3)} \left( (1-s)\hat{\theta} + s\hat{\theta}^{\|n} \right) \right\|
$$

we can upper bound the integrand by:

$$
\left\| \left( \nabla^2 F^{\|n}((1-s)\hat{\theta} + s\hat{\theta}^{\|n}) - \nabla^2 F^{\|n}(\hat{\theta}) \right) \right\|_{op}
= \frac{1}{N} \left\| \sum_{m \neq n} \left( \hat{D}_n^{(2)}((1-s)\hat{\theta} + s\hat{\theta}^{\|n}) - \hat{D}_n^{(2)}(\hat{\theta}) \right) x_m x_m^T \right\|_{op}
\leq \frac{C_n \left\| \hat{\theta}^{\|n} - \hat{\theta} \right\|_2}{N} \sum_{m \neq n} \left\| x_m \right\|_2^2.
$$

Putting this into Eq. (B.10) and using Lemma 11 gives the result Eq. (B.6) with $L_n = C_n/N \sum_{m \neq n} \left\| x_m \right\|_2^2.$

Now Eq. (B.7) follows from the triangle inequality $\left\| IJ \setminus - \hat{\theta}^{\|n} \right\|_2 \leq \left\| NS \setminus - \hat{\theta}^{\|n} \right\|_2 + \left\| IJ \setminus - NS \right\|_2.$ The bound on $\left\| IJ \setminus - NS \right\|_2$ follows from Wilson et al. [2020, Lemma 20].

Finally, the continuity of the bounds in Eqs. (B.6) and (B.7) follows from Lemma 12. In particular, the $\hat{D}_n^{(1)}$, $\hat{D}_n^{(2)}$, and $\hat{D}_n^{(3)}$ in both bounds are evaluated at $\hat{\theta}_{ALR}$, which is shown to be a continuous function of the $\varepsilon_n$ in Lemma 12. The $\varepsilon_n$ are, in turn, continuous functions of the lower singular values of the covariate matrix.

B.4.3 Proof of Theorem 4

Proof. We first note that the runtime claim is immediate, as Algorithm 2 runs in $O(NDK + K^3)$ time. That the bounds are computable in $O(DK)$ time for each $n$ follows as all derivatives $\hat{D}_n^{(1)}$ and $\hat{D}_n^{(2)}$ need only the inner product of $x_n$ and $\hat{\theta}$, which takes $O(D)$ time. Each norm $\left\| x_n \right\|_2$ is computable in $O(D)$. For models for which the optimization problem in Proposition 2 can be quickly solved – such as Poisson or logistic regression – we need only to compute a bound on $\left\| \hat{\theta}^{\|n} - \hat{\theta} \right\|_2$, which we can do in $O(D)$ via Lemma 11. The only remaining quantity to compute is the $\eta_n$, which, by Proposition 3 is computed via a projection onto the orthogonal complement of a $K$-dimensional subspace. We can compute this projection in $O(DK)$. Thus our overall runtime is $O(DK)$ per datapoint.

To prove Eq. (3.7), we use the triangle inequality $\left\| x_n^T IJ \setminus - x_n^T \hat{\theta}^{\|n} \right\| \leq \left\| x_n^T IJ \setminus - x_n^T \hat{\theta}^{\|n} \right\| + \left\| x_n^T IJ \setminus - x_n^T IJ \setminus \right\|$. We upper bound the first term by using Lemma 2. For
the latter, we note that $|x_n^T I J_{\setminus n} - x_n^T \tilde{I} J_{\setminus n}| = |\hat{D}_n^{(1)}||Q_n - \tilde{Q}_n|$, which we can bound via the $\eta_n$ of Proposition 3. The proof for NS\_n is similar.

\section*{B.4.4 Proof of Corollary 1}

\textit{Proof.} Notice that $\Omega$ from Algorithm 2 captures a rank-$K$ subspace of the column span of $X$. The error bound $\eta_n$ is the norm of $x_n$ projected outside of this subspace divided by $\lambda$. Now, assume that we have $K \geq R$. Then, as the singular values $\sigma_d$ for $d = R + 1, \ldots, D$ go to zero, the norm of any $x_n$ outside this subspace must also go to zero. Thus $\eta_n$ goes to zero. As $E_n$ is a continuous function of $\eta_n$, we also have $E_n \to 0$.

\section*{B.5 Proofs and discussion from Section 3.5}

\subsection*{B.5.1 Proofs}

For convenience, we first restate each claimed result from the main text before giving its proof.

\textbf{Proposition 3.} Let $\lambda > 0$ and suppose there is some subspace $B$ on which $H$ and $\tilde{H}$ exactly agree. Then $H^{-1}$ and $\tilde{H}^{-1}$ agree exactly on the subspace $A := HB$, and for all $n = 1, \ldots, N$:

\[ |x_n^T \tilde{H}^{-1} x_n - Q_n| \leq \frac{\|P_{A}^\perp x_n\|_2^2}{\lambda}, \]  

(B.11)

where $P_{A}^\perp$ denotes projection onto the orthogonal complement of $A$.

\textit{Proof.} First, if $H$ and $\tilde{H}$ agree on $B$, then for $A = HB = \tilde{H}B$, we have $H^{-1}A = B = \tilde{H}^{-1}A$, as claimed. Then:

\[ |Q_n - x_n^T \tilde{H}^{-1} x_n| = |x_n^T H^{-1} x_n - x_n^T \tilde{H}^{-1} x_n| \]

\[ \leq \|(P_{A}^\perp x_n)(H^{-1} - \tilde{H}^{-1})(P_{A}^\perp x_n)\| \]

\[ \leq \|P_{A}^\perp x_n\|_2^2 \left\|H^{-1} - \tilde{H}^{-1}\right\|_{op,A^\perp}, \]

where $\|\cdot\|_{op,A^\perp}$ is the operator norm of a matrix restricted to the subspace $A^\perp$. On this subspace, the action of $\tilde{H}^{-1}$ is $1/\lambda$ times the identity, whereas all eigenvalues of $H^{-1}$ are all between 0 and $1/\lambda$. Thus:

\[ \left\|\tilde{H}^{-1} - H^{-1}\right\|_{op,A^\perp} = \max_{v \in A^\perp, \|v\|_2 = 1} \left\|v^T \tilde{H}^{-1} v - v^T H^{-1} v\right\| \]

\[ = \max_{v \in A^\perp, \|v\|_2 = 1} \left[\frac{1}{\lambda} - v^T H^{-1} v\right] \leq \frac{1}{\lambda}. \]

\hfill \Box
We next restate and proof Proposition 5.

**Proposition 5** The $Q_n = x_n^T H^{-1} x_n$ satisfy $0 \leq Q_n \leq \|x_n\|^2_2 / (\lambda + \hat{D}_n^{(2)} \|x_n\|^2_2)$. Furthermore, letting $\hat{Q}_n := \min\{x_n^T H^{-1} x_n, \|x_n\|^2_2 / (\lambda + \hat{D}_n^{(2)} \|x_n\|^2_2)\}$, we have the error bound

$$|\tilde{Q}_n - Q_n| \leq \min\left\{\frac{\|P_{\mathcal{A}} x_n\|^2_2}{\lambda}, \frac{\|x_n\|^2_2}{\lambda + \hat{D}_n^{(2)} \|x_n\|^2_2}\right\}.$$  \hspace{1cm} (B.12)

**Proof.** Let $b_n := \sqrt{\hat{D}_n^{(2)}} x_n$. Let $\{v_d\}_{d=1}^D$ be the eigenvectors of $H$ with eigenvalues $\{\gamma_d + \lambda\}_{d=1}^D$, with $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_D$. The quantity $b_n^T H^{-1} b_n$ is maximized if $b_n$ is parallel to $v_D$; in this case, $b_n^T H^{-1} b_n = \|b_n\|^2_2 / (\gamma_D + \lambda)$. Now, recall that the $\gamma_d$ are the eigenvalues of $\sum_n b_n b_n^T$, meaning $\sum_n \langle b_n, v_d \rangle^2 = \gamma_d$. So, if $b_n$ is parallel to $v_D$, it must be that $\gamma_D \geq \|b_n\|^2_2$. Thus, $b_n^T H^{-1} b_n \leq \|b_n\|^2_2 / (\|b_n\|^2_2 + \lambda)$. Dividing by $\hat{D}_n^{(2)}$ gives that $Q_n = x_n^T H^{-1} x_n$ satisfies:

$$0 \leq Q_n \leq \frac{\|x_n\|^2_2}{\lambda + \hat{D}_n^{(2)} \|x_n\|^2_2}.$$  

If we estimate $Q_n$ by the minimum of this upper bound and $x_n^T \tilde{H}^{-1} x_n$, the error bound from Proposition 3 implies the error bound claimed here.

\[ \square \]

### B.5.2 Relation of Algorithm 2 to techniques from randomized linear algebra

As noted, our Algorithm 2 bears a resemblance to techniques from the randomized numerical linear algebra literature. Indeed, our inspiration for Algorithm 2 was the work of Tropp et al. [2017]. Tropp et al. [2017] propose a method to find a randomized top-$K$ eigendecomposition of a positive-semidefinite matrix $B$. Their method follows the basic steps of (1) produce a random orthonormal matrix $\Omega \in \mathbb{R}^{D \times (S+K)}$, where $S \geq 0$ is an oversampling parameter to ensure the stability of the estimated eigendecomposition, (2) compute the Nyström approximation of $B_{\text{nystr}} \approx B$ using $\Omega$, and (3) compute the eigendecomposition of $B_{\text{nystr}}$ and throw away the lowest $S$ eigenvalues.

Our Algorithm 2 can be seen as using this method of Tropp et al. [2017] to obtain a rank-$K$ decomposition of the matrix $B = (1/N) \sum_n \hat{D}_n^{(2)} x_n x_n^T$ with specific choices of $S$ and $\Omega$. First, we notice that $S = 0$ (i.e., no oversampling) is optimal in our application – the error bound of Proposition 3 decreases as the size of the subspace $\mathcal{A}$ increases. As $S > 0$ only decreases the size of this subspace, we see that our specific application is only hurt by oversampling. Next, while Tropp et al. [2017] recommend completely random matrices $\Omega$ for generic applications (e.g., the entries of $\Omega$ are i.i.d. $\mathcal{N}(0, 1)$), we note that the results of Proposition 4 suggest that we can improve upon this choice. With the optimal choice of $S = 0$, we note that $\mathcal{A} = H \Omega$. In this case, Proposition 4 implies it is optimal to set $\Omega = H^{-1} V_K$, where $V_K$ are the top-$K$ right singular vectors of $X$. Algorithm 2 provides an approximation to this optimal choice.
Figure B-2: Quality of approximation of $IJ_n$ on a synthetic Poisson regression problem using the methods from Section 3.5. (Left): We show three options for the choice of the matrix $\Omega$. Blue shows the choice of $\Omega$ having orthonormal columns selected uniformly at random, orange the optimal choice of $\Omega$ from Proposition 4, and green our approximation to this optimal choice. Percent error $|IJ_n - \tilde{IJ}_n|/|IJ_n|$ is reported to give a sense of scale. (Right): Importance of Proposition 5 for approximating $Q_n$. We show two approximations along with our upper bounds on their error: (1) $Q_n \approx x^T_n H^{-1} x_n$ and (2) our recommended $Q_n \approx \tilde{Q}_n$ from Proposition 5. We report absolute error $|IJ_n - \tilde{IJ}_n|$ so that both actual and estimated error can be plotted.

We illustrate the various possible choices of $\Omega$, including i.i.d. $\mathcal{N}(0, 1)$, in Fig. B-2. We generate a synthetic Poisson regression problem with covariates $x_{nd}$ i.i.d. $\mathcal{N}(0, 1)$ and $y_n \sim \text{Poisson}(e^{x^T_n \theta^*})$, where $\theta^* \in \mathbb{R}^D$ is a true parameter with i.i.d. $\mathcal{N}(0, 1)$ entries. We generate a dataset of size $N = 200$. The covariates have dimension $D = 150$ but are of rank 50. We compute $IJ$ for various settings of $K$ and $\Omega$, as shown in Fig. B-2. As suggested by the above discussion, we use no oversampling (i.e., $S = 0$). On the left, we see that using a diagonal approximation to $H^{-1}$ and a single subspace iteration gives a good approximation to the optimal setting of $\Omega$. On the right, we see the improvement made by use of the upper bound on $x^T_n H^{-1} x_n$ from Proposition 5.

B.5.3 Implementation of Algorithm 2

As noted by Tropp et al. [2017], finding the decomposition of $B$ in Algorithm 2 as-written can result in numerical issues. Instead, Tropp et al. [2017] present a numerically stable version which we use in our experiments. For completeness, we state this implementation here, which relies on computing the Nyström approximation of the shifted matrix $B_\nu = B + \nu I_D$, for some small $\nu > 0$:

1. Construct the shifted matrix sketch $G_\nu := B\Omega + \nu \Omega$.

2. Form $C = \Omega^T G_\nu$.

3. Compute the Cholesky decomposition $C = \Gamma \Gamma^T$. 

134
Figure B-3: Experiments from Appendix B.6 (Left): Error bounds can be vacuously large; for trial number 16, our bound exceeds exact CV by two orders of magnitude. (Right): By computing our bound for all \( n \) and re-running exact CV for the two largest values, we obtain estimates that are much closer to exact CV.

4. Compute \( E = G \nu \Gamma^{-1} \).

5. Compute the SVD \( E = U \Sigma V^T \).

6. Return \( U \) and \( \Sigma^2 - \nu I \) as the approximate eigenvectors and eigenvalues of \( B \).

B.6 Error bound experiments

Here, we provide more details on our investigation of the error bounds of Theorem 4 from Fig. 3-4. In Section 3.6, we showed that, over five randomly generated synthetic datasets, our error bound on \( x_n^T \Pi_n \) implies upper bounds on out-of-sample error that are reasonably tight. However, we noted that these bounds can occasionally be vacuously loose. On the left of Appendix B.6, we show this is the case by repeating the experiment in Fig. 3-4 for an additional fifteen trials. While most trials have similar behavior to the first five, trial 16 finds an upper bound of the out-of-sample error that is too loose by two orders of magnitude. However, we note that this behavior is mostly due to two offending points \( n \). Indeed, on the right of Appendix B.6, we show the same results having replaced the two largest bound values with those from exact CV.

B.7 Real data experiments

Here we provide more details about the three real datasets used in Section 3.6.

1. The p53 dataset is from [Danziger et al. 2009, 2007, 2006]. The full dataset contains \( D = 5,408 \) features describing attributes of mutated p53 proteins. The task is to classify each protein as either “transcriptionally competent” or
Figure B-4: Experiment from Appendix B.8. We report average error in the estimate of $Q_n$, $(1/N) \sum_n |Q_n - \tilde{Q}_n|/|Q_n|$ for both the p53 and blog datasets. We note that the errors when using $K = 300, 350, 400$ are visually indistinguishable from one another.

inactive. To keep the dimension high relative to the number of observations $N$, we subsampled $N = 8,000$ datapoints uniformly at random for our experiments here. We fix $K = 500$ to compute $\tilde{Q}_n$ for both $x_n^T \tilde{I}_n$ and $x_n^T \tilde{S}_n$.

2. The rcv1 dataset is from Lewis et al. [2004]. The full dataset is of size $N = 20,242$ and $D = 47,236$. Each datapoint corresponds to a Reuters news article given one of four labels according to its subject: “Corporate/Industrial,” “Economics,” “Government/Social,” and “Markets.” We use a pre-processed binarized version from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html, which combines the first two categories into a "positive" label and the latter two into a "negative" label. We found running CV on the full dataset to be too computationally intensive, and instead formed a smaller dataset of size $N = D = 20,000$. The data matrix is highly sparse, so we chose our 20,000 dimensions by selecting the most common (i.e., least sparse) features. We then chose $N = 20,000$ datapoints by subsampling uniformly at random. We fix $K = 1,000$ in this experiment.

3. The blog dataset is from Buza [2014]. The base dataset contains $D = 280$ features about $N = 52,397$ blogs. Each feature represents a statistic about web traffic to the given blog over a 72 hour period. The task is to predict the number of unique visitors to the blog in the subsequent 24 hour period. We first generate a larger dataset by considering all possible pairwise features $x_{nd_1} x_{nd_2}$ for $d_1, d_2 \in \{1, \ldots, D\}$. The resulting problem has too high $N$ and $D$ to run exact CV on in a reasonable amount of time, so we again subsample to $N = 20,000$ and $D = 20,280$. We again choose the 20,000 least sparse pairwise features and then add in the original 280 features. Finally, we choose our 20,000 datapoints uniformly at random.
B.8 Sensitivity of results to $\lambda$

In our main real-data experiments in Section 3.6, we chose a moderate value of $\lambda = 5.0$ to speed up the convergence of exact CV. Here, we investigate how sensitive our results are to the choice of this parameter. We randomly select $N = 600, D = 400$ subsets of the p53 and blog datasets. We exactly compute $Q_n$, as well as compute our approximation $\hat{Q}_n$ from Algorithm 2 for $K \in \{100, 150, 200, 250, 300, 350, 400\}$. In Fig. B-4, we see that the choice of $\lambda$ has only a mild effect on the results.
Appendix C

Appendix for: Approximate cross-validation for structured models (Chapter 4)

C.1 Related work: Approximate CV methods

A growing body of recent work has focused on various methods for approximate CV (ACV). As outlined in the introduction, these methods generally take one of two forms. The first is based on taking a single Newton step on the leave-out objective starting from the full data fit, \( \hat{\Theta} \). This approximation was first proposed by Obuchi and Kabashima [2016, 2018] for the special cases of linear and logistic regression and first applied to more general models by Beirami et al. [2017]. While this approximation is generally applicable to any CV scheme (e.g. beyond LOOCV) and any model type (e.g. to structured models), it is only efficiently applicable to LOOCV for GLMs. In particular, approximating each \( \hat{\Theta}^{(o)} \) requires the computation and inversion of the \( D \times D \) Hessian matrix. In the case of LOOCV GLMs, this computation can be performed quickly using standard rank-one matrix updates; however, in more general settings, no such convenience applies.

Various works detail the theoretical properties of the NS approximation. Beirami et al. [2017], Rad and Maleki [2020] provide some of the first bounds on the quality of the NS approximation, but under fairly strict assumptions. Beirami et al. [2017] assume boundedness of both the parameter and data spaces, while Rad and Maleki [2020] require somewhat hard-to-check assumptions about the regularity of each leave-out objective (although they successfully verify their assumptions on a handful of problems). Koh et al. [2019] prove bounds on the accuracy of the NS approximation with fairly standard assumptions (e.g. Lipschitz continuity of higher-order derivatives of the objective function), but restricted to models using \( \ell_2 \) regularization. Wilson et al. [2020] also prove bounds on the accuracy of NS using slightly more complex assumptions but avoiding the assumption of \( \ell_2 \) regularization. More importantly, Wilson et al. [2020] also address the issue of model selection, whereas all previous
works had focused on the accuracy of NS for model assessment (i.e. assessing the error of a single, fixed model). In particular, [Wilson et al. 2020] give assumptions under which the NS approximation is accurate when used for hyperparameter tuning.

Finally, we note that in its simplest form, the NS approximation requires second differentiability of the model objective. [Obuchi and Kabashima 2016, 2018, Rad and Maleki 2020, Beirami et al. 2017, Stephenson and Broderick 2020] propose workarounds specific to models using $\ell_1$-regularization. More generally, [Wang et al. 2018] provide a natural extension of the NS approximation to models with either non-differentiable model losses or non-differentiable regularizers.

Again, while these NS methods can be applied to the structured models of interest here, the repeated computation and inversion of Hessian matrices brings their speed into question. To avoid this issue, we instead focus on approximations based on the infinitesimal jackknife (IJ) from the statistics literature [Jaeckel 1972, Efron 1981]. The IJ was recently conjectured as a potential approximation to CV by [Koh and Liang 2017] and then briefly compared against the NS approximation for this purpose by [Beirami et al. 2017]. The IJ was first studied in depth for approximating CV in an empirical and theoretical study by [Giordano et al. 2019b]. The benefit of the IJ in our application is that for any CV scheme and any (differentiable and i.i.d.) model, the IJ requires only a single matrix inverse to approximate all CV folds. [Koh et al. 2019] give further bounds on the accuracy of the IJ approximation for models using $\ell_2$ regularization. As in the case for NS, [Wilson et al. 2020] give bounds on the accuracy of IJ beyond $\ell_2$ regularized models but with slightly more involved assumptions; [Wilson et al. 2020] also give bounds on the accuracy of IJ for model selection.

Just as for the NS approximation, the IJ also requires second differentiability of the model objective. [Stephenson and Broderick 2020] deal with this issue by noting that the methods of [Wang et al. 2018] for applying the NS to non-differentiable objectives can be extended to cover the IJ as well. We note that the use of the IJ for model selection for non-differentiable objectives seems to be more complex than for the NS approximation. In particular, [Stephenson and Broderick 2020, Appendix G] show that the IJ approximation can have unexpected and undesirable behavior when used for tuning the regularization parameter for $\ell_1$ regularized models. [Wilson et al. 2020] resolve this issue by proposing a further modification to the IJ approximation based on proximal operators.

\[^1\]The methods of [Giordano et al. 2019b] apply beyond CV to other “reweight and retrain” schemes such as the bootstrap. The methods presented in this chapter apply more generally as well, although we do not explore this extension.
C.2 Hidden Markov random fields

Here we show that HMMs are instances of (hidden) MRFs. Recall that a MRF models the joint distribution,

\[- \log p(x, z; \Theta) = - \sum_{n \in [N]} \log p(x_n, z_n; \Theta) = Z(\Theta) + \sum_{n=1}^{N} \left\{ \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \left[ \sum_{c \in F} \phi_c(z_{nc}; \Theta) \right] \right\}. \tag{C.1} \]

Hidden Markov models We recover hidden Markov models as described in Section 4.2 by setting \( \psi_t(x_{nt}, z_{nt}; \Theta) = \log F(x_{nt} | \theta_z z_{nt}) \), setting \( F \) to the set of all unary and pairwise indices, and defining \( \phi_{t,t-1}(z_{nt}, z_{nt-1}; \Theta) = \log \text{Cat}(z_{nt} | z_{nt-1}) \), and \( \phi_1(z_{n1}) = \log \text{Cat}(z_{n1} | \pi) \), and \( \phi_t(z_{nt}) = 0 \), for \( t \in [T] - 1 \). The log normalization constant is \( Z(\Theta) = 0 \).

C.3 Leave structure out cross-validation (LSCV)

\begin{algorithm}
\caption{Structured approximate LSCV}
\begin{algorithmic}[1]
\Procedure{APPXLSCV}{\( \Theta_1, x, O \)}
\State Marginalize over \( z_n \): \( \log p(x_n; \Theta) = \text{MARG}(x_n; \Theta), \forall n \in [N] \)
\State Compute \( \log p(x; \Theta, w) = \sum_n w_n \log p(x_n; \Theta) \)
\State Compute \( H = \left( J_{dn} \right) := \left( \frac{\partial^2 \log p(x; \Theta, w) + \log p(\Theta)}{\partial \Theta \partial w_n} \right)_{\Theta=\Theta_1, w=1_N} \)
\State Compute \( J = (J_{dn}) := \left( \frac{\partial^2 \log p(x; \Theta, w) + \log p(\Theta)}{\partial \Theta \partial w_n} \right)_{\Theta=\Theta_1, w=1_N} \)
\For{\( o \in O \),} \( \hat{\Theta}_{\text{ACV}}(\Theta_1, x, o) := \Theta_1 + \sum_{n \in o} H^{-1} J_n \quad \triangleright J_n \) is \( n \)th column of \( J \)
\EndFor
\State \Return \{ \( \hat{\Theta}_{\text{ACV}}(\Theta_1, x, o) \) \}_{o \in O} \)
\EndProcedure
\end{algorithmic}
\end{algorithm}

C.4 Efficient weighted marginalization (WeightedMarg) for chain-structured MRFs

For chain-structured pairwise MRFs with discrete structure, we can use a dynamic program to efficiently marginalize out the structure. Assume that \( z_t, \forall t \in [T] \) can take one of \( K \) values. Define \( \alpha_{1k} = \exp[w_{1} \psi_1(x_{1}, z_{1} = k)] \), and then compute \( \alpha_{tk} \)
recursively:
\[
\alpha_{t,k} = \sum_{\ell=1}^{K} \alpha_{t-1,\ell} \exp \left[ w_t \psi_t(x_t, z_t = k; \Theta) + \phi_{t,t-1}(z_t = k, z_{t-1} = \ell) \right],
\]  
(C.2)

if using weighting scheme (A) from Equation Eq. (4.4) or,

\[
\alpha_{t,k} = \sum_{\ell=1}^{K} \alpha_{t-1,\ell} \exp \left[ w_t \psi_t(x_t, z_t = k; \Theta) + w_t \phi_{t,t-1}(z_t = k, z_{t-1} = \ell) \right],
\]  
(C.3)

if using weighting scheme (B) from Equation Eq. (4.5). Then, for either (A) or (B), we have \( p(x; \Theta, w) = \sum_{k=1}^{K} \alpha_T k \). When \( w = 1_T \) we recover the empirical risk minimization solution. As is the case for non-weighted models, this recursion implies that \( p(x; \Theta, w) \) is computable in \( O(TK^2Q) \) time instead of the usual \( O(TKQ) \) time required by brute-force summation (recall \( Q \) is the time required to evaluate one local potential). Likewise, we can also compute the derivatives needed by Algorithm 3 in \( O(TK^2Q) \) time either by manual implementation or automatic differentiation tools [Bartholomew-Biggs et al., 2000].

C.5 Equivalence of weighting (A) and (B) for leave-future-out for chain-structured graphs

As noted in the main text, weighting schemes (A) and (B) are equivalent when the graph is chain structured. Formally,

**Proposition 24.** Consider a chain-structured pairwise MRF with ordered indices \( t \) on the chain (such as an HMM). Weighting styles (A) and (B) above are equivalent for leave-future-out CV. That is, choose \( o = \{T', \ldots, T\} \) for some \( T' \in [T - 1] \) (i.e., indices that are in the “future” when interpreted as time). Then set \( \forall t \in o, w_t = 0 \) and \( \forall t \in [T] - o, w_t = 1 \).

This result does not hold generally beyond chain-structured graphical models – consider a four-node “ring” graph in which node \( t \) is connected to nodes \( t - 1 \) and \( t + 1 \) (mod 4) for \( t = 0, \ldots, 3 \). Weighting scheme (B) produces a distribution that is chain-structured over three nodes, whereas (A) produces a distribution without such conditional independence properties. We now prove Proposition 24.

**Proof.** Recall that for a chain structured graph, we can write:

\[
p(x, z) = p(x \mid z)p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}).
\]

Let \( o = \{T', T' + 1, \ldots, T\} \) for some \( T' < T \); that is, we are interested in dropping out time steps \( T', \ldots, T \). For weighting scheme (A) (Eq. (4.4)), we drop out only the
observations, obtaining:

\[
p_A(x, z; w_o) = \left( \prod_{t=1}^{T'-1} p(x_t | z_t) \right) p(z_1) \prod_{t=2}^{T} p(z_t | z_{t-1}).
\]

When we sum out all \(z\) to compute the marginal \(p_A(x; w_o)\), we can first sum over \(z_{T}, \ldots, z_{T'}\). As \(\sum_{z_t} p(z_t | z_{t-1}) = 1\) for any value of \(z_{T-1}\), we obtain:

\[
p_A(x; w_o) = \sum_{z_1, \ldots, z_{T'-1}} \left( \prod_{t=1}^{T'-1} p(x_t | z_t) \right) p(z_1) \prod_{t=2}^{T'} p(z_t | z_{t-1}),
\]

which is exactly the formula for \(p_B(x; w_o)\), the marginal likelihood from following weighting scheme (B) (Eq. (4.5)), in which we drop out both the \(x_t\) and \(z_t\) for \(t \notin o\).

\[\square\]

### C.6 Conditional random fields

Conditional random fields assume that the labels \(z\) are observed and model the conditional distribution \(p(z | x; \Theta)\). While more general dependencies between \(x\) and \(z\) are possible a commonly used variant \cite{Ma2016,Lample2016} captures the conditional distribution of the joint defined in Equation Eq. (C.1). Note,

\[
\log p(z_n | x_n; \Theta) = \log p(x_n, z_n; \Theta) - \log p(x_n; \Theta)
\]

\[
= -Z(\Theta) + \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \sum_{c \in F} \phi_c(z_{nc}; \Theta)
\]

\[
+ Z(\Theta) - \int_{z_n} \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \sum_{c \in F} \phi_c(z_{nc}; \Theta) dz_n \tag{C.4}
\]

Defining, \(Z(x_n; \Theta) = - \int_{z_n} \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \sum_{c \in F} \phi_c(z_{nc}; \Theta) dz_n\), then gives us the following conditional distribution,

\[
- \log p(z | x; \Theta) = \sum_{n=1}^{N} \left\{ Z(x_n; \Theta) + \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \sum_{c \in F} \phi_c(z_{nc}; \Theta) \right\} . \tag{C.5}
\]

Note that \(Z(x_n; \Theta)\) is an observation specific negative normalization constant.

### C.7 CV for conditional random fields

Analogously to the MRF case, we have two variants for CRFs — LSCV and LWCV. While LSCV is frequently used in practice, for example, \cite{Decaprio2007}, we are unaware of instances of LWCV in the literature. Thus, while we derive approximations to both CV schemes, our CRF-based experiments in Section 4.5 only use LSCV.
C.7.1 LSCV for CRFs

Leave structure out CV is analogous to the MRF case and is detailed in Algorithm 8 where \( \log \tilde{p}(z_n, x_n; \Theta) := \sum_{t \in [T]} \psi_t(x_{nt}, z_{nt}; \Theta) + \sum_{c \in F} \phi_c(z_{nc}; \Theta) \). Since all input, label pairs \( \{x_n, z_n\} \) are independent, \( \log p(z_n | x_n; \Theta) \) is just a weighted sum across \( n \) and the losses \( -\log p(z_n | x_n; \tilde{\Theta}^{(w)}(w_n)) \) and \( -\log p(z_n | x_n; \tilde{\Theta}^{(I)}(w_n)) \) do not depend on \([N] - n\).

**Algorithm 8** Structured approximate cross-validation (LSCV) for CRFs

1: procedure APPX LSCV FOR CRFS\((\Theta_1, x, z, O)\)
2: 3: Compute \( Z(x_n; \Theta) = -\operatorname{Marg}(x_n; \Theta), \forall n \in [N] \)
4: Compute \( \log p(z | x; \Theta, w) = \sum_n w_n [Z(x_n; \Theta) + \log \tilde{p}(z_n, x_n; \Theta)] \)
5: Compute \( H = \left. \frac{\partial^2 \log p(z | x; \Theta, w) + \log p(\Theta)}{\partial \Theta \partial w} \right|_{\Theta=\Theta_1, w=1_N} \)
6: Compute matrix \( J := (J_{dn}) = \left( \frac{\partial^2 \log p(z | x; \Theta, w) + \log p(\Theta)}{\partial \Theta \partial w_n} \right|_{\Theta=\Theta_1, w=1_N} \)
7: for \( o \in O, \hat{\Theta}_{ACV}(\Theta_1, x, z, o) := \Theta_1 + \sum_{n \in o} H^{-1} J_n \quad \triangleright J_n \text{ is } n\text{th column of } J \) return \( \{\hat{\Theta}_{ACV}(\Theta_1, x, z, o)\}_{o \in O} \)
8: end procedure

C.7.2 LWCV for CRFs

Leave within structure out for CRFs again comes with a choice of weighting scheme. Given a single input, label pair \( x, z \), the \( z_t \) are the outputs at location \( t \), and the \( x_t \) are the corresponding inputs. A form of CV arises when we drop the outputs \( z_t \), for \( t \in o \). This gives us weighting scheme \( (C) \),

\[
\hat{\Theta}(w) = \arg \min \Theta Z(\Theta, w, x) + \left[ \sum_{t \in [T]} w_t \psi_t(x_t, z_t; \Theta) + (1 - w_t) \int_{z_t} \psi_t(x_t, z_t; \Theta) \, dz_t \right] + \left[ w_t \sum_{c \in F} \phi_c(z_c; \Theta) + (1 - w_t) \int_{z_t} \sum_{c \in F} \phi_c(z_c; \Theta) \, dz_t \right] - \log p(\Theta).
\]  

(C.6)

For linear chain structured CRFs with discrete outputs \( z \) a variant of the forward algorithm can be used to efficiently compute \( Z(\Theta, w, x) \) as well as the marginalizations over \( \{z_t | t \in o\} \) required by Eq. (C.6). See Bellare and McCallum [2007], Tsuboi et al. [2008] for details. Algorithm 9 summarizes the steps involved.
Algorithm 9 Approximate leave-within-structure-out cross-validation for CRFs

1: procedure Appx LWCV for CRFs(Θ₁, x, z, O)
2: 3: Compute unweighted marginalization over z, Z(x; Θ) = −Marg(xₙ; Θ), ∀n ∈ [N]
4: 5: Compute weighted marginalization over z: Z(x; Θ, w) = WeightedMarg(x, Θ, w).
6: 7: Compute log p(z | x; Θ) = Z(x; Θ, w) + Z(xₙ; Θ)
8: 9: Compute H = ∂² log p(x; Θ, w)+log p(Θ) ∂Θ∂Θt | Θ=Θ₁, w=1, w=1
10: for o ∈ O, do:  ΔΘACV(Θ₁, x, o) := Θ₁ + ∑ Jn H⁻¹ Jn  ΔJn is nth column of J
11: end procedure

C.8 Computational cost of one Newton-step-based ACV

Recall that we define M to be the cost of one marginalization over the latent structure z and noted above that the cost of computing the Hessian via automatic differentiation is O(M). For the Newton step (NS) approximation, recall that we need to compute a different Hessian for each fold o. While this can be avoided using rank-one update rules in the case of leave-one-out CV for generalized linear models, this is not the case for the CV schemes and models considered here. Thus, to use the Newton step approximation here, we require O(M|O|) time to compute all needed Hessians. Compared to the O(M) time spent computing Hessians by our algorithms, the Newton step is significantly more expensive. For this reason, we do not consider Newton step based approximations here.

C.9 Comparison of approximations afforded by one Newton-step-based and IJ based ACV

We revisit the LWCV experiments in time varying Poisson processes described in Section 4.5. We again focus on the Tsub = 10,000 subset of observations, plotted in the top panel of Fig. 4-1. In Fig. C-1 we compare estimates provided by ACV based on one NS to those provided by IJ based ACV. The left plot depicts i.i.d LWCV and the right depicts contiguous LWCV when m = 10% of the subset is held out. Similar results hold for m = 2% and m = 5%. For each of |O| = 10 folds and for each point x_l left out in each fold, we plot a red dot with the NS bases approximate fold loss as its horizontal coordinate and our IJ based approximation as its vertical coordinate.
We can see that every point lies close to the dashed black $x = y$ line; that is, the quality of the two approximations largely agree across the thousands of points in each plot.

![Figure C-1: Comparison of NS and IJ approximate LWCV for time-varying Poisson processes. Scatter plots comparing NS based ACV loss (horizontal axis) at each point in each fold (red dots) to IJ based ACV loss (vertical axis). Black dashed line shows perfect agreement. Left plot contains i.i.d. LWCV results and the right plot contains contiguous LWCV results.](image)

C.10 Derivation of IJ approximations

In all cases considered here (i.e., the “exchangeable” leave-one-out CV considered by previous work or the more structured variants for chain-structured or general graph structured models) can be derived similarly. In particular, once we have derived the relevant weighted optimization problem for each case, the derivation of the IJ approximation is the same. Let the relevant weighted optimization problem be defined for $w \in \mathbb{R}^T$:

$$\hat{\Theta}(w) := \arg \min_{\Theta} \Theta \in \mathbb{R}^D F(\Theta, w),$$

where $F$ is some objective function with $F(\cdot, 1_T)$ corresponding to the “full-data” fit (i.e., without leaving out any data). We now follow the derivation of the IJ in Giordano et al. [2019b]. The condition that $\hat{\Theta}(1_T)$ is an exact optimum is:

$$\frac{\partial F}{\partial \Theta} |_{\hat{\Theta}(1_T), 1_T} = 0.$$
If we take a derivative with respect to $w_t$:

$$\frac{\partial^2 F}{\partial \Theta \partial \Theta^T} \bigg|_{\Theta(1_T), 1_T} \frac{d\Theta}{d w_t} \bigg|_{\Theta(1_T), 1_T} + \frac{\partial^2 F}{\partial \Theta \partial w_t} \bigg|_{\Theta(1_T), 1_T} \frac{d w_t}{d w_t} \bigg|_{\Theta(1_T), 1_T} = 0.$$ 

Noting that $d w_t / d w_t = 1$ and solving for $d \Theta / d w_t$:

$$\frac{d \Theta}{d w_t} \bigg|_{\Theta(1_T), 1_N} = - \left( \frac{\partial^2 F}{\partial \Theta \partial \Theta^T} \bigg|_{\Theta(1_T), 1_T} \right)^{-1} \frac{\partial^2 F}{\partial \Theta \partial w_t} \bigg|_{\Theta(1_T), 1_T} (1 - w_t).$$

Thus we can form a first order Taylor series of $\hat{\Theta}(w)$ in $w$ around $w = 1_N$ to approximate:

$$\hat{\Theta}_{IJ}(w) \approx \hat{\Theta}(1_T) - \sum_{t=1}^{T} \left( \frac{\partial^2 F}{\partial \Theta \partial \Theta^T} \bigg|_{\Theta(1_T), 1_T} \right)^{-1} \frac{\partial^2 F}{\partial \Theta \partial w_t} \bigg|_{\Theta(1_T), 1_T} (1 - w_t).$$

Specializing this last equation to the various $F$ and weight vectors $w$ of interest derives each of our ACV algorithms.

### C.11 Inexact optimization

We prove here a slightly more general version of Proposition 7 that covers both LWCV and LSCV, as well as arbitrary loss functions $\ell$. To encompass both in the same framework, let $w_n \in \mathbb{R}^T$ be weight vectors for each structured object $n = 1, \ldots, N$. Our weighted objective will be:

$$\hat{\Theta}(w) = \arg\min_{\Theta} \sum_{n=1}^{N} \log p(D_n; \Theta, w_n) + p(\Theta),$$

where $D = \{D_1, \ldots, D_N\}$ denotes the collection of all observed structures; i.e., each $D_n$ may be a sequence of observations $x_n$ for a HMM or observed outputs and inputs $x_n, z_n$ for a CRF. Let $\hat{\Theta}(1_{NT})$ be the solution to this problem with $w_{nt} = 1$ for all $n$ and $t$. We assume that we are interested in estimating the exact out-of-sample loss for some generic loss $\ell$ by using exact CV, $L_{CV} := (1/|O|) \sum_{o} \ell(D_o, D_{-o}, \hat{\Theta}(w_o))$; e.g., we may have $\ell(D_o, D_{-o}, \hat{\Theta}(w_o)) = -\log p(x_o | x_{[T]_{-o}}; \hat{\Theta}(w_o))$ in the case of a HMM with $N = 1$. Notice here that $o \subset [N] \times [T]$ indexes arbitrarily across structures. We can now state a modified version of Assumption 11.

**Assumption 24.** Let $B \subset \mathbb{R}^D$ be a ball centered on $\hat{\Theta}(1_{NT})$ and containing $\Theta(S)$. Then the objective $\sum_n \log p(x_n; \Theta, 1_T) + p(\Theta)$ is strongly convex with parameter $\lambda_{\text{min}}$ on $B$. Additionally, on $B$, the derivatives $g_n(\Theta) := \partial^2 \log p(x_n; \Theta, w_n)/\partial \Theta \partial w_{nt}$ are Lipschitz continuous with constant $L_g$ for all $n, t$ and the inverse Hessian of the objective is Lipschitz with parameter $L_{\text{Hinv}}$. Finally, on $B$, $\ell(D_o, D_{-o}, \Theta)$ is a Lipschitz function of $\Theta$ with parameter $L_\ell$ for all $o$.

We now prove our more general version Proposition 7.
**Proposition 25.** Take Assumption 24. Then the approximation error of $L_{IJ}(\Theta^{(S)})$ is bounded by:

$$|L_{IJ}(\Theta^{(S)}) - L_{CV}| \leq C\varepsilon_\Theta + \varepsilon_{IJ},$$

(C.8)

where $C$ is given by

$$\left( L_\ell + \frac{L_\ell L_g}{\lambda_{\min}} + \frac{L_\ell L_{Hinv}}{|\mathcal{O}|} \sum_o \left\| \sum_{t \in o} g_{nt}(\hat{\Theta}(1_{NT})) \right\|_2 \right).$$

**Proof.** By the triangle inequality:

$$|L_{IJ}(\Theta^{(S)}) - L_{CV}| \leq$$

$$|L_{IJ}(\Theta^{(S)}) - L_{IJ}(\hat{\Theta}(1_{NT}))|$$

$$+ |L_{IJ}(\hat{\Theta}(1_{NT})) - L_{CV}|.$$  

The second term is just the constant $\varepsilon_{IJ}$. Now we just need to bound the first term using our Lipschitz assumptions. We have, by the triangle inequality

$$|L_{IJ}(\hat{\Theta}(1_{NT})) - L_{IJ}(\Theta^{(S)})|$$

$$\leq \frac{1}{|\mathcal{O}|} \sum_o \ell \left( D_o, \hat{\Theta}(1_{NT}) + H^{-1}(\hat{\Theta}(1_{NT})) \sum_{t \in o} g_{nt}(\hat{\Theta}(1_{NT})) \right)$$

$$- \ell \left( D_o, \hat{\Theta}(1_{NT}) + H^{-1}(\Theta^{(S)}) \sum_{t \in o} g_{nt}(\Theta^{(S)}) \right).$$

Continuing to apply the triangle inequality and our Lipschitz assumptions:

$$\leq \frac{L_\ell}{|\mathcal{O}|} \sum_o \left( \left\| \hat{\Theta}(1_{NT}) - \Theta^{(S)} \right\|_2 \right.$$

$$+ \left\| H^{-1}(\hat{\Theta}(1_{NT})) \sum_{t \in o} g_{nt}(\hat{\Theta}(1_{NT})) - H^{-1}(\Theta^{(S)}) \sum_{t \in o} g_{nt}(\Theta^{(S)}) \right\|_2 \right)$$

$$\leq L_\ell \varepsilon_\Theta + \frac{L_\ell}{|\mathcal{O}|} \sum_o \left\| H^{-1}(\Theta^{(S)}) \sum_{t \in o} \left( g_{nt}(\hat{\Theta}(1_{NT})) - g_{nt}(\Theta^{(S)}) \right) \right\|_2$$

$$+ \frac{L_\ell}{|\mathcal{O}|} \sum_o \left\| \left( H^{-1}(\hat{\Theta}(1_{NT})) - H^{-1}(\Theta^{(S)}) \right) \sum_{t \in o} g_{nt}(\hat{\Theta}(1_{NT})) \right\|_2$$

$$\leq \left( L_\ell + \frac{L_\ell L_g}{\lambda_{\min}} + \frac{L_\ell L_{Hinv}}{|\mathcal{O}|} \sum_o \left\| \sum_{t \in o} g_{nt}(\hat{\Theta}(1_{NT})) \right\|_2 \right) \varepsilon_\Theta.$$

Defining the term in the parenthesis as $C$ finishes the proof.

As noted after the statement of Proposition 7 in the main text,
may depend on \(T, N\) or \(O\), but we expect it to converge to a constant given reasonable distributional assumptions on the data. To build intuition, we consider the case of leave-one-out CV for generalized linear models, where we observe a dataset of size \(N > 1\) and have \(T = 1\). In particular, we have 
\[
\log(x_n, y_n; \Theta) = f(x_n^T \Theta, y_n),
\]
where \(x_n \in \mathbb{R}^D\) are the covariates and \(y_n \in \mathbb{R}\) are the responses. In this case, 
\[
g_{nt} = \hat{D}_n^{(1)} x_n, \quad \text{where} \quad \hat{D}_n^{(1)} = \frac{df(z)}{dz} \bigg|_{z = x_n^T \hat{\Theta}(1_T)}.
\]
Then, given reasonable distributional assumptions on the covariates and some sort of control over the derivatives \(\hat{D}_n^{(1)}\), we might suspect that 
\[
(1/N) \sum_{n=1}^N \|\nabla f(\hat{\Theta}(1_T), x_n)\|_2 - \sqrt{D} \geq t
\]
will converge to a constant. As an example, we consider logistic regression with sub-Gaussian data, for which we can actually prove high-probability bounds on this sum.

**Definition 9** (e.g., Vershynin [2018]). For \(c_x > 0\), a random variable \(V\) is \(c_x\)-sub-Gaussian if
\[
\mathbb{E}\left[\exp\left(V^2/c_x^2\right)\right] \leq 2.
\]

**Proposition 26.** For logistic regression, assume that the components of the covariates \(x_{nt}\) are i.i.d. from a zero-mean \(c_x\)-sub-Gaussian distribution for \(d = 1, \ldots, D\). Then we have that, for any \(t \geq 0\):
\[
\Pr\left[\left|\frac{1}{N} \sum_{n=1}^N \|\nabla f(\hat{\Theta}(1_T), x_n)\|_2 - \sqrt{D}\right| \geq t \right] \leq \exp\left[-C \frac{N t^2}{c_x^2}\right],
\]
where \(C > 0\) is some global constant, independent of \(N, D\), and \(c_x\).

**Proof.** First, we can use the fact that 
\[
\|\nabla f(\hat{\Theta}(1_T), x_n)\|_2 \leq \|x_n\|_2,
\]
as for logistic regression, \(|\hat{D}_n^{(1)}| \leq 1\). Next, we can use the fact that \(\|x_n\|_2 - \sqrt{D}\) is a zero-mean sub-Gaussian random variable by Theorem 3.1.1 of Vershynin [2018]. We can then apply Hoeffding’s inequality [Vershynin, 2018, Theorem 2.6.3] to complete the proof.

**C.12 Experimental details**

We provide further experimental details in this section.

**C.12.1 Time varying Poisson processes**

We briefly summarize the time-varying Poisson process model from Ihler et al. [2006] here. Our data is a time series of loop sensor data collected every five minutes over a span of 25 weeks from a section of a freeway near a baseball stadium in Los Angeles. In all, there are 50,400 measurements of the number of cars on that span of the freeway. Ihler et al. analyze the resulting time series of counts \(x_t\) to detect the presence or absence of an event at the stadium. Following their model, we use a background Poisson process with a time varying rate parameter \(\lambda_t\) to model non-event counts, \(x_{b_t} \sim \text{Poisson}(\lambda_t)\). To model the daily variation apparent in the data, we define \(\lambda_t = \lambda_0 \delta_{d_t}\), where \(d_t\) takes one of seven values, each corresponding to one
day of the week and \([\delta_1/7, \ldots, \delta_7/7] \sim \text{Dir}(1, \ldots, 1)\). We use binary latent variables \(z_t\) to indicate the presence or absence of an event and assume a first order Markovian dependence, \(z_t \mid z_{t-1} \sim A_{z_{t-1}}\). Next, \(z_t = 0\) indicates a non-event at time step \(t\) and the observed counts are generated as \(x_t = x_{b_t}\). An event at time step \(t\) corresponds to \(z_t = 1\) and \(x_t = x_{b_t} + x_{e_t}\), and \(x_{e_t} \sim \text{NegBinomial}(x_{e_t} \mid a, b/(1+b))\), where \(x_{e_t}\) are unobserved excess counts resulting from the event. We place Gamma priors on \(\lambda_0, a, b\) and Beta priors on \(A_{00}\) and \(A_{11}\), and learn the MAP estimates of the parameters \(\Theta = \{\lambda_0, \delta_1, \ldots, \delta_7, a, b, A\}\) while marginalizing \(x_{e_t}\) and \(z_1, \ldots, z_T\). We refer the interested reader to [Ihler et al., 2006] for further details about the model and data.

**Contiguous LWCV.** In contiguous LWCV we leave out contiguous blocks from a time series. To drop \(m\)% of the data, we sample an index \(t\) uniformly at random from \([\lfloor mT/100 \rfloor + 1, \ldots, T]\) and set \(o = \{t - \lfloor mT/100 \rfloor, \ldots, t\}\).

**Numerical values from Fig. 4-1** In Table C.1 we present an evaluation of the LWCV approximation quality for time-varying Poisson processes. The results presented are a numerical summary of the results visually illustrated in Fig. 4-1. Table C.1: Evaluation of approximate LWCV for time-varying Poisson processes. Mean ACV relative error, \(|\text{acv} - \text{cv}|/\text{cv}\) and two standard deviations, over ten folds with \(T = 10000\). The numbers summarize the scatter plots in the lower left six panels of Fig. 4-1. The column headers indicate the percentage of data in the held out fold.

<table>
<thead>
<tr>
<th></th>
<th>2 %</th>
<th>5%</th>
<th>10 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.i.d</td>
<td>0.005 ± 0.009</td>
<td>0.006 ± 0.01</td>
<td>0.006 ± 0.005</td>
</tr>
<tr>
<td>contiguous</td>
<td>0.003 ± 0.003</td>
<td>0.007 ± 0.02</td>
<td>0.007 ± 0.006</td>
</tr>
</tbody>
</table>

Table C.2: Wall clock time from the two lower right panels in Fig. 4-1 at \(T = 50000\) and with \(m\)% = 10% of the data in the held out fold.

<table>
<thead>
<tr>
<th>T</th>
<th>i.i.d ACV</th>
<th>i.i.d ACV (NS)</th>
<th>Exact CV</th>
<th>contiguous ACV</th>
<th>contiguous ACV (NS)</th>
<th>Exact CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>1.1 mins</td>
<td>10.5 hours</td>
<td>61.1 hours</td>
<td>1.3 mins</td>
<td>10.5 hours</td>
<td>61.3 hours</td>
</tr>
<tr>
<td>10000</td>
<td>2.2 mins</td>
<td>19.9 hours</td>
<td>185.8 hours</td>
<td>2.4 mins</td>
<td>19.9 hours</td>
<td>182.4 hours</td>
</tr>
<tr>
<td>50000</td>
<td>11.0 mins</td>
<td>98.6 hours</td>
<td>682.2 hours</td>
<td>10.6 mins</td>
<td>99.1 hours</td>
<td>683.9 hours</td>
</tr>
</tbody>
</table>

**C.12.2 Neural CRF**

We employed a bi-directional LSTM model with a CRF output layer. We used a concatenation of a 300 dimensional Glove word embeddings [Pennington et al., 2014] and
a character CNN [Ma and Hovy 2016] based character representation. We employed variational dropout with a dropout rate of 0.25. The architecture is detailed below.

```
LSTMCRFVD(
  (dropout): Dropout(p=0.25, inplace=False)
  (char_feats_layer): CharCNN(
    (char_embedding): CharEmbedding(
      (embedding): Embedding(96, 50, padding_idx=0)
      (embedding_dropout): Dropout(p=0.25, inplace=False)
    )
    (cnn): Conv1d(50, 30, kernel_size=(3,), stride=(1,), padding=(2,))
  )
  (word_embedding): Embedding(2196016, 300)
  (rnn): StackedBidirectionalLstm(
    (forward_layer_0): AugmentedLstm(
      (input_linearity): Linear(in_features=330, out_features=200, bias=False)
      (state_linearity): Linear(in_features=50, out_features=200, bias=True)
    )
    (backward_layer_0): AugmentedLstm(
      (input_linearity): Linear(in_features=330, out_features=200, bias=False)
      (state_linearity): Linear(in_features=50, out_features=200, bias=True)
    )
    (forward_layer_1): AugmentedLstm(
      (input_linearity): Linear(in_features=100, out_features=200, bias=False)
      (state_linearity): Linear(in_features=50, out_features=200, bias=True)
    )
    (backward_layer_1): AugmentedLstm(
      (input_linearity): Linear(in_features=100, out_features=200, bias=False)
      (state_linearity): Linear(in_features=50, out_features=200, bias=True)
    )
    (layer_dropout): InputVariationalDropout(p=0.25, inplace=False)
  )
  (rnn_to_crf): Linear(in_features=100, out_features=9, bias=True)
  (crf): ConditionalRandomField()
)
```

Training We used Adam for optimization. Following the recommendation of Reimers and Gurevych [2017] we used mini-batches of size 31 [Reimers and Gurevych 2017]. We employed early stopping by monitoring the loss on the validation set. Freezing all but the CRF layers we further fine-tuned only the CRF layer for an additional 60 epochs. In Fig. C-2 we plot the mean absolute approximation error in the held out probability under exact CV and our approximation across all 500 folds as a function of (wall clock) time taken by the optimization procedure.

C.12.3 Philadelphia crime experiment

Our crime data comes from [opendataphilly.org](http://opendataphilly.org) where the Philadelphia Police Department publicly releases the time, type, and location of every reported time. For each census tract, we have a latent label $z_t \in \{-1, 1\}$, and model the number of reported crimes $x_t$ with a simple Poisson mixture model: $x_t|z_t \sim \text{Poisson}(\lambda_{z_t})$ where $\lambda_{-1}, \lambda_1 > 0$ are the unknown mean levels of crime in low- and high-crime areas, respectively. Since we might expect adjacent census tracts to be in the same latent state, we model the $z_t$’s with an MRF so that

$$
\log p(x, z; \Theta) = \sum_t [-\lambda_{z_t} + x_t \log \lambda_{z_t} - \log(x_t!)] + \beta \sum_{t' \in V(t)} 1\{z_t = z_{t'}\} - \log Z(\beta)
$$
where \( \Theta = \{ \lambda_1, \lambda_2 \} \), \( \Gamma(t) \) is the collection of census tracts that are spatially adjacent to census tract \( t \) and \( \log Z(\beta) \) is the log normalizer for the latent field \( p(z) \). The potential \( \mathbf{1}\{z_t = z_{t'}\} \) expresses prior belief that adjacent census tracts should be in the same latent class. The connection strength \( \beta \) is treated as a hyper-parameter. For each \( \beta \) fixed, \( \Theta \) is estimated using expectation maximization [Dempster et al., 1977] on \( \sum_z \log p(x, z; \Theta) \). M-step computation is analytical, given the posteriors \( p(z_t | x; \Theta) \). Exact E-step computation is reasonably efficient through smart variable elimination [Koller and Friedman, 2009, Chapter 9]: the number of states is small and common heuristics to find good elimination orderings, such as MinFill, worked well. This efficient variable elimination order is also used to implement the \texttt{WEIGHTEDMARG} routine of 3.

### C.13 Additional experiments

We present additional experimental validation in support of the ACV methods in this section.

#### C.13.1 Motion capture analysis

**Data.** We analyze motion capture recordings from the CMU MoCap database (http://mocap.cs.cmu.edu), which consists of several recordings of subjects performing a shared set of activities. We focus on the 124 sequences from the “Physical activities and Sports” category that has been previously been studied [Fox et al., 2009] [Hughes et al., 2012] [Fox et al., 2014] in the context of unsupervised discovery of shared activities from the observed sequences. At each time step we retain twelve measurements deemed informative for describing the activities of interest, as recommended by Fox et al.. Auto-regressive hidden Markov models have been shown effective for this task, motivating their use in this section.

**Accurate LSCV — auto-regressive HMMs** We confirm here that ACV is accurate and computationally efficient for structured models in the case studied by previous work: LSCV with exact model fits. We present comparisons between embarrassingly parallel exact CV and LSCV with parallelized Hessian computation (“Approx. Parallel”, i.e., we parallelize the Hessian computation over different structures \( n \)), alleviating the primary computational bottleneck for ACV. We model the collection of MoCAP sequences via a \( K \)-state HMM with an order-p auto-regressive
(AR(p)) observation model. We also consider variants where each state’s auto-regressive model is parameterized via a neural network. Figure C.3 visualizes a MoCAP sequence where we have retained only the 12 relevant dimensions. For this experiment, we retain up to 100 (= T) measurements per sequence. We employ the following auto-regressive observation model,

\[ p(x_{nt} \mid x_{nt-1}, \ldots, x_{nt-p}, z_{nt}) = \mathcal{N}(x_{nt} \mid \sum_{m=1}^{p} B_{z_{nt}} x_{nt-m} + b_{z_{nt}}, \sigma^2 I), \]

where \( p \) is the order of the auto-regression. Neural auto-regressive observation models are defined as,

\[ p(x_{nt} \mid x_{nt-1}, \ldots, x_{nt-p}, z_{nt}) = \mathcal{N}(x_{nt} \mid B_{1}^{1} h(\sum_{m=1}^{p} B_{0}^{0} x_{nt-m} + b_{0}^{0} z_{nt}) + b_{1}^{1} z_{nt}, \sigma^2 I), \]

where \( \theta_{k} = \{B_{0}^{0}, \theta_{k}, B_{1}^{1}, b_{0}^{1}\} \), and \( h \) denotes a tanh non-linearity, and \( B_{k}^{0}, B_{k}^{1} \in \mathbb{R}^{12 \times 12} \) and \( b_{k}^{0}, b_{k}^{1} \in \mathbb{R} \), i.e., a 12-12-12 fully connected network.

While past work has explored AR(0) and AR(1) observation models, a thorough exploration of the effect of \( p \) has been lacking. ACV provides an effective tool for exploring such questions accurately and inexpensively. We split the sequences into a 80/20% train and test split and perform LSCV on the training data (\( N = 100 \)) to compare AR(p) models with \( p \) ranging from zero through five and the neural variant with \( p = 1 \) (NAR(1)), in terms of how well they describe the left out sequence. Following Fox et al., we fix \( K = 16 \). Figure C.3 summarizes our results. First, we see that the ACV loss is quite close to the exact CV loss and that both track well with the held-out test loss. Furthermore, consistent with previous studies, we find that using an AR(1) observation model is significantly better than using an AR(0) or higher-order AR model. Interestingly, the out-of-sample loss for the AR(1) model is comparable to neural variant, NAR(1).

In terms of computation, the ACV is significantly faster than exact CV. In fact, for the higher order auto-regressive likelihoods and the neural variant, exact CV was too expensive to perform. Instead, we report estimated time for running such experiments by multiplying the average time taken to run three folds of LSCV with the number of training instances. For AR(0) and AR(1) we compare against exact CV implemented via publicly available optimized Expectation Maximization code [Hughes and Sudderth, 2014]. The higher order AR and the NAR(1) model, were fit by BFGS as implemented in scipy.optimize.minimize. We find that computing the embarrassingly parallel version provides significant speedups over their serial counterparts.

**Accurate LWCV for MoCAP** Next, we present LWCV results on a 1,484 measurement long sequence extracted from the MoCAP dataset. We explore three variants of LWCV: i.i.d LWCV, contiguous LWCV, and a special case of contiguous LWCV: leave-future-out CV. Figures C.4 and C.5 present these results. We find that the IJ approximations again provide accurate approximations to exact CV. The performance deteriorates for contiguous LWCV when large chunks of the sequence are left out. Since large changes to the sequence result in large changes to the fit parameters, a Taylor series approximation about the original fit is less accurate. Also, for high dimensional models such as NAR(1) IJ approximations tend to be less accurate [Stephenson and Broderick, 2020], explaining the drop in LFOCV performance for the NAR(1) model.
Figure C-3: Motion capture analysis through auto-regressive HMMs. *(Top)* A twelve dimensional MoCap sequence that serves as the observed data and the number of parameters $D$ for different models under consideration. The high dimensionality of the models make alternate ACV methods based on a single Newton step infeasible. *(Middle)* Scatter plots comparing leave one out loss, where x-axis is $-\ln p(x_n | \Theta(w_{(n)}))$ and y-axis is $-\ln p(x_n | \hat{\Theta}_{IJ}(w_{(n)}))$ for different auto-regressive orders under exact and IJ approximated leave one out cross validation. Points along the diagonal indicate accurate IJ approximations. *(Bottom)* Timing and held out negative log probability across different models. For IJ and Exact the error bars represent two jackknife standard error. The IJ approximations are significantly faster but closely approximate exact leave one out loss across models and track well with test loss computed on the held out 20% of the dataset.
Figure C-4: Within sequence leave out experiments. We took the longest MoCAP sequence containing 1484 measurements and fit a five state HMM with Gaussian emissions. We find that even for the MoCAP data IJ approximations to i.i.d. LWCV is very accurate. As the contiguous LWCV involves making larger scale changes to the sequence, for instance at 10% we end up dropping chunks of 140 time steps from the sequences, resulting in larger changes to the parameters, IJ approximations are relatively less accurate. (Top) Scatter plots comparing i.i.d LWCV loss $-\ln p(x_t | x_{[T]-o}; \Theta(w_o))$ (horizontal axis) with $-\ln p(x_t | x_{[T]-o}; \hat{\Theta}_{IJ}(w_o))$ (vertical axis), for each point $x_t$ left out in each fold, computed under exact CV for different omission rates $m\% = 2\%, 5\%$, and 10% on $M = 10$ trials. (Bottom) Results for contiguous LWCV.
Figure C-5: Leave Future Out CV for MoCAP data on a single MoCAP sequence containing 1484 measurements. The scatter plots compare $-\ln p(x_{T'} \mid x_{[T]-o}; \Theta(w_o))$ (horizontal axis) with $-\ln p(x_{T'} \mid x_{[T]-o}; \hat{\Theta}_{11}(w_o))$ (vertical axis), with $o = \{T', T'+1, \ldots, T\}$, for some $T' \leq T$, for a five state HMM with Gaussian emissions (left), order 1 auto-regressive emissions (middle), neural auto-regressive emissions (right). The rightmost plot shows the number of parameters in each model. We vary $T'$ from 1337 to 1484 for Gaussian and AR(1) emissions. Since exact fits the NAR model are more expensive we only vary $T'$ between 1455 and 1484 for NAR(1). We find that ACV to be accurate. The NAR model which is an instance of a higher dimensional optimization problem, leads to approximations that are less accurate than the lower dimensional AR(0) and AR(1) cases.
Appendix D

Appendix for: can we globally optimize cross-validation loss?  
Quasiconvexity in ridge regression (Chapter 5)

D.1 Real dataset descriptions

Here, we give the details for the real datasets used to generate Fig. 5-1.

Life expectancy. Our first real dataset contains \( N = 2,938 \) observations of life expectancy in a country, along with \( D = 20 \) covariates such as country of origin or alcohol use. The dataset is available from [Rajarshi, 2018]. In this case, \( \mathcal{L} \) for the full dataset is quasiconvex. But now consider some standard data pre-processing. Practitioners often perform principal component regression (PCR) with the aim of reducing noise in the estimated \( \theta \). That is, they take the singular value decomposition of \( X = USV \); they then produce an \( N \times R \) dimensional covariate matrix \( X' \) by retaining just the upper \( R \) singular values of \( X' \): \( X' = U_{:,R}S_{:,R} \). If we include this pre-processing step, the resulting LOOCV curve \( \mathcal{L} \) is non-quasiconvex for many values of \( R \); in the center panel of Fig. 5-1 we show one example for \( R = 15 \).

This dataset does contain information about people. However, it is only reported at the aggregated level by a given country per year. It is not clear to us whether or not consent was obtained by the individuals living in these countries; however, we feel the publication of such data is unlikely to negatively affect any given individual. Additionally, while we do not know if the data reveals any identifying information about an individual, we feel it is unlikely to do so, as it is published at the country level.

Wine dataset. Our second dataset consists of recorded wine quality of \( N = 1,599 \) red wines. The goal is to predict wine quality from \( D = 11 \) observed covariates relating to the chemical properties of each wine [Cortez et al., 2009a,b]. We find that subsets of this dataset often exhibit non-quasiconvex \( \mathcal{L} \). We search over 400 random subsets of this dataset of size \( N = 50 \). In Fig. 5-1, Twelve of these led to non-quasiconvex losses \( \mathcal{L} \), and Fig. 5-1 shows one of these examples.

This dataset does not contain information about people, and so concerns about consent and personally identifying information do not seem relevant here.
D.2 Proof of Proposition 9

We now restate and then prove Proposition 9

Proposition 9. Assume Condition 3 holds. The quasiconvexity of $\mathcal{L}$ is independent of the following

1. The matrix of right singular vectors, $V$
2. The norm of the responses, $\|Y\|_2$
3. The scaling of the singular values (i.e. changing $S$ into $S/c$ for $c \in \mathbb{R}_{>0}$)

in the sense that altering any of these quantities does not change whether or not $\mathcal{L}$ is quasiconvex.

Proof. First, it is easiest to write our function of interest in a simpler form:

$$\mathcal{L}(\lambda) = \sum_{n=1}^{N} \frac{1}{(1 - Q_n(\lambda))^2} (x_n^T \hat{\theta}_\lambda - y_n)^2,$$

(D.1)

where $Q_n(\lambda) := x_n^T (X^T X + \lambda I_D)^{-1} x_n$ and $\hat{\theta}_\lambda := (X^T X + \lambda I_D)^{-1} X^T Y$.

Let the singular value decomposition of $X$ be $X = U \text{diag}(S)V$. Then:

**V does not affect does not affect the quasiconvexity of $\mathcal{L}$.** To prove this claim, note that $x_n = u_n^T \text{diag}(S)V$, where $u_n$ is the $n$th row of $U$. So:

$$Q_n(\lambda) = u_n^T \text{diag}(S)V^T (V \text{diag}(S^2 + \lambda)^{-1} V^T) V \text{diag}(S) u_n$$

$$= u_n^T \text{diag}(S) \text{diag}(S^2 + \lambda)^{-1} \text{diag}(S) u_n.$$

So $Q_n(\lambda)$ is actually independent of $V$. Next,

$$x_n^T \hat{\theta}_\lambda = u_n^T \text{diag}(S)V^T V \text{diag}(S^2 + \lambda)^{-1} V^T V \text{diag}(S) U^T Y$$

$$= u_n^T \text{diag}(S) \text{diag}(S^2 + \lambda)^{-1} \text{diag}(S) U^T Y,$$

which is also independent of $V$.

**$\|Y\|_2$ does not affect the quasiconvexity of $\mathcal{L}$.** In particular, we can treat $Y$ as sitting on the $D$-dimensional unit sphere. To see this, take two different $Y$'s related by a scaling: $y_n(1) = cy_n(0)$ for some scalar $c \in \mathbb{R}$. Then, using the same superscripts:

$$\hat{\theta}_\lambda^{(1)} = (X^T X + \lambda I_D)^{-1} X^T Y^{(1)} = c \hat{\theta}_\lambda^{(0)}.$$ 

So, we can relate the two LOOCV functions by:

$$\mathcal{L}^{(1)}(\lambda) = \sum_{n} \frac{1}{(1 - Q_n(\lambda))^2} (c x_n^T \hat{\theta}_\lambda^{(0)} - c y_n^{(0)})^2 = c^2 \mathcal{L}^{(0)}(\lambda).$$

(D.2)

So scaling $Y$ by $c$ uniformly scales $\mathcal{L}(\lambda)$ by $c^2$. Mutliplying $\mathcal{L}$ by a constant does not affect is quasiconvexity.

**The scaling of the singular values $s_1, \ldots, s_D$ does not affect the quasiconvexity of $\mathcal{L}$.** We have already shown that $V$ does not affect the quasiconvexity of $\mathcal{L}$, so fix $V = I_D$ to simplify the proof. Pick some scaling $c > 0$, and fix some spectrum $S^{(1)}$. Define $S^{(0)} := c S^{(1)}$. Using the same superscripts, we have:

$$Q_n^{(0)}(\lambda) = \sum_{d=1}^{D} u_{nd}^2 \frac{c^2 s_d^2}{c^2 s_d^2 + \lambda} = \sum_{d=1}^{D} u_{nd}^2 \frac{s_d^2}{s_d^2 + (\lambda/c^2)} = Q_n^{(1)} \left(\frac{\lambda}{c^2}\right)$$

(D.3)
Figure D-1: (Left): Severity of non-quasiconvexity where \( \lambda_{\text{worst-min}} \) is the \( \lambda \) recovered by grid search. (Right): Severity of non-quasiconvexity where \( \lambda_{\text{worst-min}} \) is the \( \lambda \) recovered by gradient descent. Note the color scales on the left and right figures differ slightly and also differ significantly from Fig. 5-2 in the main text.

Similarly, define \( (x_n^T \hat{\theta})^{(0)}(\lambda) \) to be the inner product of \( x_n^{(0)} \) and \( \hat{\theta}^{(0)}(\lambda) \). Then:

\[
(x_n^T \hat{\theta})^{(0)}(\lambda) = u_n^T \text{diag}(S^{(0)}) \left( \text{diag}(S^{(0)})^2 + \lambda I_D \right)^{-1} \text{diag}(S^{(0)}) U^TY \tag{D.4}
\]

\[
= u_n^T \text{diag}(S^{(1)}) \left( \text{diag}(S^{(1)})^2 + \frac{\lambda}{c^2} I_D \right)^{-1} \text{diag}(S^{(1)}) U^TY \tag{D.5}
\]

\[
= (x_n^T \hat{\theta})^{(1)} \left( \frac{\lambda}{c^2} \right). \tag{D.6}
\]

This, along with Eq. (D.3) implies that \( L^{(0)}(\lambda) = L^{(1)}(\lambda/c^2) \). That is, multiplying the singular values by \( c \) stretches out \( L \) by a factor of \( c \). This does not change the quasiconvexity of \( L \).

### D.3 Measuring severity of non-quasiconvexity with other losses and optimization methods

Fig. 5-2 shows the severity of non-quasiconvexity when searching over all possible \( N = 3, D = 2 \) regression problems for a fixed spectrum \( S \). Recall that we measured the severity of non-quasiconvexity as

\[
\text{severity} := \frac{L(\lambda_{\text{worst-min}}) - L(\lambda^*)}{L(\lambda_{\text{worst}}) - L(\lambda^*)}, \tag{D.7}
\]

where \( \lambda_{\text{worst}}, \lambda^* \), and \( \lambda_{\text{worst-min}} \) are the \( \lambda \) maximizing \( L \), the \( \lambda \) minimizing \( L \), and the \( \lambda \) corresponding to the local minimum with largest \( L \), respectively. Eq. (D.7) measures the relative quality of the optima found by a hypothetical optimizer that always finds the worst possible local optimum, where the quality of an optimum is measured by \( L \). We view even modest values of this measure of severity as fairly severe: e.g. a severity of 0.1 indicates that the excess loss incurred by finding a bad minimum is 10% of the excess loss incurred by using the worst possible \( \lambda \). Here, we investigate what happens when we replace different parts of Eq. (D.7).

**Realistic optimization methods.** First, we ask what would happen when using more realistic optimization methods. In particular, we consider replacing \( \lambda_{\text{worst-min}} \) by the \( \lambda \) of the minimum reported by either gradient descent or grid search. For gradient descent, we have to choose a \( \lambda \) at

158
which to initialize. There does not seem to be any natural data-driven heuristic to initialize gradient
descent with, so we always initialize at $\lambda = 0$. For grid search, we need to choose the range of our
grid and how fine a grid we will use. For the leftmost point of our grid, we use $\lambda = 0$. For the
rightmost point of our grid, we use $\lambda = 2$. To justify this choice of a maximum $\lambda$, recall we have set
the maximum singular value to be equal to 1. Thus $\lambda > 1$ implies that the regularizer has exceeded
the scale of the covariates; it seems reasonable to not consider $\lambda$ too much larger than this scale, so
we select $\lambda = 2$ as our maximum. We assume that if no minimum is encountered in the range of
our grid that any reasonable user would continue the grid search until finding a minimum; in such
a case, we say that grid search reports the minimum with the smallest $\lambda$ value. Given this range
of $\lambda$’s, we set our grid to be 200 $\lambda$’s evenly spaced grid on the log-scale. We have never found a
difference in our experiments by making this grid finer, so we expect this has no effect on the results
presented here.

We could replicate the $N = 3, D = 2$ experiment of Fig. 5-2 using grid search and gradient
descent. However, it seems that in the case of $N = 3, D = 2$, the leftmost minimum of $L$ is always
at some $\lambda < 1$ and is always the global optimum of $L$. So, our implementations of grid search and
gradient descent always find the global optimum of $L$ in this case. To create examples with more
interesting behavior, we instead consider the case of $N = 50, D = 2$. Here, we can still parameterize
$\theta^*$ by a scalar on the unit circle, but cannot do the same for $U$. Instead, we vary $\theta^*$ and the singular
values of $X$ over a grid and average the severity over many random settings of $U$ for each setting
of $S$ and $\theta^*$. We set the singular values of $X$ to be $S_1 = 1, S_2 = \alpha$, for some $\alpha \in (0, 1]$. We let $\alpha$
vary on a log-scale grid between 0 and 1 and $\theta^*$ on a grid between 0 and $2\pi$. For each setting of
$\alpha$ and $\theta^*$, we construct 1000 random linear regression problems by drawing $U$ at uniform from all
zero-column-mean orthonormal matrices (Appendix D.7). We then set $Y = U \text{diag}(S)\theta^* + E$, where
$E \in \mathbb{R}^N$ has i.i.d. $\mathcal{N}(0, 0.1)$ entries. We run either grid search or gradient descent on the resulting
LOOCV loss $L$ and compute the severity of its non-quasiconvexity as:

\[
K(\lambda_{\text{worst-min}}) - K(\lambda^*) - K(\lambda_{\text{worst}}) - K(\lambda^*),
\]

Figure D-2: (Left): Severity of non-quasiconvexity in 5-fold CV. (Centre): Severity of non-quasiconvexity in 10-fold CV. (Right): Severity of non-quasiconvexity in $\mathcal{L}$, where the quality of $\lambda_{\text{worst}}, \lambda^*$ and $\lambda_{\text{worst-min}}$ are measured using the test loss. Note that the color scale on this plot is different from that on previous plots.

Other CV losses. One might ask if our finding of non-quasiconvexity is limited to leave-one-
out CV. To show this is not the case, we consider the same $N = 50, D = 2$ setting as above, and
again vary $\theta^*$ and $\alpha$ on a grid. We let $\mathcal{K}(\lambda)$ be the $K$-fold CV loss and compute the severity of its
non-quasiconvexity as:

\[
\frac{\mathcal{K}(\lambda_{\text{worst-min}}) - \mathcal{K}(\lambda^*)}{\mathcal{K}(\lambda_{\text{worst}}) - \mathcal{K}(\lambda^*)},
\]
Figure D-3: Experiment from Appendix D.4. Orange dots show the decay of $\|u_{\text{max}}\|^2$ for uniformly drawn non-zero-column-mean orthonormal matrices $U$; we see that, as proven by Candés and Recht [2009], these matrices satisfy Assumption 14. Blue dots show the decay of $\|u_{\text{max}}\|^2$ for uniformly drawn zero-column-mean orthonormal matrices. While such matrices are not known to satisfy Assumption 14, we see that their $\|u_{\text{max}}\|^2$ decays at exactly the same rate as the non-zero-column-mean matrices.

where $\lambda_{\text{worst-min}}, \lambda^*, \text{ and } \lambda_{\text{worst}}$ are the $\lambda$ of the local minimum of $K$ with highest loss, the global minimum of $K$, and the $\lambda$ maximizing $K$, respectively. In the left and center of Fig. D-2 we show the result for 5-fold and 10-fold CV, respectively. We find similar results as before; there are settings of $\theta^*$ and $\alpha$ for which the $K$-fold CV loss has severe non-quasiconvexity.

**Test loss.** Ultimately, we are hoping to find a $\lambda$ such that the test loss $T(\lambda)$ is small; we hope that minimizing $\mathcal{L}$ will give us such a $\lambda$. Here, we ask whether or not the presence of non-quasiconvexity in $\mathcal{L}$ can make it hard to find such a $\lambda$. Here, we define the severity of non-quasiconvexity as:

$$\frac{T(\lambda_{\text{worst-min}}) - T(\lambda^*)}{T(\lambda_{\text{worst}}) - T(\lambda^*)},$$

where $\lambda_{\text{worst-min}}, \lambda^*, \text{ and } \lambda_{\text{worst}}$ are the $\lambda$ of the local minimum of $\mathcal{L}$ with highest loss, the global minimum of $\mathcal{L}$, and the $\lambda$ maximizing $\mathcal{L}$, respectively. We use the same setup as above with $N = 50, D = 2$. The right of Fig. D-2 shows the results. We see that there are settings of $\theta^*$ and $\alpha$ for which selecting the worst minimum of $\mathcal{L}$ typically leads to a worse test loss than does the global minimum of $\mathcal{L}$. Interestingly, there are also settings of $\theta^*$ and $\alpha$ for which using the worst minimum of $\mathcal{L}$ leads to a better test loss than the global minimum of $\mathcal{L}$ (negative values in the right of Fig. D-2). We note that the absolute scale of severity under the test loss is smaller than in the other plots presented here, all of which measure the quality of a $\lambda$ using CV loss. This leaves open the possibility that while different local optima may have substantively different CV losses, their performance in practice – as measured by test loss – may be fairly similar.

### D.4 Empirical validation of Assumption 14

As noted in the main text, Assumption 14 can be interpreted as an assumption about the coherence of the $U$ matrix, a quantity commonly found in the compressed sensing literature [Candés and Recht].
In particular, Assumption 14 requires that the coherence of \( U \) decay with \( N \) sufficiently fast. Similar conditions have been studied in the literature for other matrices. For example, Lemma 2.2 of Candès and Recht [2009] shows that if \( U \) is drawn uniformly at random from the set of all orthonormal \( N \times D \) matrices, then \( \max_n \| u_{\max} \|^2 = O(\log(N)/N) \) with probability going to 1 as \( N \to \infty \). As \( \log(N)/N \) tends towards zero faster than \( N^{-p} \) for any \( 0 < p < 1 \), Lemma 2.2 of Candès and Recht [2009] proves that Assumption 14 holds with high probability if \( U \) is drawn uniformly from the set of orthogonal matrices.

However, the \( U \)'s of interest here have an additional constraint: that their columns be zero-mean. It is not clear how to adapt the proof of Candès and Recht [2009] to this situation. Instead, we offer empirical evidence that Assumption 14 holds in the case that \( \mathbf{U} \) is drawn uniformly from the set of orthogonal matrices. We describe how to generate such matrices in Appendix D.7. For fifty values of \( N \) from 2,500 to 20,500, we draw 750 orthonormal zero-mean orthonormal matrices \( \mathbf{U} \) from the uniform distribution. We plot the average \( \| u_{\max} \|^2 \) over these 750 replicas on a log scale versus \( N \) in Fig. D-3 (orange dots). For comparison, we plot the average \( \| u_{\max} \|^2 \) over 750 replicas when \( \mathbf{U} \) is drawn uniformly from the set of all orthonormal matrices (no zero-mean constraint) as the blue dots. The decay of \( \| u_{\max} \|^2 \) with and without the zero-mean constraint is essentially identical. Given this experiment, we argue that, although theoretically unjustified, Assumption 14 places only modest restrictions on the regression problems to which Theorem 5 applies.

### D.5 Proof of Theorem 5

We restate and prove Theorem 5 from Section 5.5.

**Theorem 5.** Take any series of regression problems \( \{ X^{(N)}, Y^{(N)} \}_{N=1}^{\infty} \) satisfying Assumptions 12 to 15. Let the covariate matrix of the \( N \)th regression problem have SVD \( X^{(N)} = U^{(N)} \text{ diag}(S^{(N)}) V^{(N)} \). There is a \( N_0 > 0 \) and neighborhood \( \Delta \) of \( \mathbf{1} \in \mathbb{R}^D \) such that if \( N \geq N_0 \) and the spectrum \( S^{(N)} \in \Delta \), then \( \mathcal{L} \) is quasiconvex.

**Proof.** We first prove the theorem for an exactly uniform spectrum, \( S = 1 \). To do so, we work with a sufficient condition for a one-dimensional function \( \mathcal{L} \) to be quasiconvex: for all \( \lambda \) such that \( \mathcal{L}'(\lambda) = 0 \), we have \( \mathcal{L}''(\lambda) > 0 \) [Boyd and Vandenberghe 2009, Chapter 3.4]. With this characterization of quasiconvexity in mind, our proof can be broken into two steps. We sketch each step here and refer to later lemmas for their proofs.

1. **Bound the region where \( \mathcal{L}' \) can be zero.** Write \( \mathcal{L}' \) as:

   \[
   \mathcal{L}'(\lambda) = \frac{1}{(\lambda + 1)^4} \sum_{n=1}^{N} \frac{(\lambda + 1)^3}{(\lambda + 1 - \| u_n \|^2)^3} \left( \xi_n \lambda^2 + \xi_n^2 \lambda + \xi_n \right).
   \]

   To find where this can be zero, we can ignore the \( 1/(\lambda + 1)^4 \). Then this is almost a quadratic in \( \lambda, \sum_n \xi_n \lambda^2 + \xi_n^2 \lambda + \xi_n \). Find the most positive root of this quadratic, \( \lambda_Q = O(1) \). Bound the deviations of \( \mathcal{L}' \) away from a quadratic, and bound how far these deviations can increase the zeros \( \mathcal{L}' \) beyond \( \lambda_Q \). **Result:** \( \mathcal{L}'(\lambda) \) can only be zero for \( \lambda \in [0, \lambda_Q + o(1)] \). We prove this step in Lemma 13 below.

2. **Show \( \mathcal{L}''(\lambda) > 0 \) for any \( \lambda \in [0, \lambda_Q + \Theta(1)] \) for which \( \mathcal{L}'(\lambda) = 0 \).** Essentially the same strategy; for any \( \lambda \) for which \( \mathcal{L}'(\lambda) = 0 \), write:

   \[
   \mathcal{L}''(\lambda) = \frac{1}{(\lambda + 1)^5} \sum_{n=1}^{N} \frac{(\lambda + 1)^4}{(\lambda + 1 - \| u_n \|^2)^4} \left( a_n \lambda^2 + b_n \lambda + c_n \right)
   \]

   This is a roughly a bowl-down quadratic with only one root bigger than zero; i.e. it is positive for \( \lambda \in [0, \lambda'_Q] \), where \( \lambda'_Q = \lambda_Q + \Theta(1) \) is the location of the quadratic’s rightmost root. Show
that the deviations away from quadratic imply that \( \mathcal{L}'' \) is positive for \( \lambda \in [0, \chi_q - o(1)] = [0, \lambda_Q + \Theta(1) - o(1)] \). We prove this step in Lemma 14 below.

With the theorem proved for an exactly uniform spectrum, we note that \( \mathcal{L}' \) and \( \mathcal{L}'' \) are continuous functions of the spectrum \( S \). As \( \mathcal{L}'' \) is strictly bounded away from zero on a region that contains \([0, \lambda_Q + \Theta(1)]\), by continuity in the singular values, there is a neighborhood \( \Delta \) of \( 1 \in \mathbb{R}^D \) such that if \( S \in \Delta \), \( \mathcal{L}''(\lambda) > 0 \) for all \( \lambda \) for which \( \mathcal{L}'(\lambda) = 0 \).

Before getting into the proofs of our main lemmas, we can first rearrange \( \mathcal{L}' \) into a convenient form. In Eq. (D.61) of Appendix D.9 we gave a convenient form of \( \mathcal{L} \) when the matrix of right singular vectors satisfies \( V = I_D \). Setting \( S = 1 \in \mathbb{R}^D \) in Eq. (D.61) and then taking the derivative with respect to \( \lambda \) gives:

\[
\mathcal{L}'(\lambda) = \sum_{n=1}^{N} \frac{2}{(1 + \lambda - \|u_n\|^2)^2} \left[ \frac{1}{1 + \lambda} u_n^T U^T Y - y_n \right] \tag{D.8}
\]

Let \( \tilde{e}_n \) be the scalars such that \( y_n = (x_n, \hat{\theta}) + \tilde{e}_n \). Letting \( \hat{E} \in \mathbb{R}^N \) be the vector with entries \( \tilde{e}_n \), we have \( U^T \hat{E} = 0 \), and so \( U^T Y = U^T U \text{diag}(1) \hat{\theta} = \hat{\theta} \). Plugging this into Eq. (D.8) we get:

\[
\mathcal{L}'(\lambda) = \sum_{n=1}^{N} \frac{2}{(1 + \lambda - \|u_n\|^2)^2} \left[ \frac{1}{1 + \lambda} (u_n, \hat{\theta}) - \|u_n\|^2 (\lambda + 1 - \|u_n\|^2)^2 \right]
\]

Finally, rearranging to group terms multiplying \( \lambda^2 \) and \( \lambda \), we get:

\[
\mathcal{L}'(\lambda) = \frac{2}{(1 + \lambda)^4} \sum_{n=1}^{N} \frac{(1 + \lambda)^3}{(1 + \lambda - \|u_n\|^2)^3} \left[ (1 - \|u_n\|^2)(u_n, \hat{\theta})^2 - \|u_n\|^2 \tilde{e}_n^2 + (1 - 2\|u_n\|^2)\tilde{e}_n (u_n, \hat{\theta}) - \lambda \right]
\]

We can write this more compactly

\[
\mathcal{L}'(\lambda) = \frac{2}{(1 + \lambda)^4} \sum_{n=1}^{N} \frac{(1 + \lambda)^3}{(1 + \lambda - \|u_n\|^2)^3} \left[ \xi_1 \lambda^2 + \xi_2 \lambda + \xi_3 \right], \tag{D.10}
\]

where the \( \xi_i \) are defined by matching up coefficients between Eq. (D.9) and Eq. (D.10). Now we can prove the two main Lemmas needed to prove Theorem 5.
Lemma 13. Take Assumptions 12 to 15. For a flat spectrum \( S = 1 \in \mathbb{R}^D \), there is some \( \lambda_Q \) that is \( O(1) \) such that \( \mathcal{L}'(\lambda) = 0 \) implies that \( \lambda \in [0, \lambda_Q + o(1)] \).

Proof. First, we can discard the \( 1/(1 + \lambda)^4 \) in front of Eq. (D.10) for the purposes of deciding where \( \mathcal{L}' = 0 \); let \( g(\lambda) = \mathcal{L}'(\lambda)(1 + \lambda)^4 \):

\[
g(\lambda) = \sum_{n=1}^{N} \frac{(1 + \lambda)^3}{(1 + \lambda - \|u_n\|^2)^3} (\xi_n \lambda^2 + \xi_n^2 \lambda + \xi_n^3),
\]

(D.11)

Notice that \( g \) is nearly a quadratic; in particular, if \( \|u_n\|^2 = 0 \), then \( g \) is a quadratic. The idea is to let \( \lambda_Q \) be the rightmost root of this quadratic; we then show that the perturbations away from this quadratic are small enough to imply that all zeros of \( g \) lie in \([0, \lambda_Q + o(1)]\).

Write \( \xi_i := \sum_n \xi_{ni} \). Then, via the quadratic formula, the roots of \( g_Q(\lambda) := \xi_1 \lambda^2 + \xi_2 \lambda + \xi_3 \) are

\[
\lambda = -\frac{-\xi_2 \pm \left[\xi_2^2 - 4\xi_1 \xi_3\right]^{1/2}}{2\xi_1}.
\]

(D.12)

We can apply the following facts from Lemma 16: \( \xi_1 \) and \( \xi_2 \) are \( \Theta(1) \) and \( \xi_3 \) is negative or \( o(1) \). We can conclude that the positive root of Eq. (D.12) is larger than the negative root and is \( O(1) \); call the positive root \( \lambda_Q \). Now we need to bound the deviations of \( g \) away from the quadratic \( g_Q \). Let \( \delta(\lambda) := g(\lambda) - g_Q(\lambda) \) be these deviations:

\[
\delta(\lambda) := \sum_{n=1}^{N} \left( \frac{\lambda + 1}{\lambda + 1 - \|u_n\|^2} \right)^3 (\xi_n \lambda^2 + \xi_n^2 \lambda + \xi_n^3) - 1
\]

(D.13)

Notice that our quadratic \( g_Q \) is convex, as \( \xi_1 > 0 \) by Lemma 16. Thus the way to move the roots of \( g \) further right than \( \lambda_Q \) is to have \( \delta(\lambda) \) be negative. We can lower bound \( \delta(\lambda) \geq \delta(0) \) by noting that \( \xi_1, \xi_2 > 0 \). Thus:

\[
\delta(\lambda) \geq \delta(0) = \sum_{n=1}^{N} \left( \frac{1}{(1 - \|u_n\|^2)^3} - 1 \right) \left( -\|u_n\|^2 \hat{e}_n^2 + (1 - \|u_n\|^2) \hat{e}_n (u_n, \hat{\theta}) \right).
\]

By Lemma 17 we have that \( \delta(0) = o(1) \).

As we know \( g(\lambda) = g_Q(\lambda) + \delta(\lambda) \geq g_Q(\lambda) + \delta(0) \), the final step of our proof is to find the rightmost \( \lambda \) for which \( g_Q(\lambda) = -\delta(0) \), as beyond such a \( \lambda \), \( g > 0 \). In fact, an upper bound on this \( \lambda \) will suffice. Using convexity with the fact that \( g_Q(\lambda_Q) = 0 \), we have that beyond \( \lambda = \lambda_Q + \delta(0)/g_Q'(\lambda_Q) \), \( g_Q(\lambda) \geq \delta(0) \), and thus \( g(\lambda) \geq 0 \). If we knew that \( g'(\lambda_Q) = 2\lambda_Q \xi_1 + \xi_2 \) were \( \Theta(1) \), we would be done, as:

\[
\lambda_Q = \frac{\delta(0)}{g_Q'(\lambda_Q)} = \lambda_Q + \frac{\delta(0)}{2\lambda_Q \xi_1 + \xi_2} = \lambda_Q + \frac{o(1)}{\Theta(1)} = \lambda_Q + o(1),
\]

(D.14)

To see that \( 2\lambda_Q \xi_1 + \xi_2 = \Theta(1) \), recall that we have \( \lambda_Q = O(1) \) and positive. And by Lemma 16, \( \xi_1 \) and \( \xi_2 \) are positive and \( \Theta(1) \). We conclude that \( 2\lambda_Q \xi_1 + \xi_2 = \Theta(1) \).

Lemma 14. Take Assumptions 12 to 15. Let \( \lambda_Q \) be as defined in Lemma 13 and assume the covariate matrix \( X \) has a flat spectrum \( S = 1 \in \mathbb{R}^D \). Then for any \( \lambda \in [0, \lambda_Q + \Theta(1)] \) such that \( \mathcal{L}'(\lambda) = 0 \), it holds that \( \mathcal{L}''(\lambda) > 0 \).

Proof. The strategy is similar to the proof of Lemma 13: we show that \( \mathcal{L}'' \) is nearly a quadratic, find the root of this quadratic, and then show that the location of this root can only change by \( o(1) \) due to the deviations away from quadratic.

163
First, we need to compute \( L'' \). Differentiating \( L' \) as given by Eq. (D.10) gives:

\[
L''(\lambda) = \frac{2}{1 + \lambda} \sum_{n=1}^{N} \left( -\frac{\xi_{n1}\lambda^3 + \xi_{n2}\lambda + \xi_{n3}}{(1 + \lambda)(1 + \lambda - \|u_n\|^2)^3} - \frac{3(\xi_{n1}\lambda^2 + \xi_{n2}\lambda + \xi_{n3})}{(\lambda + 1 - \|u_n\|^2)^4} \right) \]

\[+ \frac{2\xi_{n1}\lambda + \xi_{n2}}{(\lambda + 1 - \|u_n\|^2)^3} \]  

Now, the first term in this sum is exactly \( L'(\lambda) \). By the conditions of the lemma, we have that this term sums to zero. Using this fact and some algebra, we can see that \( L'' \) is also almost a quadratic:

\[
L''(\lambda) = \frac{2}{(1 + \lambda)^5} \sum_{n=1}^{N} \frac{(1 + \lambda)^4}{(1 + \lambda - \|u_n\|^2)^4} \left( -\xi_{n1}\lambda^2 + (1 - \|u_n\|^2)\xi_{n2}\lambda + ((1 - \|u_n\|^2)\xi_{n3} - 3\xi_{n3}) \right),
\]

where the \( \xi_{ni} \)'s are as defined in the proof of Lemma 13. As we are interested in the region where \( L'' > 0 \), we can neglect the \( 1/(1 + \lambda)^5 \) factor in front; define \( h(\lambda) := (1 + \lambda)^5 L''(\lambda) \). Now, define the following

\[
a_n := -\xi_{n1} = 2\|u_n\|^2\hat{\varepsilon}_n^2 - (1 - \|u_n\|^2)\langle u_n, \hat{\theta} \rangle^2
\]

\[
b_n := 2(1 - \|u_n\|^2)\xi_{n1} - 2\xi_{n2} = (2(1 - \|u_n\|^2)^2 - (1 - \|u_n\|^2))\langle u_n, \hat{\theta} \rangle^2 + (2(1 - \|u_n\|^2))^2\|u_n\|^2 - 2(2 - 3\|u_n\|^2)\|u_n\|^2 + 3\|u_n\|^2)\hat{\varepsilon}_n^2
\]

\[
c_n := (1 - \|u_n\|^2)\xi_{n2} - 3\xi_{n3} = (1 - \|u_n\|^2)^2\langle u_n, \hat{\theta} \rangle^2 + ((1 - \|u_n\|^2)(2 - 3\|u_n\|^2) - 3(1 - \|u_n\|^2))\|u_n\|^2
\]

\[
h_Q(\lambda) := \sum_{n=1}^{N} a_n\lambda^2 + b_n\lambda + c_n
\]

\[
\delta^{(2)}(\lambda) := \sum_{n=1}^{N} \left( \frac{1 + \lambda}{1 + \lambda - \|u_n\|^2} \right)^4 - 1 \left( a_n\lambda^2 + b_n\lambda + c_n \right).
\]

Note that \( h = h_Q + \delta^{(2)} \). Let \( a = \sum_n a_n \), and likewise for \( b, c \). Application of Proposition 27 and Assumptions 12 and 14 gives:

\[
c = \xi_3 - 3\xi_3 + o(1).
\]

In particular, noting that \( \xi_3 < 0 \) or is \( o(1) \) and \( \xi_3 \) is \( \Theta(1) \) and positive, (Lemma 16), we have \( c > 0 \) with \( |c| > |\xi_3| \) for large enough \( N \). Now, in general \( h_Q \) will have two roots:

\[
\lambda = \frac{-b \pm [b^2 - 4ac]^{1/2}}{2a} = \frac{b \pm [b^2 + 4\xi_1c]^{1/2}}{2\xi_1}.
\]

Now, as \( [b^2 + 4\xi_1c]^{1/2} > b \), only one of these roots is positive; call this root \( \lambda_Q \). We now want to
show that $\lambda'_{Q} - \lambda_{Q} = \Theta(1)$ and is positive. We have that:

$$\lambda'_{Q} - \lambda_{Q} = \frac{b + \xi_{2} + [b^{2} + 4\xi_{1}((1 - \|u_{n}\|^{2})\xi_{1} - 3\xi_{3})]^{1/2} - [\xi_{2} - 4\xi_{1}\xi_{3}]^{1/2}}{\xi_{1}}$$  \hspace{1cm} (D.26)$$

We know the denominator is positive and $\Theta(1)$ by Lemma \ref{lem:technicalLemma16} so we just need to show the numerator is $\Theta(1)$ and positive. Combining Assumption \ref{assumption:technicalAssumption14} with the fact that $b$ is positive or $o(1)$ (by Lemma \ref{lem:technicalLemma16}), we have that the numerator satisfies:

$$\geq \xi_{2} + [4\xi_{1}^{2} - 4\xi_{1}O(N^{-p}) - 12\xi_{3}\xi_{1}]^{1/2} - [\xi_{2} - 4\xi_{1}\xi_{3}]^{1/2}.$$  

Now, as $\xi_{1} = \Theta(1)$:

$$\geq \xi_{2} + [\Theta(1) - 12\xi_{3}\xi_{1}]^{1/2} - \xi_{2} - [4\xi_{1}\xi_{3}]^{1/2}.$$  

By Lemma \ref{lem:technicalLemma16} $\xi_{3}$ is either negative or $o(1)$. So the numerator satisfies:

$$\geq [\Theta(1) - 12\xi_{3}\xi_{1}]^{1/2} - [4\xi_{1}\xi_{3}]^{1/2},$$

which is $\Theta(1)$ and positive. Thus $\lambda'_{Q} - \lambda_{Q}$ is $\Theta(1)$ and is positive.

Finally, we need to lower bound $\delta^{(2)}(\lambda)$ on $[0, \lambda_{Q} + \Theta(1)]$. As $\lambda_{Q} = \Theta(1)$, Lemma \ref{lem:technicalLemma18} shows that $\delta^{(2)}(\lambda) = o(1)$ for all $\lambda \in [0, \lambda_{Q} + \Theta(1)]$. Thus for all $\lambda \in [0, \lambda_{Q} + O(1) - o(1)]$ for which $\mathcal{L}'(\lambda) = 0$, we have $h(\lambda) > 0$, and so $\mathcal{L}''(\lambda) > 0$. \hfill \Box

### D.5.1 Technical Lemmas

**Lemma 15.** Take real numbers $s_{1}, \ldots, s_{N}$ and $r_{1}, \ldots, r_{N}$, where $r_{n} \in [\ell, u]$ and $\sum_{n=1}^{N} s_{n} = 0$. Then:

$$\left| \sum_{n=1}^{N} r_{n}s_{n} \right| \leq \frac{u - \ell}{2} \sum_{n=1}^{N} |s_{n}|.$$  

**Proof.** Let $s^{+}_{n} := \max(0, s_{n})$ and $s^{-}_{n} := \max(0, -s_{n})$. Then the condition $\sum_{n} s_{n} = 0$ implies that $\sum_{n} s^{+}_{n} = \sum_{n} s^{-}_{n} = (1/2) \sum_{n=1}^{N} |s_{n}|$. So:

$$\left| \sum_{n=1}^{N} r_{n}s_{n} \right| \leq u \sum_{n=1}^{N} s^{+}_{n} - \ell \sum_{n=1}^{N} s^{-}_{n} = \frac{u - \ell}{2} \sum_{n=1}^{N} |s_{n}|.$$  

\hfill \Box

Now we state a useful consequence of our above assumptions:

**Proposition 27.** Take Assumptions \ref{assumption:technicalAssumption12} to \ref{assumption:technicalAssumption14} We have:

$$\sum_{n=1}^{N} (1 - \|u_{n}\|^{2})\hat{\lambda}_{n}(u_{n}, \hat{\theta}) = o(1)$$

**Proof.** Notice that $\sum_{n} \hat{\lambda}_{n}(u_{n}, \hat{\theta}) = \hat{E}^T U \hat{\theta} = 0$. So, we are trying to bound $\sum_{n} \|u_{n}\|\hat{\lambda}_{n}(u_{n}, \hat{\theta})$. Assumption \ref{assumption:technicalAssumption14} implies that $\|u_{n}\|^{2}$ is $O(N^{-p})$. As $\|u_{n}\|^{2}$ is lower bounded by zero, we can apply Lemma \ref{lem:technicalLemma15} to get the upper bound:

$$\left| \sum_{n=1}^{N} \|u_{n}\|\hat{\lambda}_{n}(u_{n}, \hat{\theta}) \right| \leq O(N^{-p}) \sum_{n=1}^{N} |\hat{\lambda}_{n}(u_{n}, \hat{\theta})|.$$  \hspace{1cm} (D.27)
By Cauchy-Schwarz, we can upper bound the sum as:

$$\sum_n |\hat{\varepsilon}_n \langle u_n, \hat{\theta} \rangle| \leq \left( \sum_{n=1}^{\infty} \hat{\varepsilon}_n^2 \right)^{1/2} \left( \sum_{n=1}^{\infty} \langle u_n, \hat{\theta} \rangle^2 \right)^{1/2}$$  \hspace{1cm} (D.28)

By Assumption 12 and the fact that $\sum_n \langle u_n, \hat{\theta} \rangle^2 = ||\hat{\theta}||^2$, we have overall:

$$\left| \sum_{n=1}^{\infty} ||u_n||^2 \hat{\varepsilon}_n \langle u_n, \hat{\theta} \rangle \right| \leq O(N^{-p})O(\sqrt{N}) ||\hat{\theta}||^2 = o(1),$$  \hspace{1cm} (D.29)

where the final equality holds because by assumption, $p > 1/2$ and $||\hat{\theta}|| = O(1)$.

Our next lemma concerns the quadratic coefficients that show up in $\mathcal{L}'$ and $\mathcal{L}''$.

**Lemma 16.** Recall the definitions of the coefficients of the quadratic parts of $\mathcal{L}'$ and $\mathcal{L}''$ from our proofs above:

- $\xi_{n1} := (1 - ||u_n||^2)\langle u_n, \hat{\theta} \rangle^2 - ||u_n||^2 \hat{\varepsilon}_n^2 + (1 - 2||u_n||^2)\hat{\varepsilon}_n \langle u_n, \hat{\theta} \rangle$
- $\xi_{n2} := (1 - ||u_n||^2)\langle u_n, \hat{\theta} \rangle^2 - 2||u_n||^2 \hat{\varepsilon}_n^2 + (2 - 3||u_n||^2)\hat{\varepsilon}_n \langle u_n, \hat{\theta} \rangle$
- $\xi_{n3} := -||u_n||^2 \hat{\varepsilon}_n^2 + (1 - ||u_n||^2)\hat{\varepsilon}_n \langle u_n, \hat{\theta} \rangle$
- $a_n := -\xi_{n1} = 2||u_n||^2 \hat{\varepsilon}_n^2 - (1 - ||u_n||^2)\langle u_n, \hat{\theta} \rangle^2$
- $b_n := 2(1 - ||u_n||^2)\xi_{n1} - 2\xi_{n2}$
- $c_n := (1 - ||u_n||^2)\xi_{n2} - 3\xi_{n3}$

Further, recall $\xi_1 := \sum_{n=1}^{N} \xi_{n1}$, and likewise for $\xi_2, \xi_3, a, b,$ and $c$. The following statements hold:

1. $\xi_1$ is positive and $\Theta(1)$.
2. $\xi_2$ is positive and $\Theta(1)$.
3. Either $b > 0$ or $b$ is positive and $o(1)$.
4. Either $\xi_3 < 0$ or $\xi_3$ is positive and $o(1)$.

**Proof.** We prove each item below.

1. $\xi_1$ is positive and $\Theta(1)$. Using Proposition 27 we have

$$\xi_1 = ||\hat{\theta}||^2 - \sum_{n=1}^{N} ||u_n||^2 (\langle u_n, \hat{\theta} \rangle^2 + \hat{\varepsilon}_n^2) + o(1)$$  \hspace{1cm} (D.30)

$$\geq ||\hat{\theta}||^2 - \sum_{n=1}^{N} ||u_n||^2 (\langle u_n, \hat{\theta} \rangle^2 + 2\hat{\varepsilon}_n^2) + o(1)$$  \hspace{1cm} (D.31)

$$= \Theta(1),$$  \hspace{1cm} (D.32)

where in the final equality we have used Assumption 15.

166
2. \( \xi_2 \) is positive and \( \Theta(1) \). The proof of this is identical to the proof that \( \xi_1 \) is positive and \( \Theta(1) \) but with different constants.

3. Either \( b > 0 \) or \( b \) is positive and \( o(1) \). From the definition of \( b_n \):

\[
b = \sum_{n=1}^{N} \|u_n\|^2 \xi_n^2 + \|u_n\|^2 \xi_n (u_n, \hat{\theta}) - (\|u_n\|^2 - (\|u_n\|^2)^2)\xi_n (u_n, \hat{\theta})^2
\]

By Assumption 14, any term with (\|u_n\|^2)^2 sums up to \( o(1) \). Thus:

\[
b = o(1) + \sum_{n=1}^{N} \|u_n\|^2 \xi_n^2 - \|u_n\|^2 \xi_n (u_n, \hat{\theta})^2
\]

Thus we have that \( b \) is either positive or is \( o(1) \).

4. Either \( \xi_3 < 0 \) or \( \xi_3 \) is positive and \( o(1) \). By Proposition 27

\[
\xi_3 = o(1) - \sum_{n=1}^{N} \|u_n\|^2 \xi_n^2.
\]

So \( \xi_3 \) is either \( o(1) \) and positive or is negative.

\[
\square
\]

**Lemma 17.** Take Assumptions 12 to 14. We have:

\[
\sum_{n=1}^{N} \left( \frac{1}{(1 - \|u_n\|^2)^3} - 1 \right) \left( -\|u_n\|^2 \xi_n^2 + (1 - \|u_n\|^2)\xi_n (u_n, \hat{\theta}) \right) = o(1).
\]

**Proof.** We first show that

\[
- \sum_{n=1}^{N} \left( \frac{1}{(1 - \|u_n\|^2)^3} - 1 \right) \|u_n\|^2 \xi_n^2 = o(1).
\]

First, note that as \( 0 \leq \|u_n\|^2 < 1 \), this quantity is strictly negative. So, it suffices to lower bound it by a quantity that is \( o(1) \). We apply the lower bound

\[
- \sum_{n=1}^{N} \left( \frac{1}{(1 - \|u_n\|^2)^3} - 1 \right) \|u_n\|^2 \xi_n^2 \geq - \left( \frac{1}{(1 - \|u_{\max}\|^2)^3} - 1 \right) \|u_{\max}\|^2 \sum_{n=1}^{N} \xi_n^2.
\]

By a Taylor expansion around \( \|u_{\max}\|^2 = 0 \), we have:

\[
= - \left( \|u_{\max}\|^2 + 6(\|u_{\max}\|^2)^2 + O((\|u_{\max}\|^2)^3) - \|u_{\max}\|^2 \right) \sum_{n=1}^{N} \xi_n^2.
\]

By Assumptions 12 and 14 this is equal to \( O(N^{-2p})O(N) = o(1) \).

Next we show that

\[
\sum_{n=1}^{N} \left( \frac{1}{(1 - \|u_n\|^2)^3} - 1 \right) (1 - \|u_n\|^2)\xi_n (u_n, \hat{\theta}) = o(1).
\]
To start, note that $\sum_n \hat{\varepsilon}_n(u_n, \hat{\theta}) = \hat{\varepsilon}^T U^T \hat{\theta} = 0$. We can then apply Lemma 15 to upper bound the absolute value of our quantity of interest:

$$\left| \sum_{n=1}^{N} \left( \frac{1}{1 - ||u_n||^2} - 1 \right) \left( 1 - ||u_n||^2 \right) \hat{\varepsilon}_n(u_n, \hat{\theta}) \right| \leq \left( \frac{1}{1 - ||u_{\max}||^2} - 1 \right) \left( 1 - ||u_{\max}||^2 \right) \sum_{n=1}^{N} |\hat{\varepsilon}_n(u_n, \hat{\theta})| \quad (D.37)$$

Now, by a Taylor expansion of the quantity outside the sum around $||u_{\max}||^2 = 0$

$$= (0 + 3||u_{max}||^2 + O(||u_{\max}||^2)^2)) \sum_{n=1}^{N} |\hat{\varepsilon}_n(u_n, \hat{\theta})|. \quad (D.38)$$

Applying Cauchy Schwarz along with Assumption 14

$$\leq O(N^{-p}) \left( \sum_{n=1}^{N} \hat{\varepsilon}_n^2 \right)^{1/2} \left( \sum_{n=1}^{N} (u_n^2\hat{\theta})^2 \right)^{1/2} \quad (D.39)$$

Applying Assumption 12 and then Assumption 13

$$= O(N^{-p})O(N^{1/2})||\hat{\theta}||^2 = o(1). \quad (D.40)$$

Lemma 18. Take Assumptions 13 to 14 and as in the proof of Lemma 14 define:

$$\delta(2)(\lambda) := \sum_{n=1}^{N} \left( \left( \frac{1 + \lambda}{1 + \lambda - ||u_n||^2} \right)^4 - 1 \right) (a_n\lambda^2 + b_n\lambda + c_n), \quad (D.41)$$

where $a_n, b_n, c_n$ are as defined in the proof of Lemma 14 Then, for $\lambda \in \{0, c\}$, where $c > 0$ is some constant in $N$, we have that $\delta(2)(\lambda) = o(1)$.

Proof. First, we have:

$$\left( \frac{1 + \lambda}{1 + \lambda - ||u_n||^2} \right)^4 - 1 \leq \left( \frac{1 + \lambda}{1 + \lambda - ||u_{\max}||^2} \right)^4 - 1 \quad (D.42)$$

$$= (0 + 4||u_{\max}||^2 + O(||u_{\max}||^2)^2) \quad (D.43)$$

$$= O(N^{-p}) \quad (D.44)$$

where the second equality holds by a Taylor expansion around $||u_{\max}||^2 = 0$, and the third equality holds by Assumption 14. Now, we can bound $\delta(2)(\lambda)$ as:

$$\delta(2)(\lambda) \leq O(N^{-p}) \sum_{n=1}^{N} |a_n\lambda^2 + b_n\lambda + c_n| \leq O(N^{-p}) \sum_{n=1}^{N} |a_n| + |b_n| + |c_n|, \quad (D.45)$$

where the second inequality is a result of $\lambda \leq c = O(1)$. We now bound the sums $\sum_n |a_n|, \sum_n |b_n|,$
and $\sum_n |c_n|$ to complete the proof.

\[
\sum_{n=1}^{N} |a_n| \leq \sum_{n=1}^{N} 2\|u_n\|^2 \hat{\varepsilon}_n^2 + (1 - \|u_n\|^2)\langle u_n, \hat{\theta}\rangle^2
\]

\[
= O(N^{1-p}) + (1 - O(N^{-p})) \|\hat{\theta}\|^2
\]

\[
= O(N^{1-p}),
\]

where the first line holds by the definition of $a_n$, the next by Assumptions 12 and 14, and the third by Assumption 13. We continue to bound:

\[
\sum_{n=1}^{N} |b_n|
\]

\[
\leq \sum_{n=1}^{N} 2(1 - \|u_n\|^2)^2 \langle u_n, \hat{\theta}\rangle^2 + 4\|u_n\|^2 \hat{\varepsilon}_n^2 + (2 + 4(\|u_n\|^2)^2 + 6\|u_n\|^2)\hat{\varepsilon}_n \langle u_n, \hat{\theta}\rangle
\]

\[
= (2 + O(N^{-p}) \|\hat{\theta}\|^2 + O(N^{1-p}) + (2 + O(N^{-p})) \left(\sum_{n=1}^{N} \hat{\varepsilon}_n^2\right)^{1/2} \left(\sum_{n=1}^{N} \langle u_n, \hat{\theta}\rangle^2\right)^{1/2}
\]

\[
= O(\sqrt{N}),
\]

where the first line is by definition of $b_n$, the second is by Assumptions 12 and 14 and the Cauchy-Schwarz inequality, and the third is by Assumptions 12 and 13. We continue by bounding:

\[
\sum_{n=1}^{N} |c_n|
\]

\[
\leq \sum_{n=1}^{N} (1 - \|u_n\|^2)^2 \langle u_n, \hat{\theta}\rangle^2 + 4\|u_n\|^2 \hat{\varepsilon}_n^2 + (2 + 3(\|u_n\|^2)^2 + 3\|u_n\|^2)\hat{\varepsilon}_n \langle u_n, \hat{\theta}\rangle
\]

\[
\leq \|\hat{\theta}\|^2 + O(N^{1-p}) + (2 + O(N^{-p})) \left(\sum_{n=1}^{N} \hat{\varepsilon}_n^2\right)^{1/2} \left(\sum_{n=1}^{N} \langle u_n, \hat{\theta}\rangle^2\right)^{1/2}
\]

\[
= O(\sqrt{N}),
\]

where the line-by-line reasoning is the same as that of our bound on $\sum_n |b_n|$ above. Plugging our bounds into Eq. (D.45), we get that for all $\lambda \in [0, c]$, we have that $\delta^{(2)}(\lambda) \leq O(N^{-p})O(\sqrt{N}) = o(1)$, as $p > 1/2$ by Assumption 14.

**D.6 Proof of Corollary 3**

We first state a theorem about the concentration of i.i.d. sub-Gaussian matrices. Let $s_D \leq \cdots \leq s_1$ be the singular values of $X$.

**Theorem 8** (Theorem 4.6.1 from [Vershynin 2018]). Suppose that the $x_n$ are independent sub-Gaussian isotropic random vectors with maximum sub-Gaussian constant $K$. Then for some constant $C > 0$ and any $t \geq 0$, the following holds with probability at least $1 - 2e^{-t^2}:

\[
\sqrt{N} - CK^2(\sqrt{D} + t) \leq s_D \leq s_1 \leq \sqrt{N} + CK^2(\sqrt{D} + t).
\]

We now restate and then prove Corollary 3.

**Corollary 3.** Take Assumptions 14 and 17. Assume we have a well-specified linear model for some
\( \theta^* \in \mathbb{R}^D \); that is, \( y_n = (x_n, \theta^*) + \epsilon_n \), where \( \epsilon_n \sim_{i.i.d.} \mathcal{N}(0, \sigma^2) \). If \( \sigma \) is sufficiently small, \( \hat{\theta} \) is consistent for \( \theta^* \), and the entries of the covariate matrix \( x_{nd} \) are i.i.d. sub-Gaussian random variables, then \( \mathcal{L} \) is quasiconvex with probability tending to 1 as \( N \to \infty \).

**Proof.** The idea is to show that our assumptions imply that Assumptions 12 and 13 hold, as well as that the spectrum of \( X \) becomes uniform with high probability. Our assumption that \( \hat{\theta} \) is consistent for \( \theta^* \) immediately implies that \( \| \hat{\theta} \| = O(1) \). Next, as discussed after Assumption 12, we can stack the \( \epsilon_n \) into a vector \( E \in \mathbb{R}^N \). We then have that \( \| E \|_2^2 \geq \| (I_N - UU^T)E \|_2^2 = \| E \|_2^2 = \sum \epsilon_n^2 \). As \( \| E \|_2^2 = O(N) \) with probability tending towards 1 as \( N \to \infty \), we have that Assumption 12 holds with probability tending towards 1 as \( N \to \infty \).

Now, by Theorem 8 with \( t = N^{1/3} \) (any \( t \) that goes to infinity with \( N \) but is \( o(\sqrt{N}) \) will work), we have that the singular values of \( X \) satisfy
\[
\sqrt{N} - o(\sqrt{N}) \leq s_D \leq s_1 \leq \sqrt{N} + o(\sqrt{N}).
\]
with probability at least \( 1 - o(1) \). By Proposition 9, the quasiconvexity of \( \mathcal{L} \) is invariant to a scaling of the singular values; thus we can divide all singular values by \( \sqrt{N} \) to get that all singular values lie in the interval \( [1 - o(1), 1 + o(1)] \) with probability going to 1. Thus, for any neighborhood \( \Delta \) of \( 1 \in \mathbb{R}^D \), the (normalized) singular values will eventually lie within \( \Delta \) with arbitrarily high probability. Thus all of the conditions of Theorem 7 are met, implying that \( \mathcal{L} \) will be quasiconvex with probability tending towards 1 as \( N \to \infty \).

\[ \square \]

### D.7 Generating zero-mean orthonormal matrices uniformly at random

In our experiments in Section 5.6, we draw \( N \times D \) orthonormal matrices \( U \) with zero-mean columns from the uniform distribution over such matrices. To do so, we generate vectors \( a_1, \ldots, a_D \in \mathbb{R}^D \) such that \( a_{nd} \sim_{i.i.d.} \mathcal{N}(0, 1) \). We then use the Gram-Schmidt process to orthogonalize the vectors \( \{1, a_1, \ldots, a_D\} \); the second through \( D + 1 \)th outputted vectors make up the columns of \( U \). Notice this procedure requires \( N < D \).

In our experiments surrounding Assumption 12, we need to generate vectors \( R \in \mathbb{R}^N \) such that \( U^TR = 0 \) uniformly over such \( R \)'s. To do so, we generate a vector \( a \sim \mathcal{N}(0, I_N) \). We then compute \( b = (I_N - UU^T)a \); setting \( R = b/\|b\| \) yields the result.

Why is this uniform over all vectors in the null space of \( U \)? Recall that \( a \) is an isotropic random vector. It is well-known that this implies that \( a/\|a\| \) is uniform over the unit sphere. As multiplication by \( I_N - UU^T \) is an orthogonal projection, we have that \( b = (I_N - UU^T)a \) is isotropic over the null space of \( U \). Thus \( b/\|b\| \) is uniform over the null-space of \( U \). The same reasoning shows that the use of the Gram-Schmidt algorithm to generate orthonormal zero-column-mean matrices \( U \) is uniform over such matrices – we start with isotropic random vectors, Gram Schmidt applies orthogonal projections to each and then normalizes the results.

### D.8 Replicating experiments with error bars

For each of the experiments in Fig. 5-3 (our experiments about the size of the neighborhood \( \Delta \) and \( U \)'s violating Assumption 14), we repeat the experiment five times to understand the random variability in each experiment. Fig. 5-4 shows the result. All plots are created exactly as in Fig. 5-3 except each dot is now an average over all five trials. Error bars are equal to two times the standard deviation across these five trials. We see that our conclusions from Fig. 5-3 still hold. In some cases, the error bars are so small that they are barely visible on the scale of these plots.
Figure D-4: Replication of Fig. 5-3 with error bars; we repeat the original caption here for reading convenience. (Upper left): We generate many datasets and plot the fraction that are not quasiconvex, varying $N$ and the distance of the spectrum from uniformity ($\|S - \mathbf{1}\|_1$). (Upper right): We generate two sets (orange, blue) of left-singular vector matrices $U$. In the blue case, we check that the maximum of $\log \|u_{\text{max}}\|_2^2$ across all $U$ for a particular $N$ decreases roughly linearly on a log-log plot (i.e. the blue set satisfies Assumption 14). In the orange case, we check that the minimum of $\log \|u_{\text{max}}\|_2^2$ across all $U$ for a particular $N$ is roughly constant (i.e. the orange set does not satisfy Assumption 14). (Lower): For all the $U$ matrices from the upper right plot, we generate many datasets and plot the fraction that are not quasiconvex.
D.9 Implementation details of experiments

D.9.1 Efficiently computing $\mathcal{L}(\lambda)$

The LOOCV loss is given in Eq. (5.2) of the main text as:

$$L(\lambda) := \sum_{n=1}^{N} \left( \langle x_n, \hat{\theta}^{\backslash n}(\lambda) \rangle - y_n \right)^2,$$  \hspace{1cm} (D.59)

While this seems to require solving $N$ regression problems to obtain $\hat{\theta}^{\backslash n}(\lambda)$ for $n = 1, \ldots, N$, there is a well-known explicit formula for $L$ that makes use of the Sherman-Morrison formula:

$$L(\lambda) = \sum_{n=1}^{N} \frac{1}{(1 - Q_n)^2} \left( \langle x_n, \hat{\theta} \rangle - y_n \right)^2,$$  \hspace{1cm} (D.60)

where $Q_n := x_n^T (X^T X + \lambda I_D)^{-1} x_n$.

Using the results of Proposition 9, we know that the matrix of right singular vectors $V$ does not matter for the quasiconvexity of $L$. So, in all of our experiments, we set $V = I_D$, which further simplifies Eq. (D.60). In particular, letting $u_n$ be the $n$th row of $U$, we have $x_n = u_n \text{diag}(S)$ and $Q_n = u_n^T \text{diag}(S^2/(S^2 + \lambda)) u_n$. Then:

$$L(\lambda) = \sum_{n=1}^{N} \frac{1}{(1 - u_n^T \text{diag} \left( \frac{S^2}{S^2 + \lambda} \right) u_n)^2} \left( u_n^T \text{diag} \left( \frac{S^2}{S^2 + \lambda} \right) U^T y - y_n \right)^2.$$  \hspace{1cm} (D.61)

We are given $U$ and $S$ in all of our experiments, so this formula is much faster to compute than Eq. (D.60), as it requires no matrix inversions.

D.9.2 Numerically checking for quasiconvexity

To check for quasiconvexity in our experiments, we evaluate $L$ on a dense, regularly spaced grid $\lambda_1, \ldots, \lambda_T$. In this appendix, we describe how, given $L(\lambda_1), \ldots, L(\lambda_T)$, we check whether $L$ is quasiconvex. In short, we numerically check whether $L'$ ever switches from positive to negative (the condition for a local maximum); note that this must occur inbetween any two local minima, so we do not have to count the number of local minima. To approximate the sign of $L'$, we use $s_i := \text{Sign}(L(\lambda_{i+1}) - L(\lambda_i))$ for $i = 1, \ldots, T - 1$. We then report quasiconvexity if there exists any $i$ for which $s_i = 1$ and $s_{i+1} = -1$; that is, if our approximation to the derivative changes from positive to negative.

Note there are two ways in which this procedure can fail. The first is that our grid of $\lambda_i$’s may be insufficiently dense to capture non-quasiconvex behavior; however, in practice, we have never observed this to be an issue, as we have never found increasing the density of the grid of $\lambda$’s to reveal extra local minima. The second issue is that non-quasiconvex behavior may occur beyond the maximal $\lambda_T$ we specify. We again do not have an exact way of preventing this in practice. However, in all of our experiments, we specify a $\lambda_T$ that is orders of magnitude larger than the maximal singular value of $X$. At this point, we expect $L(\lambda)$ to be an essentially flat function; thus, any missed local minima are likely of effectively the same value as $L(\infty)$. In any case, if either failure mode were occurring in our of our experiments, fixing it would only make the (already concerning) non-quasiconvexity in our experiments look more severe.
Appendix E

Appendix for: Measuring the sensitivity of Gaussian processes to kernel choice (Chapter 6)

E.1 Details of spectral density constraints

Here, we give the details of how we optimize over spectral densities to produce a stationary kernel as summarized in Algorithm 5. Our goal is to optimize over the set of stationary kernels. It is not immediately clear how to enforce this constraint; however, Bochner’s theorem [Rasmussen and Williams, 2006, Thm. 4.1] tells us that every stationary kernel \( k(x, x') = k(\tau) \), where \( \tau = x - x' \) has a positive finite spectral measure \( \mu \) on \( \mathbb{R}^D \) such that:

\[
    k(\tau) = \int_{\mathbb{R}^D} e^{2\pi i \tau^T \omega} \mu(\omega) d\omega.
\]  

(E.1)

A common assumption in the literature on kernel discovery [Wilson and Adams, 2013, Benton et al., 2019, Wilson et al., 2016] is to assume that \( \mu \) has a density \( S \) with respect to the Lebesgue measure; that is, we can write:

\[
    k(\tau) = \int_{\mathbb{R}^D} e^{2\pi i \tau^T \omega} S(\omega) d\omega.
\]  

(E.2)

These works have shown that the class of stationary kernel with spectral densities is a rich, flexible class of kernels. We thus focus on the class of stationary kernels with spectral densities as this allows us to transform the problem of optimizing over stationary kernels into the problem of optimizing over positive real valued functions. In all of our examples optimizing over spectral densities, we have \( D = 1 \). We thus assume \( D = 1 \) in the rest of our development here. In this case, it must be that \( S \) is symmetric around the origin to obtain a real-valued \( k \). So, we can simply Eq. (E.2) further as:

\[
    k(\tau) = \int_0^\infty \cos(2\pi \tau \omega) S(\omega) d\omega.
\]  

(E.3)

Optimizing over positive functions \( S \) on the positive real line seems at least somewhat more tractable than optimizing over stationary positive-definite functions \( k(\tau) \). However, this is still an infinite dimensional optimization problem. To recover a finite dimensional optimization problem, we follow Benton et al. [2019] and choose a grid \( \omega_1, \ldots, \omega_G \). We can then optimize over the finite values \( S(\omega_1), \ldots, S(\omega_G) \) and use the trapezoidal rule to approximate the integral in Eq. (E.3). Benton et al. [2019] find that \( G = 100 \) gives reasonable performance in their experiments; we find the same in ours, and fix \( G = 100 \) throughout. Benton et al. [2019] recommend setting \( \omega_g = 2\pi g/(8\tau_{max}) \), where \( \tau_{max} \) is the maximum spacing between datapoints. We find this to sometimes give inaccurate
results in the sense that using the trapezoidal rule / an exact formula to compute the density of \( k_0, S(\omega_1), \ldots, S(\omega_\ell) \) and then using the trapezoidal rule to recover the gram matrix \( k_0(X, X) \) gives an inaccurate approximation to \( k_0(X, X) \). This is problematic in our case, as it would imply \( k_0(X, X) \) is not in the constraint set for small \( \varepsilon \). Instead, we recommend setting our \( \omega_y \)'s as a uniform grid from \( \omega_1 = 0 \) up to an \( \omega_\ell \) such that \( S_0(\omega_\ell) \) is equal to the floating point epsilon (\( 10^{-15} \) in our experiments); some manual experimentation will be required to implement this rule.

As we are only interested in kernels nearby \( k_0 \), we will have to put some kind of constraint on \( k_1 \)'s spectral density, \( S_1(\omega_1), \ldots, S_1(\omega_\ell) \). We use a simple \( \varepsilon \)-ball given by:

\[
\max \left( 0, (1 - \varepsilon)S_0(\omega_g) \right) \leq S_1(\omega_g) \leq (1 + \varepsilon)S_0(\omega_g), \quad g = 1, \ldots, G, \tag{E.4}
\]

Because our posterior functional of interest \( F^* \) is a differentiable function of the kernel matrix, we can compute gradients of \( F^* \) with respect to our discretized spectral density. Rather than manually work out the derivatives of the trapezoidal rule combined with \( F^* \), we use the automatic differentiation package \texttt{jax} \cite{Bradbury2018}. Given a gradient of \( F^* \), we take a step in the direction of the gradient and then project the current iterate onto our constraint set in Eq. \( \text{(E.4)} \) by clipping the resulting spectral density.

### E.2 Additional details of synthetic-data experiment

We generated the \( x \)-component of the synthetic data by first drawing 25 uniform random numbers in \( [0, 5] \). To investigate what happens when interpolating in a region of dense training data, we then draw 10 uniform random numbers in \( [1.9, 2.1] \). \( \text{recall the interpolation point is} \ x^* = 2.0 \). The extrapolation point \( x^* = 5.29 \) lies 0.5 to the right of the largest \( x \) value drawn. The \( y \)-component is defined to be

\[
y_i = \frac{x_i^2}{2} + \cos(\pi x_i) + \epsilon_i,
\]

where \( \epsilon_i \sim \mathcal{N}(0, 0.01) \).

We chose the \( \varepsilon \) grid for extrapolation \( (x^* = 5.29) \) to be 15 evenly-spaced values between 0.2 and 0.8. The grid for interpolation \( (x^* = 2.00) \) is 15 evenly-spaced values on the log grid between \( 10^{0.1} \) and \( 10^3 \).

To discretize the spectral density, we follow Appendix \[E.1\] in using 100 frequencies evenly-spaced from 0 to 2. To optimize over nearby spectral densities, we also perform constrained gradient descent with randomized initializations in the sense of Appendix \[E.1\]. For extrapolation \( (x^* = 5.29) \), using 25 random seeds, we find non-robustness. For interpolation \( (x^* = 2.00) \), even with 40 random seeds, we do not find non-robustness.

Fig. \[E.5\] compares the distance between \( k_1 \) and \( k_0 \) to the distances between \( k_0 \) and \( k^{(r)} \), where \( k^{(r)} \) is a bootstrapped version of \( k_0 \).

Computing is done using a computing cluster, which has xeon-p8 computing cores. We request 7 nodes, each using 15 cores to run parallel experiments across both \( \epsilon \) and the random seed for initialization. Total wall-clock time comes to roughly 5 minutes.

### E.3 Additional details for the heart rate example

Here, we give additional details for our heart rate modeling example from Section \[6.3\] According to \cite{Reyna2019}, the data was collected under the approval of appropriate institutional review boards, and personal identifiers were removed. Following \cite{Colopy2010}, we first take the log
Figure E-1: Sensitivity of heart rate analysis in Appendix E.3 for an example where we do not find non-robustness. (Top-left): Heart rate data; notice the data is trending downwards at the end of the time series. (Top-right): Prior draws from our original kernel $k_0$ from Eq. (E.6). (Bottom-left): Prior draws from our decision-changing kernel $k_1$ that achieves $F^* = L$, noise matched by color to the draws from $k_0$. (Bottom-right): Comparison of the difference between $k_0$ and $k_1$ (red line) to posterior hyperparameter uncertainty (histogram).

Transform of our heart rate observations $y_n$. We then zero-mean the observations ($\sum_{n=1}^{N} y_n = 0$) and set them to have unit variance ($\sum_{n=1}^{N} y_n^2 = 1$). The kernel used by Colopy et al. [2016] to model the resulting data log-scaled standardized data is a Matérn 5/2 kernel plus a squared exponential kernel:

$$k_0(x, x') = h_1^2 \left( 1 + \frac{\sqrt{5}|x - x'|}{\lambda_1} + \frac{5|x - x'|^2}{3\lambda_1} \right) \exp \left[ -\frac{\sqrt{5}|x - x'|}{3\lambda_1} \right] + h_2^2 \exp \left[ -\frac{|x - x'|^2}{2\lambda_2^2} \right],$$

where $h_1, h_2, \lambda_1, \lambda_2 > 0$ are kernel hyperparameters, which we set via MMLE. While all inferences are done on the zero-mean, unit-variance log-scaled data, all of our plots and discussion are given in the untransformed (i.e. raw bpm) scale for ease of interpretability.

In the main text, we showed an example where our workflow in Algorithm 4 discovered non-robustness in predicting whether a patient’s heart rate would be likely to be above 130 BPM or not 1.5 hours in the future. We noted that there was some evidence in the data supporting this finding: the patient’s heart rate was trending upward towards the end of the observed data, so we might expect that small changes to the prior could result in significant posterior mass being placed on high heart rates. To demonstrate that we do not always find GP analyses non-robust to the choice of the prior, we give an example here where we do not find non-robustness. For our example, we use a different patient from the Computing in Cardiology challenge Reyna et al. [2019], Goldberger et al. [2000]. The heart rate for this patient is plotted in Fig. E-1; notice that their heart rate is trending down at the end of the observed data.

As in Section 6.3 we use the constraint set and objective specified by Algorithm 5 (i.e. we constrain ourselves to stationary kernels with spectral densities close to the density of $k_0$). Following Algorithm 4, we solve Eq. (6.1) to find $k_1(\varepsilon^*)$ such that $F^*(k_1(\varepsilon^*)) = L$. We then assess whether
the recovered \( k_1(\varepsilon^*) \) is qualitatively interchangeable with \( k_0 \). We plot noise-matched prior draws from \( k_0 \) and \( k_1(\varepsilon^*) \) in Fig. E-1. We see that \( k_1(\varepsilon^*) \) has obvious qualitative deviations from \( k_0 \); the functions drawn from \( k_1(\varepsilon^*) \) have noticeably larger variance (count the number of times the functions from \( k_1(\varepsilon^*) \) pass 130 bpm). Additionally, we see in Fig. E-1 that the 2-Wasserstein distance between \( k_0(X, X) \) and \( k_1(X, X) \) is much larger than the typical deviations around \( k_0 \) due to hyperparameter uncertainty. We conclude that \( k_0 \) and \( k_1(\varepsilon^*) \) are not qualitatively interchangeable. Thus, we say that we do not find non-robustness in the sense of Definition 6. Again, this conclusion is fairly sensible: at the final observation, the patient’s heart rate is below 80 BPM and is trending downwards. It thus seems reasonable that it would take a somewhat unusual prior to predict that the patient’s heart rate would suddenly spike to 130.

The two heart rate experiments in were run on a laptop with a six-core i7-9750H processor. The experiments took roughly five minutes each to complete.

## E.4 Additional details for CO\(_2\) experiment

Here, we give additional details on the CO\(_2\) experiment from Section 6.4. Our dataset is a series of monthly CO\(_2\) levels taken from Mauna Loa in Hawaii between 1958 and 2021 [Keeling et al., 2005], we download our data from [this link](http://cdiac.esd.ornl.gov/ftp/trends/co2/maunaloa.co2). Rasmussen and Williams [2006, Section 5.4.3] predict future CO\(_2\) levels using a GP. Their kernel is the sum of four terms:

\[
k_0(x_1, x_2) = \theta_1^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\theta_2^2}\right) + \theta_3^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\theta_4^2} - \frac{2\sin^2(\pi(x_1 - x_2))}{\theta_5^2}\right) + \theta_6^2 \left(1 + \frac{(x_1 - x_2)^2}{2\theta_7^2}\right)^{-\theta_8} + \theta_9^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\theta_{10}^2}\right),
\]

where the \( \theta_i \) comprise the kernel hyperparameters (in addition to the noise variance \( \sigma^2 \)). The different components of this kernel encode different pieces of prior knowledge. The two squared exponentials encode long-term trends and small-scale noise, respectively. The rational quadratic kernel (Eq. (E.8)) encodes small seasonal variability in CO\(_2\) levels between different years. The periodic kernel captures the periodic trend in CO\(_2\) levels, which peak in the summer and reach their minimum in the winter. This periodic is multiplied by a squared exponential to allow deviations away from exact periodicity.

Similar to Rasmussen and Williams [2006, Section 5.4.3], we first transform the training data by making the CO\(_2\) levels have zero mean. To set the GP hyperparameters, we find that the hyperparameters values reported in Rasmussen and Williams [2006, Section 5.4.3] are close, but not exactly, the MMLE solution on our data set (the gradient of the marginal log-likelihood has an entry substantially different from zero under the parameters from Rasmussen and Williams [2006]). We set hyperparameters by 10 random restarts of MMLE, where the solution iterates are initialized at the values reported in Rasmussen and Williams [2006, Section 5.4.3]. The fitted values are \( \theta_1 = 68.58, \theta_2 = 69.09, \theta_3 = 2.55, \theta_4 = 87.60, \theta_5 = 1.44, \theta_6 = 0.66, \theta_7 = 1.18, \theta_8 = 0.74, \theta_9 = 0.18, \theta_{10} = 0.13, \theta_{11} = 0.19 \). They are, for the most part, within 5% of the values reported in Rasmussen and Williams [2006, Section 5.4.3].

When Rasmussen and Williams [2006] ran their analysis, only data up to 2003 were available. As it turns out, their analysis significantly underestimates current CO\(_2\) levels. In particular, they

\[\text{http://cdiac.esd.ornl.gov/ftp/trends/co2/maunaloa.co2}\]

was no longer responsive, for instance.
Figure E-2: Sensitivity analysis of Mauna Loa. Each plot shows noise matched samples from a zero mean Gaussian process with original and perturbed kernel functions. These plots provide a zoomed in view of the prior samples shown in Fig. 6-5. We note that draws from $k_1(\varepsilon^*)$ are in-phase with those of $k_0$ (i.e. $k_1(\varepsilon^*)$ captures the seasonal maxima and minima of CO$_2$ just as well as $k_0$ does). Overall, there is high agreement between functions sampled from the two GPs.
fail to predict the fact that CO$_2$ levels hit 415 ppm for the first time in human history in 2019; in fact, the maximum of the predicted CO$_2$ levels in 2019 is over three posterior standard deviations away from 415 ppm. We ask if a qualitatively interchangeable kernel could have changed this result. Ideally, we would set $F^*$ to be the max of all posterior predictions in 2019. However, this is not a smooth function of the kernel. So we instead let $F^*$ to be the smooth max of the posterior means of all the test points in 2019, $\{\mu(x_t)\}_{t=1}^T$. The smooth-max we use is a scaled log-sum-exp, with a scale $\alpha > 0$:

$$F^*(k) = \log \left( \sum_{t=1}^T e^{\alpha \mu(x_t)} \right) / \alpha.$$ 

Larger values of $\alpha$ provide a better approximation to the actual max function but may cause numerical difficulties; we choose $\alpha = 10$ as it seems to provide a reasonable approximation to the max function without introducing numerical problems. While we optimize using this approximation to the max, our experiments show that the recovered $k_1(\varepsilon^*)$ have an exact max prediction in 2019 of 415 ppm.

$k_0$ is stationary, so we could search for alternative stationary kernels using our spectral density framework from Section 6.2.1 (Algorithm 5). However, there is good reason to think we might want to consider non-stationary prior beliefs. Developments in technology and/or global policy could have a large impact on CO$_2$ levels. Thus, we might encode past / expected future changes in technology and policy into our prior beliefs, making our prior beliefs non-stationary.

Thus, we use the input warping approach from Section 6.2.1 (Algorithm 6). However, we do not input warp the entirety of $k_0$. As we know CO$_2$ data has a regular periodicity, we leave the periodic component of the kernel, $\exp[-2\sin^2(\pi(x_1 - x_2))/\theta_2^2]$ unwarped. In preliminary experiments, we input warped the entirety of $k_0$; the resulting prior draws sometimes had minima in the summer and maxima in the winter, a clear violation of our prior knowledge about CO$_2$ levels. We input warp all other parts of $k_0$ using a a two hidden layer fully connected network, with 50 units and ReLU nonlinearities to parameterize $h$. Finally, to ensure the optimal $k_1(\varepsilon)$ is finite, we use $\ell(k; F^*, L) = (F^*(k) - L)^2$ in Algorithm 6 which guarantees that our objective is bounded below.

We plot noise matched prior draws for $k_0$ and $k_1(\varepsilon^*)$ in Fig. E-2. The samples from $k_1(\varepsilon^*)$ appropriately line up with the expected maxima and minima of CO$_2$ levels (to see this, note that the draws from $k_3(\varepsilon^*)$ are in-phase with those from $k_0$, which correctly captures the seasonal maxima and minima). The deviations between the noise-matched samples do not seem significant, so we say that $k_1(\varepsilon^*)$ and $k_0$ are qualitatively interchangeable. Further, we find that the distance between $k_1(\varepsilon^*)$ and $k_0$ is smaller than what we might expect to arise from sampling uncertainty about $k_0$’s hyperparameters (see Fig. E-8 in Appendix E.6). We therefore conclude that the prediction of CO$_2$ levels under $k_0$ is non-robust to the choice of the kernel in the sense of Definition 6.

The Mauna Loa experiments were run on a laptop with a 2.3 GHz 8-Core Intel Core i9, with 64 GB of RAM. The experiment (which optimizes from five random initialization) took about 15 minutes to run, with each seed taking about 3 mins.

### E.5 More details on MNIST experiments

We use the publicly available neural-tangents [Novak et al., 2020] package for constructing the kernels in our MNIST experiments. We follow the experimental setup of Lee et al. [2018] where the authors use a Gaussian process with a kernel corresponding to a 20 layer, infinitely wide, fully connected, deep neural network with ReLU non-linearities. They place zero mean Gaussian priors over the weights, $\mathcal{N}(0, \sigma_w^2)$, and biases, $\mathcal{N}(0, \sigma_b^2)$, and set the hyper-parameters $\sigma_w^2 = 1.45$ and $\sigma_b^2 = 0.28$ via a grid search over parameters to maximize held-out predictive performance. Lee et al. [2018] use a GP with $C = 10$ outputs (classes). They pre-process one-hot encoded output vectors to have zero mean, i.e. $y_{ic} = 0.9$ if $i$ is the correct class for the $ith$ training point, and $y_{ic} = -0.1$ for all incorrect classes; input images are flattened and an overall mean is subtracted from every image. Test prediction is made by selecting a class corresponding to the GP output with mean closest to 0.9. The resulting GP trained on one thousand images from the MNIST training set and evaluated
Figure E-3: Additional MNIST experiments. Here we visualize the training and test set performances along the hyperparameter grid used for assessing qualitative interchangeability. The train and test accuracies exhibit high performance and low variability across the grid.

Figure E-4: Example test images from Section 6.5. \( x^* \) (upper) and their warps \( g(x^*) \) (lower) and predicted class labels (above and below).

on the MNIST test set achieves an accuracy of 92.79%.

In our experiments we assess the robustness of their kernel. The 28 x 28 MNIST images require a warping function \( g : \mathbb{R}^{784} \rightarrow \mathbb{R}^{784} \). We use a fully connected multi-layer perceptron with one 784 unit hidden layer, 784 input, and 784 output units with ReLU non-linearities to parametrize \( g \). Let \( c_0 \) be the prediction under the original kernel at a target test image \( x^* \). We define \( c_1 := |c_0 - 1| \) and create a “fake” output \( y^* \) with \( y^*_c = 0.9 \) and \( y^*_c = -0.1 \) for \( c \neq 1 \). We find parameters of \( g \) by minimizing the objective in Algorithm 6 plugging in

\[
\ell(k; F^*, L) = -\frac{1}{C} \sum_{c=1}^{10} \log p(y^*_c | X, x^*, y),
\]  

(E.10)
i.e. the negative log-likelihood of the “fake” output at a particular test image \( x^* \) under the perturbed kernel; \( X \) and \( Y \) are the train inputs and outputs. As we discussed in the main text, directly optimizing the posterior quantity of interest \( F^* = |\mu_{c_0}(x^*) - 0.9| - |\mu_{c_1}(x^*) - 0.9| \) produces unrealistic outputs, e.g. \( \mu_{c_0}(x^*) \ll -0.1 \). Such predictions would look obviously suspicious to a user, so we would say that our supposed malicious actor has not achieved their goal in this case. Instead, we optimize the surrogate loss in Eq. (E.10). With this surrogate loss we are able to find kernel perturbations yielding benign-looking outputs and achieving the goal of the malicious actor to change the prediction at \( x^* \) to \( c_1 \), i.e. \( \mu_{c_1}(x^*) \approx 0.9 \) and \( \mu_{c}(x^*) \approx -0.1 \) for all \( c \neq 1 \). In this case, we feel that a user would not be able to identify these predictions as obviously wrong, and so we say that the malicious actor has achieved their goal of changing the predictions of \( k_0 \) without detection.

**Hyperparameter sensitivity.** To quantify variability in the Gram matrices arising from
hyperparameter uncertainty, we vary $\sigma^2_w$ over 30 uniformly spaced points between 1.4 and 1.5, and
$\sigma^2_b$ over 30 uniformly spaced points between 0.23 and 0.33. This defines a grid that is ten times
smaller than the grid [Lee et al. 2018] optimize their hyperparameters over when searching for $\hat{\theta}$.
Thus we have no reason to pick $\theta$ as the true optimum over any of our grid points; that is, our grid
points provide a natural (conservative) notion of uncertainty in $\hat{\theta}$. Fig. E-3 shows that over the 900
possible hyperparameter combinations the train and test accuracies remain high and exhibit low
variability.

The experiment took approximately 55 minutes to run for a single test image. We ran the
computations for the 1000 test images in parallel on a compute cluster with Intel Xeon E5-2667 v2,
3.30GHz cores, requesting one core each time.

E.6  Additional Gram matrix comparisons

To assess qualitative interchangeability, we compared the 2-Wasserstein distance between the Gram
matrices $k_0(X, X)$ and $k_1(\varepsilon^*)(X, X)$ to the 2-Wasserstein distance between $k_0(X, X)$ and $k^{(r)}(X, X)$,
where $k^{(r)}$ had the same functional form as $k_0$ but different hyperparameters. One might be con-
cerned that our findings of (non-)qualitative interchangeability might be sensitive to this choice of
the 2-Wasserstein distance. We show here that in all of our experiments, we would have reached
the same decision about the qualitative interchangeability of $k_1(\varepsilon^*)$ and $k_0$ if we instead used any
of: Frobenius norm, nuclear norm, spectral norm, infinity norm, or symmetrized Kullback-Leibler
distance. Fig. E-5 and Fig. E-6 show our results for our synthetic extrapolation and interpolation
examples. Fig. E-7 shows our results for our heart rate example from Section 6.3 and Fig. E-10
shows our results from our additional heart rate example in Appendix E.3. Fig. E-8 shows our
results from our CO$_2$ modeling example in Section 6.4. Finally, Fig. E-9 shows our results from our
MNIST example in Section 6.5.
Figure E-5: Extra hyperparameter uncertainty histograms for our synthetic extrapolation example in Section 6.2.3 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.
Figure E-6: Extra hyperparameter uncertainty histograms for our synthetic interpolation example in Section 6.2.3 in which we find do not find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.

Figure E-7: Extra hyperparameter uncertainty histograms for our heart rate experiment in Section 6.3 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.
Figure E-8: Extra hyperparameter uncertainty histograms for our Mauna Loa experiment in Section 6.4 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.
Figure E-9: Extra hyperparameter uncertainty histograms for our MNIST experiment in Section 6.5 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances. Note, distances are plotted on the log-scale.

Figure E-10: Extra hyperparameter uncertainty histograms for our additional heart rate experiment in Appendix E.3 in which we find non-robustness. We compare the difference between $k_0$ and $k_1(\varepsilon^*)$ (red) to bootstrapped hyperparameter uncertainty (gray) in several distances.


