Interpretations of Machine Learning and Their Application to Therapeutic Design

by

Brandon M. Carter

S.B., Massachusetts Institute of Technology (2017)
M.Eng., 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

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Brandon M. Carter
Department of Electrical Engineering and Computer Science
May 19, 2023

Certified by: David K. Gifford
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by: Tommi S. Jaakkola
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Interpretations of Machine Learning and Their Application to Therapeutic Design

by

Brandon M. Carter

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

We introduce a framework for interpreting black-box machine learning (ML) models, discover *overinterpretation* as a failure mode of deep neural networks, and discuss how ML methods can be applied for therapeutic design, including a pan-variant COVID-19 vaccine. While ML models are widely deployed and often attain superior accuracy compared to traditional approaches, deep learning models are functionally complex and difficult to interpret, limiting their adoption in high-stakes environments. In addition to safer deployment, model interpretation also aids scientific discovery, where validated ML models trained on experimental data can be used to uncover biological mechanisms or to design therapeutics through biologically faithful objective functions, such as vaccine population coverage.

For interpretation of black-box ML models, we introduce the Sufficient Input Subsets (SIS) method that is model-agnostic, faithful to underlying functions, and conceptually straightforward. We demonstrate ML model interpretation with SIS in natural language, computer vision, and computational biological settings. Using the SIS framework, we discover overinterpretation, a novel failure mode of deep neural networks that can hinder generalizability in real-world environments. We posit that overinterpretation results from degenerate signals present in training datasets. Next, using ML models that have been calibrated with experimental immunogenicity data, we develop a flexible framework for the computational design of robust peptide vaccines. Our framework optimizes the n-times coverage of each individual in the population to activate broader T cell immune responses, account for differences in peptide immunogenicity across individuals, and reduce the chance of vaccine escape by mutations. Using this framework, we design vaccines for SARS-CoV-2 that have superior population coverage to published baselines and are conserved across variants of concern. We validate this approach *in vivo* through a COVID-19 animal challenge study of our vaccine. This thesis demonstrates distinct ways model interpretation enables ML methods to be faithfully deployed in biological settings.

Thesis Supervisor: David K. Gifford
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Tommi S. Jaakkola
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

There are many people I need to thank who played a role during my time at MIT — without them, this journey would not have been possible.

First, I am grateful to my fantastic advisors and thesis committee members: David Gifford, Tommi Jaakkola, and Michael Birnbaum. Dave helped me understand how to do science that has real-world impact, taught me to think creatively about scientific problems that span different domains, and I am thankful for Dave's thoughtful guidance and support throughout my research endeavours. I appreciate Dave's high standard for scientific rigor and always leave meetings with Dave encouraged about our next experiments. Tommi sparked my interest in machine learning when I took his NLP class in Fall 2015 as an undergraduate, and I have appreciated his support and feedback on my work throughout my time at MIT. Michael helped me understand how the methods we developed actually affect living systems through his deep understanding of immunology. I am thankful for such a wonderful thesis committee that has provided broad viewpoints into my research and shaped my scientific outlook.

The Gifford Lab provided a fun and collaborative environment for doing science. I am fortunate to have worked closely with Jonas Mueller, Siddhartha Jain, and Ge Liu, who were outstanding collaborators and mentors. I am also grateful to have been surrounded by talented and fun peers that made my time in the Gifford Lab so enjoyable: Alexander Dimitrakakis, Benjamin Holmes, Bianca Lepe, Grace Yeo, Haoyang Zeng, Hyunjin Park, Jennifer Hammelman, Jonathan Krog, Konstantin Krismer, Sachit Saksena, Yuchun Guo, and Zheng Dai. Much of my work would not have been possible without the support of our collaborators in the Birnbaum Lab, Novartis Institutes for BioMedical Research, The University of Texas Medical Branch, Acuitas Therapeutics, the Ragon Institute, Boston University, Tufts University, and the National Institutes of Health. I also thank Linda Lynch, Sally Lee, and Janet Fischer for their guidance and support throughout my Ph.D.

I am thankful to have worked closely with Arvind Satyanarayan. Arvind is a great mentor and collaborator and taught me to think about user-centered design of ML

methods. I was also fortunate to have had the opportunity to work at Google Brain with Jamie Smith, Max Bileschi, and Lucy Colwell, and my time at Google was both educational and rewarding. I also thank Lucas Liebenwein for a fruitful collaboration on pruned models.

Throughout my time at MIT, I was surrounded by great friends and colleagues and have many fond memories. In particular, I thank Ken Leidal, who is a close friend and was a frequent course project partner and offered helpful feedback on my work.

Finally, thank you to my parents, grandparents, and American and Canadian families. Without their endless support, none of this would have been possible. To Angie, thank you for your unconditional support, for your feedback and advice that have immensely improved my research, for enlightening scientific discussions of machine learning and interpretability, and for making me smile and laugh every day.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent progress in machine learning (ML) has led to rapid deployment of machine learning models in many domains. ML methods are employed to understand natural language, speech, and images, to make medical diagnoses and propose treatment interventions, and as tools in basic scientific research. Machine learning algorithms are capable of extracting the potentially complex patterns present in observed data to make predictions about data that have not yet been observed. Deep learning methods (e.g., neural networks) are particularly expressive and flexible models because they act hierarchically — latent representations of the data are expressed in terms of simpler primitive representations that are also learned by the models. As a consequence, deep learning models often achieve state-of-the-art performance over traditional learning algorithms and are widely deployed. A common supervised ML pipeline involves training a model and measuring accuracy on held-out test data to evaluate whether the model can adequately generalize to unseen data. If the model is found to have high test accuracy, it may be deployed. However, to permit their functional expressivity, neural networks typically contain a large number of parameters representing complex and nonlinear functions. Thus, deep learning models are generally poorly understood and have earned reputations as opaque "black-boxes." Performance gains often come at the cost of interpretability, a trade-off that is concerning as we demonstrate that models may attain high test accuracy but still rely upon spurious features present in the training data.

Interpretability plays an important role in the development and deployment of ML models and in building trust among stakeholders in the ML process (Hong et al., 2020). The lack of interpretability of ML models is concerning because it means practitioners cannot understand *how* their models make decisions. In many settings, it is crucial that decisions made by ML models can be explained. For example, consider ML models used by physicians to screen patients for a disease. Given patient data, a black-box model simply outputs Disease or No Disease, but without also providing an explanation for this decision, the model cannot be trusted. Indeed, machine learning methods have been known to make erroneous predictions in such settings (Zech et al., 2018; Patel, 2017). Without transparency in how models make decisions, humans may lack trust in model outputs. Interpretability is necessary to ensure that a prediction was not made as a result of spurious correlations or biases present in training data.

In addition to making complex model decision-making more transparent to users prior to or during model deployment, interpretability has other use cases in ML and in scientific applications as illustrated in Figure 1-1. Prior to real world deployment, model interpretation can be a useful tool for debugging, where an understanding of *why* a model makes certain misclassifications may permit users to revise the training procedure or model architecture design to improve predictive accuracy. Interpretation can also expose idiosyncratic behaviors of individual models and be used as part of a model selection pipeline: which model to select from a series of models trained for the same task. An understanding of the features learned by each model may help expose how the models differ to decide which is best suited for adoption in a particular system. Additionally, interpretability can be adopted in scientific settings, where accurate predictive models trained on experimental data can be interpreted to extract the scientific mechanisms underlying the observed data (Greener et al., 2022; Azodi et al., 2020). The resulting principles can serve as hypotheses in follow-up experiments that can generate new data or be used to refine the training procedure to produce models whose decision-making criteria are more aligned with the ground-truth mechanisms. Finally, interpretation can be a tool to validate ML models prior to integration of the model into a downstream task, such as therapeutic design. In this setting, the

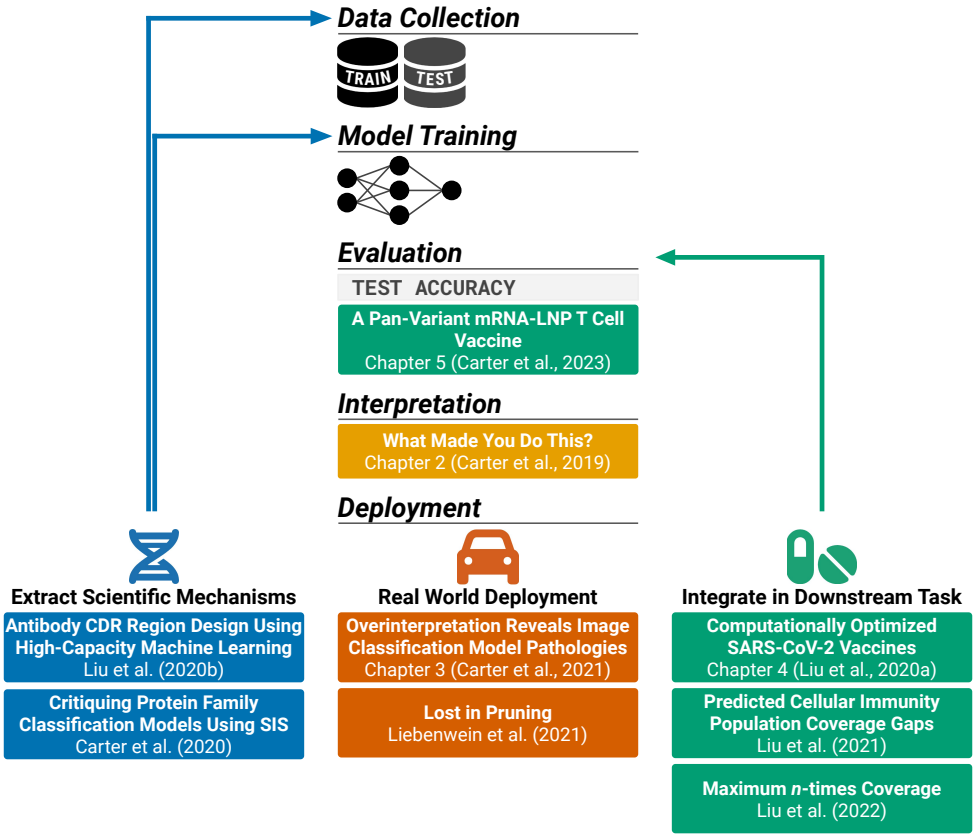


Figure 1-1: Overview of the roles of model interpretation in a machine learning pipeline. Model interpretation provides a mechanism to validate ML model behavior prior to or during real world deployment, extract underlying mechanisms behind the training data (e.g., in scientific settings), or permit integration of ML models into downstream tasks (e.g., therapeutic design). This thesis develops methods for interpretation of ML models and the application of validated ML models, including to the downstream task of vaccine design.

ML model can be adopted an one input to a task-specific objective function, whose output (e.g., a vaccine) can be validated through a separate experiment, and the results potentially used to evaluate or refine the ML model or objective function. In this thesis, we develop methods for interpretation of black-box ML models and the application of validated ML models to vaccine design.

One application requiring accurate and robust machine learning models is the computational design of therapeutic molecules, an advance enabled by the curation of biological experimental data into publicly available repositories. For instance, high-capacity machine learning models can be trained on large repositories of bio-

logical data and used to design new antibodies with superior properties (Liu et al., 2020b) or as part of a pipeline to design personalized cancer vaccines (Hu et al., 2018). Deep learning methods offer advantages over traditional bioinformatic approaches as a result of their expressive power and ability to efficiently learn complex patterns that may be unknown to scientists *a priori*. Machine learning models can also guide or replace expensive or time-consuming laboratory experiments to identify novel molecules (Stokes et al., 2020). Given this role, robust behavior of machine learning models and an understanding of their decision-making are crucial in biological settings.

In the computational vaccine design setting, deep learning models can now accurately predict the binding between peptide ligands and major histocompatibility complex (MHC) molecules, a prerequisite for eliciting a cellular immune response against the peptide (Reynisson et al., 2020a; Zeng and Gifford, 2019). These models are trained and validated on experimental data in curated repositories, including the Immune Epitope Database (IEDB) (Vita et al., 2019), and can be further calibrated with clinical data. The trained and validated models can then be used to rapidly predict the presentation of many peptides by a diverse range of MHC alleles *in silico*.

However, there is a need for vaccine design frameworks that can use the ML predictions to do principled selection among potentially thousands of candidate peptides to design effective peptide vaccines with broad population coverage. In particular, a flexible and robust vaccine design framework should consider the distribution of MHC haplotypes across human populations to more accurately estimate population coverage, provide redundancy for a stronger immune response in each individual, and select optimal peptide candidates based upon criteria including mutation rate across pathogenic variants. The framework should also ensure that each individual is covered by multiple peptide-HLA "hits" to allow for variation in T cell repertoire and peptide immunogenicity across individuals. We define a *peptide-HLA hit* as a pair of peptide and HLA allele where the peptide is predicted to be displayed by the HLA and immunogenic in the individual. If these criteria are not met, a vaccine design may have limited population coverage and vaccine protection may be more easily escaped

by pathogenic drift.

This thesis contributes methods to multiple stages of the ML pipeline to demonstrate applications of model interpretation as described by Figure 1-1. We contribute and validate frameworks for interpretation of black-box ML models, including models trained on biological data, and for principled design of vaccines with broad population coverage that incorporates ML model predictions into a vaccine design objective. Our framework designs have objectives that faithfully reflect their underlying systems, are flexible, and are conceptually straightforward to facilitate adoption by users who may not be ML experts.

1.1 Thesis Outline

We first introduce the Sufficient Input Subsets (SIS) method for machine learning model interpretability (**Chapter 2**). The SIS approach is model-agnostic, faithful to underlying model functions, and the outputs can be easily understood by humans. SIS can be used to interpret model decisions locally on individual inputs, or rationales can be clustered across many inputs to gain global insights into general principles governing the model's behavior.

Second, we introduce Batched Gradient SIS to efficiently scale the SIS framework to high-dimensional data and discover a novel failure mode of deep learning models — *overinterpretation* (**Chapter 3**). We find that models trained on the popular ImageNet benchmark often rely solely on border pixels rather than salient objects. While these features may lead to fragility of models in real world deployment, we find that they are in fact valid statistical patterns in the benchmark datasets that suffice to attain high test accuracy, and thus future training datasets may need to be carefully curated to eliminate such artifacts.

Third, we introduce a flexible and robust framework for peptide vaccine design with n-times coverage (**Chapter 4**). We develop computational tools to predict the population coverage of peptide vaccine designs based upon human HLA haplotype frequencies. Our framework requires predicted HLA display of multiple peptides

per individual to strengthen the expected immune response by engaging a diverse set of T cell clonotypes and to permit differences in peptide immunogenicity across individuals. This redundancy also reduces the chance of the pathogen mutating and escaping immune protection. Our method permits incorporation of experimental immunogenicity data and design of vaccines against different target populations of interest. Vaccine peptide candidates can stem from specific proteins of interest and be filtered by multiple criteria, including mutation rate across pathogenic variants. We introduce a combinatorial optimization procedure (OptiVax) to design peptide vaccines against the n-times coverage objective. We find our vaccine designs are superior in population coverage to published baseline vaccine designs for COVID-19.

Finally, we validate the OptiVax framework through an animal challenge study with a COVID-19 variant of concern (**Chapter 5**). We show that an mRNA-LNP vaccine consisting of short, conserved epitopes derived from SARS-CoV-2 prevented mortality in HLA transgenic mice challenged with the SARS-CoV-2 Beta variant of concern. Our vaccine activated a robust cellular immune response and expanded both $CD8^+$ and $CD4^+$ T cells. The vaccine caused significantly more $CD8^+$ and $CD4^+$ T lymphocytic infiltration of the lungs compared to Pfizer-BioNTech and PBS controls. Our results demonstrate *in vivo* that the OptiVax framework can be used to design effective pan-variant peptide vaccines and highlight the importance of n-times coverage.

1.2 Prior Publications

The work presented in this thesis was done in collaboration with a number of coauthors who were key in shaping the research. The material of Chapters 2, 3, 4, and 5 has appeared in the following joint publications:

Carter, B., Mueller, J., Jain, S., and Gifford, D. (2019). What made you do this? Understanding black-box decisions with sufficient input subsets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 567–576

Carter, B., Jain, S., Mueller, J. W., and Gifford, D. (2021). Overinterpretation reveals image classification model pathologies. *Advances in Neural Information Processing Systems*, 34:15395–15407

Liu, G., Carter, B., Bricken, T., Jain, S., Viard, M., Carrington, M., and Gifford, D. K. (2020a). Computationally optimized SARS-CoV-2 MHC class I and II vaccine formulations predicted to target human haplotype distributions. *Cell Systems*, 11(2):131–144

Liu, G., Carter, B., and Gifford, D. K. (2021). Predicted cellular immunity population coverage gaps for SARS-CoV-2 subunit vaccines and their augmentation by compact peptide sets. *Cell Systems*, 12(1):102–107

Liu, G., Dimitrakakis, A., Carter, B., and Gifford, D. (2022). Maximum n-times coverage for vaccine design. In *International Conference on Learning Representations*

Carter, B., Huang, P., Liu, G., Liang, Y., Lin, P. J. C., Peng, B.-H., McKay, L. G. A., Dimitrakakis, A., Hsu, J., Tat, V., Saenkham-Huntsinger, P., Chen, J., Kaseke, C., Gaiha, G. D., Xu, Q., Griffiths, A., Tam, Y. K., Tseng, C.-T. K., and Gifford, D. K. (2023). A pan-variant mRNA-LNP T cell vaccine protects HLA transgenic mice from mortality after infection with SARS-CoV-2 Beta. *Frontiers in Immunology*, 14:1135815

Chapter 2

Sufficient Input Subsets for Interpretability

The rise of neural networks and nonparametric methods in machine learning (ML) has driven enormous improvements in prediction capabilities, while simultaneously earning the field a reputation of producing complex black-box models. Vital applications, which could benefit most from improved prediction, are often deemed too sensitive for opaque learning systems. Such applications include the use of ML models to reject loan applicants (Sirignano et al., 2018), deny defendants' bail (Kleinberg et al., 2018), or diagnose disease (Gulshan et al., 2016). For ML models to be used in these systems, it is imperative that the decisions they make can be interpretably rationalized. Interpretability is also crucial in scientific applications, where it is hoped that underlying principles may be extracted from accurate predictive models (Doshi-Velez and Kim, 2017; Lipton, 2016).

One simple explanation for *why* a particular black-box decision is reached may be obtained via a sparse subset of the input features whose values form the basis for the model's decision – a *rationale*. For text (or image) data, a rationale might consist of a subset of positions in the document (or image) together with the words (or pixel-values) occurring at these positions (examples shown in Figures 2-3 and 2-17). Here, we consider rationales that do not attempt to summarize the (potentially complex) operations carried out within a black-box model, but instead point to the relevant

features used by the model to arrive at a decision on that particular input. This property ensures that the interpretations remain faithful to any arbitrary model. Additionally, we desire that rationales are sparse, which facilitates interpretability when inputs are high-dimensional (Lei et al., 2016).

Here, we develop a local explanation framework to produce rationales for a learned model that has been trained to map inputs $\mathbf{x} \in \mathcal{X}$ via some arbitrary learned function $f : \mathcal{X} \to \mathbb{R}$. Unlike many other interpretability techniques, our approach is not restricted to vector-valued data and does not require gradients or differentiability of f. Rather, each input example is solely presumed to have a set of indexable features $\mathbf{x} = [x_1, \ldots, x_p]$, where each $x_i \in \mathbb{R}^d$ for $i \in [p] = \{1, \ldots, p\}$. Our method can be applied to interpret decisions made on inputs $\mathbf{x}$ whose features are unordered (set-valued inputs) or for which the number of features p can vary (e.g., variable-length sequences). A rationale corresponds to a sparse subset of these indices $S \subseteq [p]$ together with the specific values of the features in this subset.

To understand why a certain decision was made for a given input $\mathbf{x}$, we propose a particular rationale called a *sufficient input subset* (SIS). Each SIS consists of a minimal input pattern present in $\mathbf{x}$ that alone suffices for f to produce the same decision, even if provided no other information about the rest of $\mathbf{x}$. Presuming the decision is based on $f(\mathbf{x})$ exceeding some prespecified threshold $\tau \in \mathbb{R}$, we seek to find a minimal-cardinality subset S of the input features such that $f(\mathbf{x}_S) \geq \tau$. Throughout, we use $\mathbf{x}_S \in \mathcal{X}$ to denote a modified input example in which all information about the values of features outside subset S has been masked, with features in S remaining at their original values. Thus, each SIS characterizes a particular standalone input pattern that drives the model toward the decision, providing sufficient justification for this choice from the model's perspective, even without any information about the values of the other features in $\mathbf{x}$.

In classification settings, f might represent the predicted probability of class C where we decide to assign the input to class C if $f(\mathbf{x}) \geq \tau$, where τ can be chosen based on precision/recall considerations. Each SIS in such an application corresponds to a small input pattern that on its own is highly indicative of class C, according to the

model. Note that by suitably defining f and τ with respect to the predictor outputs, any particular decision for input $\mathbf{x}$ can be precisely identified with the occurrence of $f(\mathbf{x}) \geq \tau$, such that greater values of f are associated with greater confidence in the decision.

For a given input $\mathbf{x}$ that satisfies $f(\mathbf{x}) \geq \tau$, this work presents a simple method to find a complete collection of sufficient input subsets, each satisfying $f(\mathbf{x}_S) \geq \tau$, such that there exists no additional SIS outside of this collection. Each SIS may be understood as a disjoint piece of evidence that would lead the model to reach the same decision, and why this decision was reached for $\mathbf{x}$ can be unequivocally attributed to the SIS-collection. Furthermore, global insight on the general principles underlying the model's decision-making process may be gleaned by clustering the types of SIS extracted across different data points (Section 2.7). Such insights allow us to compare models based not only on their accuracy, but also on human-determined relevance of the concepts they target. Our method's simplicity facilitates its utilization by non-experts who may know little about the models they wish to interrogate.

Code for the experiments in this chapter is available at: `https://github.com/b-carter/SufficientInputSubsets`.

2.1 Related Work

Certain neural network variants such as attention mechanisms (Sha and Wang, 2017) and the generator-encoder of Lei et al. (2016) have been proposed as powerful yet human-interpretable learners. Other interpretability efforts have tailored decompositions to certain convolutional/recurrent networks (Murdoch et al., 2018; Olah et al., 2017, 2018; Strobelt et al., 2018), but these approaches are model-specific and only suited for ML experts. Many applications necessitate a model outside of these families, either to ensure supreme accuracy, or if training is done separately with access restricted to a black-box API (Caruana et al., 2015; Tramer et al., 2016). Thus, much recent research aims to address the critical need for methods which enable non-ML experts to rationalize the predictions of any type of model. One general approach

entails fitting a separate explanation model to the outputs of f over the same training data, for example a feature-selector (Li et al., 2017) or surrogate decision tree (Frosst and Hinton, 2017; Zhang et al., 2018; Wu et al., 2017). However, such a strategy may not be generalizable to out of sample examples (which are crucial for understanding how f would behave in certain counterfactual settings).

An alternative model-agnostic approach to interpretability produces local explanations of f for a particular input $\mathbf{x}$. Local explanation often relies on attribution methods that quantify how much each feature influences the output of f at $\mathbf{x}$. Examples include LIME, which locally approximates f (Ribeiro et al., 2016), saliency maps based on gradients of f (Baehrens et al., 2010; Simonyan et al., 2014), Layer-wise Relevance Propagation (Bach et al., 2015), as well as the discrete DeepLIFT approach (Shrikumar et al., 2017) and its continuous variant, Integrated Gradients (IG) (Sundararajan et al., 2017), developed to ensure attributions reflect the cumulative difference in f at $\mathbf{x}$ vs. a reference input. A separate class of input-signal-based explanation techniques such as DeConvNet (Zeiler and Fergus, 2014), Guided Backprop (Springenberg et al., 2015), and PatternNet (Kindermans et al., 2018) employ gradients of f in order to identify input patterns that cause f to output large values. However, many gradient-based saliency methods have been deemed unreliable, depending not only on the learned function f, but also on its specific architectural implementation and how inputs are scaled (Kindermans et al., 2019, 2018). More like our approach, recent techniques from Dabkowski and Gal (2017); Kim et al. (2018); Chen et al. (2018) also aim to identify input patterns that best explain certain decisions, but additionally require either a predefined set of such patterns or an auxiliary neural network trained to identify them. The principle of Shapley values, which are approximated by existing feature attribution methods (Lundberg and Lee, 2017), asserts that to assess the effect of a feature, its presence/absence should be considered in the context of all other possible feature subsets. In contrast, our backward selection approach only evaluates the effect of a feature in the context of the remaining not-yet-masked features, as our focus is identifying feature subsets that meet the SIS sufficiency criterion (Section 2.2), as opposed to feature attribution.

In comparison with the aforementioned methods, our SIS approach is: conceptually simple, entirely faithful to any type of model, and requires neither gradients of f nor auxiliary training of the underlying model f or a surrogate explanation model. Also related to our subset-selection methodology are the ideas of Li et al. (2017) and Fong and Vedaldi (2017), which for a particular input seek a small feature subset whose omission causes a substantial drop in f such that a different decision would be reached. However, this objective can produce adversarial artifacts that are hard to interpret. In contrast, we focus on identifying small subsets of input features whose values suffice to ensure f outputs significantly positive predictions, even in the absence of any other information about the rest of the input. While the techniques of Li et al. (2017) and Fong and Vedaldi (2017) produce rationales that remain highly dependent on the rest of the input outside of the selected feature subset, each rationale identified by our SIS approach is independently considered by f as an entirely sufficient justification for a particular decision in the absence of other information.

More broadly, our approach aims to interpret model behavior on a particular instance by identifying the input features that provide the basis for the model's decision on that input. Another class of model interpretation methods instead explains a model's behavior through the influence of instances in training data and measures the effects of deleting entire training instances on the model's predictions (Cook, 1977). The approach by Koh and Liang (2017) adopts influence functions that measure the sensitivity of the model to an infinitesimal upweighting of each training instance. In contrast, our SIS method perturbs the features *within* a particular instance to rationalize the model's prediction on that input. However, one parallel between our approach and the approach of deletion diagnostics and influence functions is the strategy of deleting a feature (or training instance) to estimate its importance. Given SIS can be applied to any black-box function, one can frame computing influential instances as applying SIS to a function that takes as input a dataset (where "features" specify which training instances are to be included) and outputs a model performance metric of interest, and SIS would return a minimal set of input features (training instances) that suffice for the model to attain performance at a specified threshold.

2.2 Sufficient Input Subsets Method

Our approach to rationalizing why a particular black-box decision is reached only applies to input examples $\mathbf{x} \in \mathcal{X}$ that meet the decision criterion $f(\mathbf{x}) \geq \tau$. For such an input $\mathbf{x}$, we aim to identify a SIS-collection of disjoint feature subsets $S_1, \ldots, S_K \subseteq [p]$ that satisfy the following criteria:

(1) $f(\mathbf{x}_{S_k}) \geq \tau$ for each $k = 1, \ldots, K$

(2) There exists no feature subset $S' \subset S_k$ for some $k = 1, \ldots, K$ such that $f(\mathbf{x}_{S'}) \geq \tau$

(3) $f(\mathbf{x}_R) < \tau$ for $R = [p] \setminus \bigcup_{k=1}^{K} S_k$ (the remaining features outside of the SIS-collection)

Criterion (1) ensures that for any SIS S_k, the values of the features in this subset alone suffice to justify the decision in the absence of any information regarding the values of the other features. To ensure information that is not vital to reach the decision is not included within the SIS, criterion (2) encourages each SIS to contain a minimal number of features, which facilitates interpretability. Finally, we require that our SIS-collection satisfies a notion of completeness via criterion (3), which states that the same decision is no longer reached for the input after the entire SIS-collection has been masked. This implies the remaining feature values of the input no longer contain sufficient evidence for the same decision. Figures 2-4 and 2-17 show SIS-collections found in text and image inputs, respectively.

Recall that $\mathbf{x}_S \in \mathcal{X}$ denotes a modified input in which the information about the values of features outside subset S is considered to be missing. We construct $\mathbf{x}_S$ as new input whose values on features in S are identical to those in the original $\mathbf{x}$, and whose remaining features $x_i \in [p] \setminus S$ are each replaced by a special mask $z_i \in \mathbb{R}^{d_i}$ used to represent a missing observation. While certain models are specially adapted to handle inputs with missing observations (Smola et al., 2005), this is generally not the case. To ensure our approach is applicable to all models, we draw inspiration

from data imputation techniques which are a common way to represent missing data (Rubin, 1976).

Two popular strategies include hot-deck imputation, in which unobserved values are sampled from their marginal feature distribution, and mean imputation, in which each z_i is simply fixed to the average value of feature i in the data. Note that for a linear model, these two strategies are expected to produce an identical change in prediction $f(\mathbf{x}) - f(\mathbf{x}_S)$. We find in practice that the change in predictions resulting from either masking strategy is roughly equivalent even for nonlinear models such as neural networks (Appendix A.1.1, Figure A-1). In this work, we favor the mean-imputation approach over sampling-based imputation, which would be computationally expensive and nondeterministic (undesirable for facilitating interpretability). One may also view $\mathbf{z}$ as the *baseline* input value used by feature attribution methods (Sundararajan et al., 2017; Shrikumar et al., 2017), a value which should not lead to particularly noteworthy decisions. Since our interests primarily lie in rationalizing atypical decisions, the average input arising from mean imputation serves as a suitable baseline. Zeros have also been used to mask image and categorical data (Li et al., 2017), but empirically, this mask appears undesirably more informative than the mean (predictions more affected by zero-masking).

For an arbitrarily complex function f over inputs with many features p, the combinatorial search to identify sets which satisfy objectives (1)–(3) is computationally infeasible. To find a SIS-collection in practice, we employ a straightforward backward selection strategy, which is here applied separately on an example-by-example basis (unlike standard statistical tools which perform backward selection globally to find a fixed set of features for all inputs). The **SIScollection** algorithm details our straightforward procedure to identify disjoint SIS subsets that satisfy (1)–(3) approximately for an input $\mathbf{x} \in \mathcal{X}$ where $f(\mathbf{x}) \geq \tau$. Disjointness of the sufficient input subsets in a SIS-collection is crucial to ensure computational tractability and that the number of SIS per example does not grow huge and hard to interpret. For a more rigorous evaluation of the properties of SIS, see Section 3.1 of Carter et al. (2019).

Our overall strategy is to find a SIS subset S_k (via **BackSelect** and **FindSIS**),

SIScollection(f, $\mathbf{x}$, τ)

1 $S = [p]$
2 **for** $k = 1, 2, \ldots$ **do**
3 $\quad R = \textbf{BackSelect}(f, \mathbf{x}, S)$
4 $\quad S_k = \textbf{FindSIS}(f, \mathbf{x}, \tau, R)$
5 $\quad S \leftarrow S \setminus S_k$
6 $\quad$ **if** $f(\mathbf{x}_S) < \tau$: **return** S_1,...,S_{k-1}

BackSelect(f, $\mathbf{x}$, S)

1 $R =$ empty stack
2 **while** $S \neq \varnothing$ **do**
3 $\quad i^* = \text{argmax}_{i \in S} f(\mathbf{x}_{S \setminus \{i\}})$
4 $\quad$ Update $S \leftarrow S \setminus \{i^*\}$
5 $\quad$ Push i^* onto top of R
6 **return** R

FindSIS(f, $\mathbf{x}$, τ, R)

1 $S = \varnothing$
2 **while** $f(\mathbf{x}_S) < \tau$ **do**
3 $\quad$ Pop i from top of R
4 $\quad$ Update $S \leftarrow S \cup \{i\}$
5 **if** $f(\mathbf{x}_S) \geq \tau$: **return** S
6 **else**: **return** *None*

mask it out, and then repeat these two steps restricting each search for the next SIS solely to features disjoint from the currently found SIS-collection $S_1, \ldots, S_k$, until the decision of interest is no longer supported by the remaining feature values. In the **BackSelect** procedure, $S \subset [p]$ denotes the set of remaining unmasked features that are to be considered during backward selection. For the current subset S, step 3 in **BackSelect** identifies which remaining feature $i \in S$ produces the *minimal* reduction

in $f(\mathbf{x}_S) - f(\mathbf{x}_{S\setminus\{i\}})$ (meaning it least reduces the output of f if additionally masked), a question trivially answered by running each of the remaining possibilities through the model. This strategy aims to gradually mask out the least important features in order to reveal the core input pattern that is perceived by the model as sufficient evidence for its decision. Finally, we build our SIS up from the last ℓ features omitted during the backward selection, selecting a ℓ value just large enough to meet our sufficiency criterion (1). Because we desire minimality of the SIS as specified by (2), it is not appropriate to terminate the backward elimination in **BackSelect** as soon as the sufficiency condition $f(\mathbf{x}_S) \geq \tau$ is violated, due to the possible presence of local minima in f along the path of subsets encountered during backward selection (as shown in Figure 2-19).

Because this approach always queries a prediction over the joint set of remaining features S, it is better suited to account for interactions between these features and ensure their sufficiency (i.e., that $f(\mathbf{x}_S) \geq \tau$) compared to a forward selection in the opposite direction which builds the SIS upwards one feature at a time by greedily maximizing marginal gains. Throughout its execution, **BackSelect** attempts to maintain the sufficiency of $\mathbf{x}_S$ as the set S shrinks. Given p input features, our algorithm requires $\mathcal{O}(p^2k)$ evaluations of f to identify k SIS, but we can achieve $\mathcal{O}(pk)$ by parallelizing each argmax in **BackSelect** (for example, by batching on GPU).

2.3 Experimental Overview

In the following sections, we apply our SIS method to analyze neural networks in three settings: (1) a natural language task involving multi-aspect sentiment analysis on beer reviews, (2) predicting transcription factor binding in biological data, and (3) image classification on handwritten digits. **SIScollection** is compared with alternative methods for producing rationales (details in Section 2.3.1). Note that our **BackSelect** procedure determines an ordering of elements, R, subsequently used to construct the SIS. Depictions of each SIS are shaded based on the feature order in R (darker = later), which can indicate relative feature importance within the SIS.

In the "Suff. IG," "Suff. LIME," and "Suff. Perturb." (*sufficiency constrained*) methods, we instead compute the ordering of elements R according to the feature attribution values output by integrated gradients (Sundararajan et al., 2017), LIME (Ribeiro et al., 2016), or a perturbative approach that measures the change in prediction when individually masking each feature (see Section 2.3.1). The rationale subset S produced under each method is subsequently assembled using **FindSIS** exactly as in our approach and thus is guaranteed to satisfy $f(\mathbf{x}_S) \geq \tau$. In the "IG," "LIME," and "Perturb." (*length constrained*) methods, we use the same previously described ordering R, but always select the same number of features in the rationale as in the SIS produced by our method (per example). We also compare against the additional "Top IG" method, in which top features from R are added into the rationale until sum of integrated gradients attributions suggests that the rationale has met our sufficiency criterion (see Section 2.3.1).

2.3.1 Details of Baseline Methods

Throughout this chapter, we employ a number of alternative methods for identifying rationales for comparison with SIS. Here, we provide detailed descriptions and implementation details of these baseline methods.

We use methods based on integrated gradients (Sundararajan et al., 2017), LIME (Ribeiro et al., 2016), and feature perturbation. Note that integrated gradients is an attribution method which assigns a numerical score to each input feature. LIME likewise assigns a weight to each feature using a local linear regression model for f around $\mathbf{x}$. In the perturbative approach, we compute the change in prediction when each feature is individually masked, as in Equation 2.1 (of Section 2.4.3). Each of these feature orderings R is used to construct a rationale using the **FindSIS** procedure (Section 2.2) for the "Suff. IG," "Suff. LIME," and "Suff. Perturb." (*sufficiency constrained*) methods.

Note that our text classification architecture (described in Section 2.4.1) encodes discrete words as 100-dimensional continuous word embeddings. The integrated gradients method returns attribution scores for each coordinate of each word embedding.

For each word embedding $x_i \in \mathbf{x}$ (where each $x_i \in \mathbb{R}^{100}$), we summarize the attributions along the corresponding embedding into a single score y_i using the L_1 norm: $y_i = \sum_d |x_{id}|$ and compute the ordering R by sorting the y_i values.

We use an implementation of integrated gradients for Keras-based models from `https://github.com/hiranumn/IntegratedGradients`. In the case of the beer review dataset (Section 2.4), we use the mean embedding vector as a baseline for computing integrated gradients. In the case of TF binding (Section 2.5), we use the $[0.25, 0.25, 0.25, 0.25]$ uniform mean vector as the baseline reference value. As suggested in Sundararajan et al. (2017), we verified that the prediction at the baseline and the integrated gradients sum to approximately the prediction of the input.

For LIME and our beer reviews dataset, we use the approach described in Ribeiro et al. (2016) for textual data, where individual words are removed entirely from the input sequence. In our TF binding dataset, LIME replaces bases with the unknown `N` base (represented as the uniform-distribution $[0.25, 0.25, 0.25, 0.25]$). We use the implementation of LIME at: `https://github.com/marcotcr/lime`. The `LimeTextExplainer` module is used with default parameters, except we set the maximal number of features used in the regression to be the full input length so we can order all input features.

Additionally, we explore methods in which we use the same ordering R by these alternative methods but select the same number of input features in the rationale to be the median SIS length in the SIS-collection computed by our method on each example: the "IG," "LIME," and "Perturb." (*length constrained*) methods. In the TF binding models, we use a baseline of zero vectors such that the integrated gradients result along the encoded sequence is also one-hot. We compute the feature ordering based on the absolute value of the non-zero integrated gradient attributions.

In TF binding data (Section 2.5), we add an additional method, "Top IG," in which we compute integrated gradients using an all-zeros baseline and order features by attribution magnitude (as in the length constrained IG method). But, we select elements for the rationale by finding the minimum number of elements necessary such that the sum of integrated gradients of those features equals $\tau - f(\mathbf{0})$, where $\mathbf{0}$ is the

all-zeros baseline for integrated gradients. Note that for the length constrained and Top IG methods, there is no guarantee of sufficiency $f(\mathbf{x}_S) \geq \tau$ for any input subset S.

2.4 Sentiment Analysis of Reviews

We first consider a dataset of beer reviews from McAuley et al. (2012) where beers receive text reviews along with numerical ratings of aspects like aroma, appearance, and palate. Taking the text of the review as input, three different LSTM recurrent neural networks (Hochreiter and Schmidhuber, 1997) are trained to predict the continuous-valued sentiment toward each aspect. We apply our SIS method to interpret and validate the decisions made by these LSTMs.

In this section, we present results interpreting the LSTM trained to predict sentiment toward the *aroma* aspect in particular. Results for the *appearance* and *palate* aspects are similar and can be found in the Supplementary Material of Carter et al. (2019).

2.4.1 Dataset and Model Details

Following Lei et al. (2016), we use a preprocessed version of the BeerAdvocate[1] dataset[2] which contains decorrelated numerical ratings toward three aspects: *aroma*, *appearance*, and *palate* (each normalized to $[0, 1]$). Dataset statistics can be found in Table 2.1. Reviews are tokenized by converting to lowercase and filtering punctuation, and we use a vocabulary containing the top 10,000 most common words. McAuley et al. (2012) also provide a subset of human-annotated reviews, in which humans manually selected full sentences in each review that describe the relevant aspects. This annotated set is never seen during training and used solely as part of our evaluation.

Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997)

[1] `https://www.beeradvocate.com/`

[2] `http://snap.stanford.edu/data/web-BeerAdvocate.html`

are commonly employed for natural language tasks such as sentiment analysis (Wang et al., 2016; Radford et al., 2017). In these experiments, we use a recurrent neural network (RNN) architecture with two stacked LSTMs as follows:

1. **Input/Embeddings Layer**: Sequence with 500 timesteps, the word at each timestep is represented by a (learned) 100-dimensional embedding
2. **LSTM Layer 1**: 200-unit recurrent layer with LSTM (forward direction only)
3. **LSTM Layer 2**: 200-unit recurrent layer with LSTM (forward direction only)
4. **Dense**: 1 neuron (sentiment output), sigmoid activation

Taking the text of a review as input, different LSTM networks are trained to predict user-provided numerical ratings of each aspect. We train the models using the Adam optimizer (Kingma and Ba, 2015) to minimize mean squared error (MSE) on the training set. We use a held-out set of 3,000 examples for validation (sampled at random from the pre-defined test set from Lei et al. (2016)). Our test set consists of the remaining 7,000 test examples. Training results are shown in Table 2.1.

2.4.2 Applying SIS to Interpret Sentiment Predictors

We apply SIS to interpret the LSTM's decisions on the set of reviews containing sentence-level annotations (Annotation fold in Table 2.1). Note that these reviews (and the human annotations) were not seen during training. Tokens in the input sequence are masked by replacement with a mean embedding taken over the learned vocabulary (see Appendix A.1.1 for further discussion of our mean imputation approach). In our formulation (Section 2.2), we apply the SIS method to inputs $\mathbf{x}$ for which $f(\mathbf{x}) \geq \tau$. For the sentiment analysis task, we analogously apply our method for both $f(\mathbf{x}) \geq \tau_+$ and $-f(\mathbf{x}) \geq -\tau_-$, where the model predicts either strong positive or strong negative sentiment, respectively. We choose thresholds $\tau_+ = 0.85$, $\tau_- = 0.45$ and extract the complete set of sufficient input subsets using our method. These thresholds were set empirically such that they were sufficiently apart, based on the predictive distribution on the held-out annotated set (shown in Figure 2-1). For

Table 2.1: Summary and performance statistics (mean squared error (MSE) and Pearson correlation coefficient ρ) for LSTM models on beer reviews data.

Aspect	Fold	Size	MSE	Pearson ρ
Appearance	Train	80,000	0.016	0.864
	Validation	3,000	0.024	0.783
	Test	7,000	0.023	0.801
	Annotation	994	0.020	0.563
Aroma	Train	70,000	0.014	0.873
	Validation	3,000	0.024	0.767
	Test	7,000	0.025	0.756
	Annotation	994	0.021	0.598
Palate	Train	70,000	0.016	0.835
	Validation	3,000	0.029	0.680
	Test	7,000	0.028	0.694
	Annotation	994	0.016	0.592

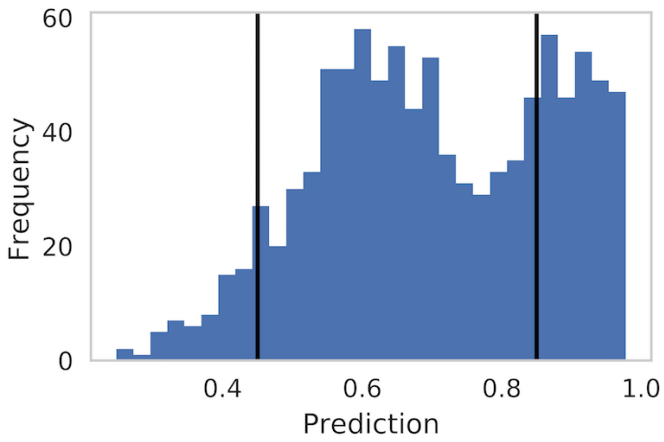


Figure 2-1: Predictive distribution on the annotation set (held-out) using the LSTM model for aroma. Vertical lines indicate decision thresholds ($\tau_+ = 0.85$, $\tau_- = 0.45$) selected for **SIScollection**.

most reviews, **SIScollection** outputs a collection containing just one or two sufficient input subsets (Figure 2-2).

Figure 2-3 shows a sample beer review in which we highlight the SIS identified for the LSTMs that predict each aspect. In this example, the SIS-collection for each of the three LSTMs only contained a single sufficient input subset. We see that each SIS only captures sentiment toward the relevant aspect (as compared to just general positive sentiment), revealing that the LSTMs have learned to make predictions based on context-specific features.

Figure 2-4 depicts the SIS-collection identified from a review the LSTM decided to flag for positive aroma. Here, the SIS-collection for this review is comprised of three sufficient input subsets. From this example, we can see that the aroma LSTM is generally making predictions based on disjoint pieces of evidence in the text suggesting positive sentiment toward the aroma aspect. However, the example also shows how this type of analysis may be used to debug or improve the model. While the rationales generally seem sound, a practitioner may desire that the models not include tokens such as "t" or "s" (which are likely artifacts of tokenization) in the rationales for its decisions.

To gain further insight into the behavior of the LSTM models, we next analyze

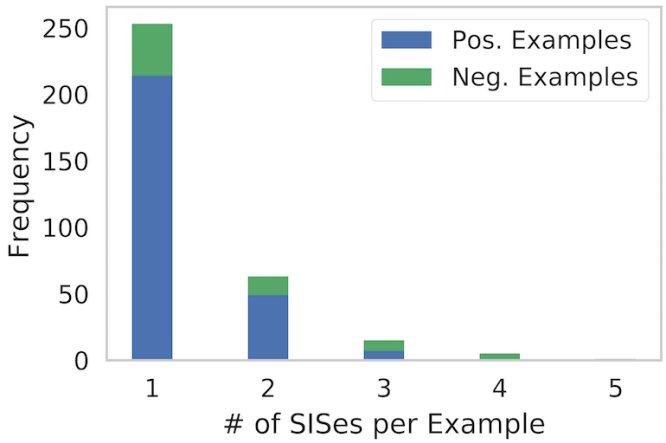


Figure 2-2: Number of sufficient input subsets for aroma identified by **SIScollection** per example.

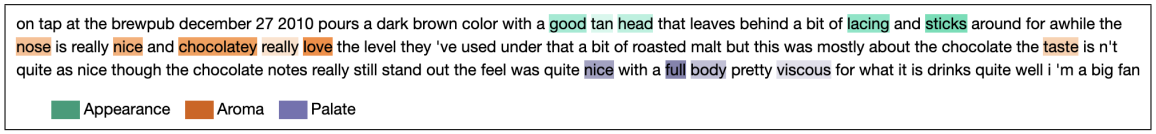


Figure 2-3: Beer review with one sufficient input subset identified for the prediction of each aspect.

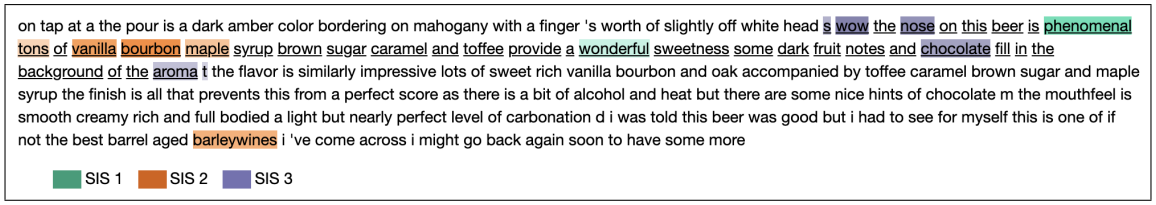


Figure 2-4: Beer review with three disjoint SIS S_1, S_2, S_3 identified for a positive aroma prediction. Underlined are sentences that human labelers manually annotated as capturing the aroma sentiment.

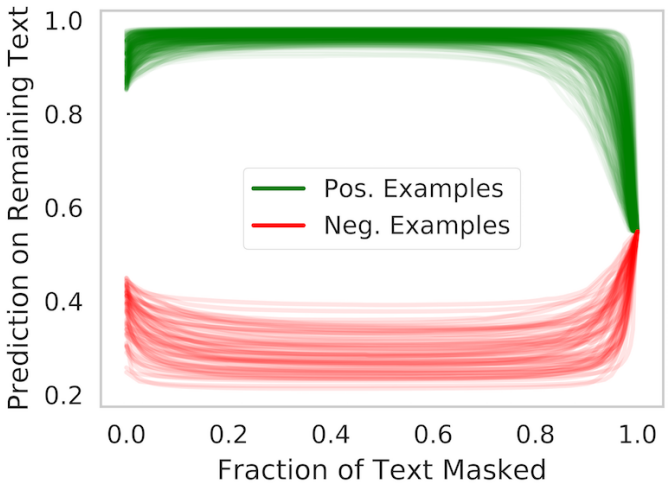


Figure 2-5: Prediction history on remaining (unmasked) text at each step of the **BackSelect** procedure, for examples where aroma sentiment is predicted.

the predictor model's output following the elimination of each feature in the **BackSelect** procedure (Section 2.2). Figure 2-5 shows the LSTM output on the remaining unmasked text $f(\mathbf{x}_{S\setminus\{i^*\}})$ at each iteration of **BackSelect**, for all examples. This figure reveals that only a small number of features are needed by the model in order to make a strong prediction (most features can be removed without changing the prediction). We observe that as the final features are removed, there is a rapid, monotonic decrease in output values. Finally, we see that the first features to be removed by **BackSelect** are those which generally provide negative evidence against the decision. The prediction becomes more positive (or negative in the case of strong negative sentiment reviews [red]) as the first features are eliminated. Note, however, that **BackSelect** may exhibit different behavior when applied to other models or architectures (see Figure 2-19 for one such example).

2.4.3 Comparing SIS to Baseline Methods

We next compare rationales produced by **SIScollection** to those from the baseline methods described in Section 2.3.1. Figure 2-6 shows the prediction on the rationale only (all other words masked) vs. length of rationale for the rationales produced by these various methods on the same set of beer reviews on which the LSTM predicts

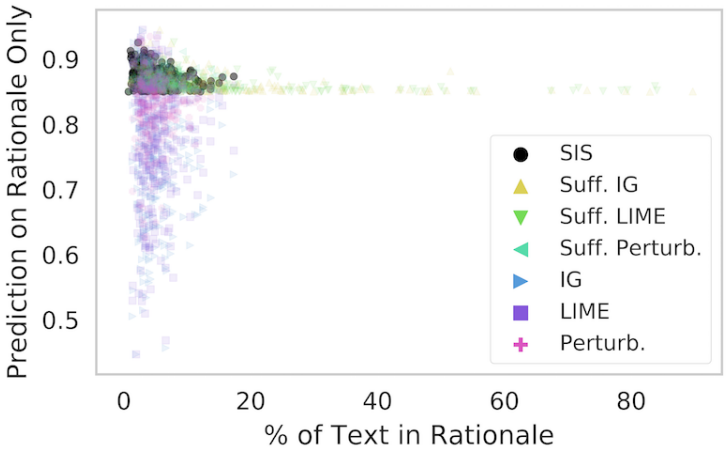


Figure 2-6: Prediction on rationales only vs. rationale length for various methods in reviews with positive aroma prediction ($\tau = 0.85$).

positive aroma. From this figure, we see that when the alternative methods are length constrained, the rationales they produce often badly fail to meet our sufficiency criterion. Thus, even though the same number of feature values are preserved in the rationale and these alternative methods select the features to which they have assigned the largest attribution values, their rationales lead to significantly reduced f outputs compared to our SIS subsets. When the lengths of these alternative rationales is allowed to grow large enough to ensure our sufficiency criterion is met, the rationales become unnecessarily long. If the sufficiency constraint is instead enforced for these alternative methods, the rationales they identify become significantly larger than those produced by **SIScollection**, and they also tend to contain a larger number of unimportant features (as shown in Table 2.2 and Figures 2-7 and 2-8, detailed below). Thus, our SIS method effectively extracts rationales that are sparse yet suffice for a strong prediction by the model.

We also compare the rationales from our SIS method those from the baseline methods by comparing the importance of features that comprise the rationales. For each word i in an input sequence $\mathbf{x}$, we quantify its marginal importance by individually perturbing only this word:

$$\text{Feature Importance}(i) = f(\mathbf{x}) - f(\mathbf{x} \setminus \{i\}) \tag{2.1}$$

Table 2.2: Statistics for rationale length and feature importance in aroma prediction. For rationale length, median and max indicate percentage of input text in the rationale. For marginal perturbed feature importance, we indicate the median importance of features in rationales and features from the other (non-rationale) text. p-values are computed using a Wilcoxon rank-sum test (comparing each distribution to that from SIS).

Method	Rationale Length (% of text)			Marg. Perturbed Feat. Import.		
	Med.	Max	p	Rationale	Other	p
SIS	**3.9%**	**17.3%**	–	0.0112	1.50e-05	–
Suff. IG	7.7%	89.7%	5e-26	0.0068	1.85e-05	3e-42
Suff. LIME	7.2%	84.0%	4e-23	0.0075	1.87e-05	1e-35
Suff. Perturb.	5.1%	18.3%	1e-06	0.0209	1.90e-05	1e-72

Note that these marginal Feature Importance scores are identical to those of the Perturb. method described in Section 2.3.1.

For rationales computed by the various methods on these beer reviews, we compute the marginal Feature Importance of features in the rationales, which are summarized in Table 2.2 and Figure 2-7. Compared to the Suff. IG and Suff. LIME methods, our **SIScollection** technique produces rationales that are much shorter and contain fewer irrelevant (i.e., not marginally important) features (Table 2.2, Figures 2-7 and 2-8). Note that by construction, the rationales of the Suff. Perturb. method contain features with the greatest Feature Importance, since this precisely how the ranking in Suff. Perturb. is defined.

2.4.4 Evaluation of SIS Rationales

Benchmarking interpretability methods is difficult because a learned f may behave counterintuitively such that seemingly unreasonable model explanations are in fact faithful descriptions of a model's decision-making process. For some reviews in our dataset, a human annotator has manually selected which sentences carry the relevant sentiment for the aspect of interest (Section 2.4.1, see examples in the underlined text of Figures 2-4 and 2-10), so we treat these annotations as an alternative rationale

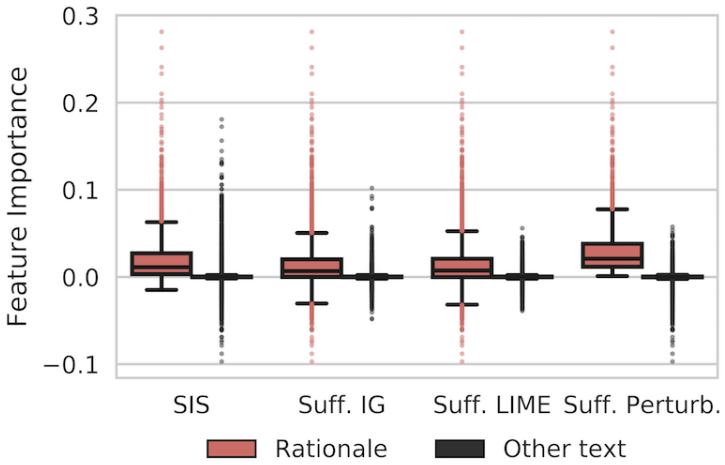


Figure 2-7: Importance of individual features in the rationales for aroma prediction in beer reviews.

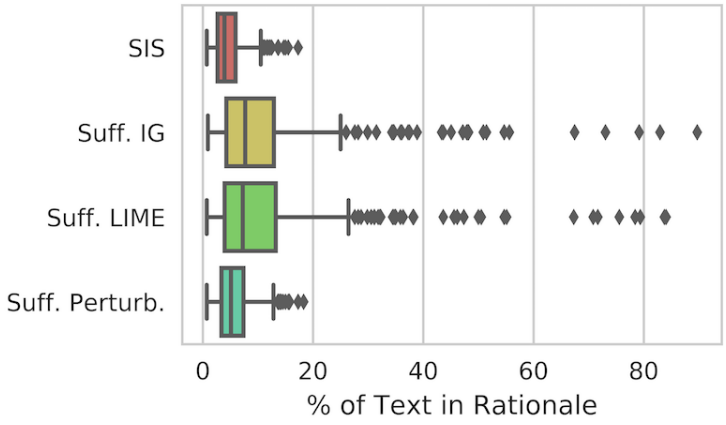


Figure 2-8: Length of rationales for aroma prediction.

for the LSTM prediction. For a review $\mathbf{x}$ whose true and predicted aroma exceed our decision threshold, we define the *quality of human-selected sentences for model explanation* (QHS) as

$$\text{QHS} = f(\mathbf{x}_S) - f(\mathbf{x}) \tag{2.2}$$

where S here is the human-selected-subset of words in the review (as opposed to a sufficient input subset).

Figure 2-9 shows the relationship between QHS and the fraction of the SIS that falls inside the human-selected sentences. There is a positive correlation between the two variables (Pearson $\rho = 0.491$, $p = 1.5\text{e}{-25}$). High variability of QHS in the annotated reviews indicates the human rationales often do not contain sufficient information to preserve the LSTM's decision. As the model diverges from alignment with the human-selected sentences (and those sentences are not necessary for prediction), fewer words in the sufficient input subsets lie within those sentences (lower left of Figure 2-9). Additionally, as the human-selected sentences become more sufficient for prediction (QHS $\to 0$), almost the entirety of the sufficient input subsets identified by our method end up lying within those sentences (upper right of Figure 2-9). Figure 2-10 provides examples from both extremes of alignment (SIS has good alignment with human-selected sentences, where QHS ≈ 0, and SIS and human-selected sentences have poor alignment, where QHS < 0). The bottom panel of Figure 2-10 illustrates an example where the LSTM is able to predict positive sentiment from features that diverge from what a human would expect, which may suggest overfitting.

2.5 Transcription Factor Binding

We next analyze SIS in the context of convolutional neural networks (CNNs) trained to classify whether a given transcription factor (TF) will bind to a specific DNA sequence (Zeng et al., 2016). This setting also provides us with ground truth motifs containing known binding sites, which we use to evaluate the ability of SIS to recover such motifs (Section 2.5.3).

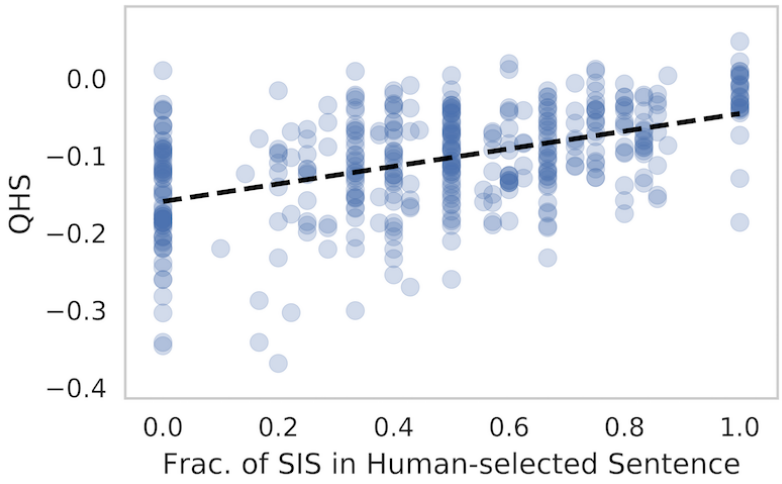


Figure 2-9: QHS (Equation 2.2) vs. similarity between SIS and annotation in the reviews with positive aroma sentiment (Pearson $\rho = 0.491$, p-value $= 1.5\text{e}{-25}$).

Good Alignment with Human-selection: QHS = 0.00296, Full Sequence Prediction = 0.900

poured from a 24oz bottle into a large appearance this pours a deep bronze amber in color this beer has some of the best head formation and retention that i have ever seen along with lots and lots of sticky lacing smell tons of piney resinous evergreen abound in this ale cant get enough of this beer its the best smelling harvest ale i 've ever had the pleasure of enjoying taste huge flavor profile with lots of bitterness and only a little malt sweetness as this beer warms up the bitterness really dominates the flavor i 'm tasting lot of subtle orange aromas mouthfeel full body beer with alot of carbonation overall i love this beer in my opinion its nevada 's best beer overall the price is i bought my for 3 99 so much flavor and biting bitterness this harvest ale has no equal

Poor Alignment with Human-selection: QHS = -0.3037, Full Sequence Prediction = 0.928

like having a bowl of honey nut of malt for breakfast this beer makes me as happy as a pearly going all from tree to tree the fat is actually pretty interesting especially when compared to most brown ales its aromas and flavor are a perfectly balanced blend of honey hazelnut toast walnut and a tad smidgen of vanilla the mouthfeel is light and smooth not hoppy in the slightest quite sweet too damn drinkable for our own good

Figure 2-10: Beer reviews (aroma) in which human-selected sentences (underlined) are aligned well (top) and poorly (bottom) with predictive model. Fraction of SIS in the human sentences corresponds accordingly. In the bottom example (poor alignment between human-selection and predictive model), our procedure has surfaced a case where the LSTM has learned features that diverge from what a human would expect (and may suggest overfitting).

2.5.1 Dataset and Model Details

We use the *motif occupancy* datasets[3] from Zeng et al. (2016), where each dataset originates from a ChIP-seq experiment from the ENCODE project (Consortium et al., 2012). Each of the 422 datasets studies a particular transcription factor, containing between 600 and 700,000 (median 50,000) 101 base-pair DNA sequences (inputs) each associated with a binary label based on whether the sequence is bound by the TF or not. Each dataset also contains a test set ranging between 150 and 170,000 sequences (median 12,000). Here, the positive and negative classes in each dataset are balanced, and we filter out all sequences containing the unknown base (`N`). The nucleotide occurring at base position (`A`, `C`, `G`, `T`) is encoded as a one-hot representation which is fed into the CNN. Zeng et al. (2016) showed that convolutional neural network architectures outperform other models for this TF binding prediction task.

For each of the 422 prediction tasks, we employ the optimal "1layer_128motif" architecture from Zeng et al. (2016), defined as follows:

1. **Input**: (101 x 4) sequence encoding
2. **Convolutional Layer 1**: Applies 128 kernels of window size 24, with ReLU activation
3. **Global Max Pooling Layer 1**: Performs global max pooling
4. **Dense Layer 1**: 32 neurons, with ReLU activation and dropout probability 0.5
5. **Dense Layer 2**: 1 neuron (output probability), with sigmoid activation

We hold out 1/8 of each train set for validation and minimize binary cross-entropy using the Adadelta optimizer (Zeiler, 2012) with default parameter settings in Keras (Chollet et al., 2015). We train each model on each of the 422 datasets for 10 epochs (using batch size 128) with early-stopping based on validation loss. Figure 2-11 shows the area under the receiver operating curve (AUC) over the 422 datasets, and we note that the performance of our models closely resembles that in Zeng et al. (2016).

[3]available at `http://cnn.csail.mit.edu`

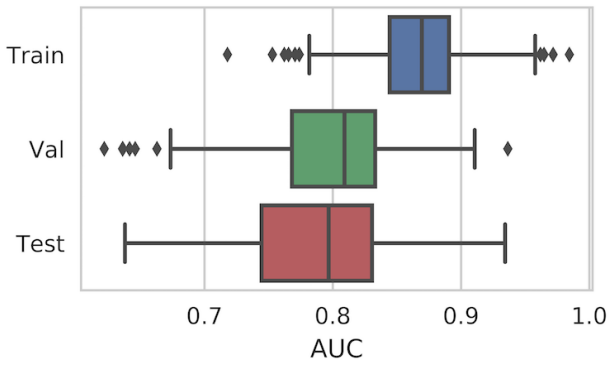


Figure 2-11: Median area under the receiver operating curve (AUC) for all 422 transcription factor binding motif occupancy datasets. The validation set is held-out at training but used to choose model parameters; the test set is not seen until after training.

2.5.2 Applying SIS to Interpret TF Binding Classifiers

From each of the 422 different datasets of DNA sequences bound-or-not by different TFs (and 422 different CNN models, see Section 2.5.1), we extract SIS-collections from sequences in the test set with high (top 10%) predicted binding affinity for the TF profiled in each dataset. The distribution of threshold τ over the 422 datasets is shown in Figure 2-12. Since `A`, `C`, `G`, `T` nucleotides all occur with similar frequency in this data, our SIS analysis simply masks each base using a uniform embedding ($[0.25, 0.25, 0.25, 0.25]$). This is also the standard strategy to represent unknown `N` nucleotides in DNA sequences that typically arise from issues in read quality. We generally find that there is only a single SIS per example for the sequences in these datasets. Figure 2-13 depicts two examples of input DNA sequences and the corresponding sufficient input subsets identified by our **SIScollection** procedure.

We evaluate the minimality and sufficiency of the rationales produced by SIS to those produced by the alternative baseline methods we explored (see Section 2.3.1). On each dataset, we compute the median rationale length (as number of bases in the rationale). The distribution of median rationale length over all datasets by the various methods is shown in Figure 2-14. Note that for the IG, LIME, and Perturb. methods, rationale length was constrained to the length of the rationales produced by our method, as described in Section 2.3.1. For the Top IG method, neither sufficiency

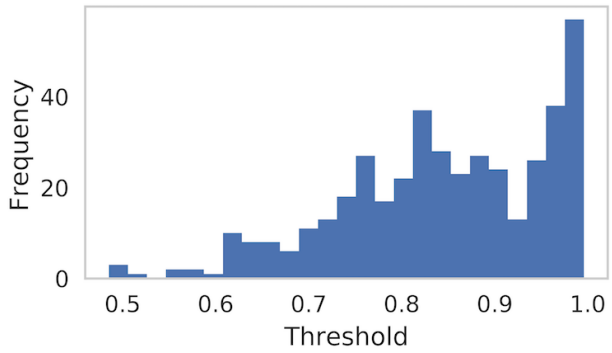


Figure 2-12: Thresholds τ used for identifying sufficient input subsets in TF binding datasets. In each dataset, the threshold is defined as the 90th percentile of the predictive test distribution.

CACTGTCATTCTCTTGGTCAGCCCTGGACATCCCTGGAAAGGATGACTCAGCTGTCCGTTTTAAACAGGGTAGTTCAGAAGAATACATTCCTGGTTATTCA
TTTTTTTCTCCCTTCGATTTCCACTATGATTTGTATTTCCTTTGTTCTGCTGACTTTGCAATTTCGGTTGTTTTTTCTAAATTTCTTAGGGTGAAAACTGA

Figure 2-13: Two DNA sequences that receive positive TF binding predictions for the MAFF factor (SIS is shaded).

nor length constraints are enforced. We see that when the sufficiency constraint is enforced in alternative methods (Suff. IG), the rationales are significantly longer than those identified by SIS. Moreover, as shown in Figure 2-15, when the sufficiency constraint is not enforced (or the rationale lengths are constrained to the length of SIS rationales) in alternative methods, the rationales have significantly less predictive power, often not satisfying $f(\mathbf{x}_S) \geq \tau$. The rationales produced via our SIS approach are shorter and better at preserving large f-values than rationales from other methods (Figures 2-14 and 2-15).

2.5.3 Evaluation of the Quality of TF Rationales

To predict binding so accurately (Figure 2-11), the CNN must faithfully reflect the biological mechanisms that relate the DNA sequence to the probability of TF occupancy. We evaluate the rationales found by SIS and our baseline methods (see Section 2.3.1) against known TF binding motifs from JASPAR (Mathelier et al., 2016) as the ground truth. We adopt KL divergence between the known motif and each proposed rationale as a measure of quality of the rationale. Each rationale is padded

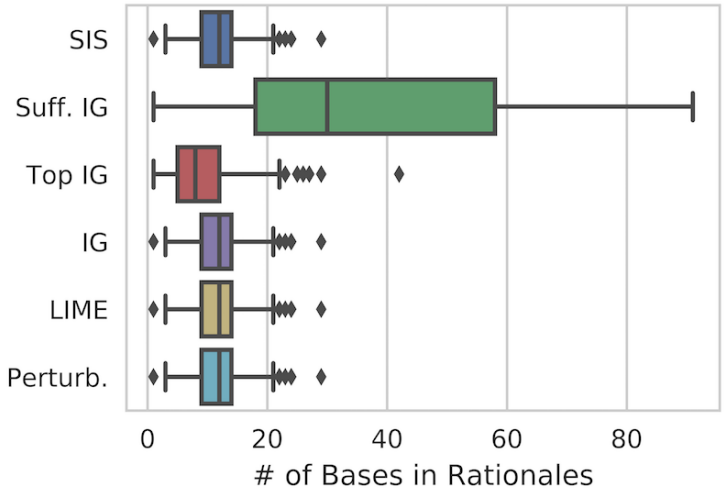


Figure 2-14: Length (number of bases) of rationales identified by various methods. Note that the sufficiency constraint ($f(\mathbf{x}_S) \geq \tau$) is only enforced for SIS and Suff. IG. The lengths of IG, LIME, and Perturb. rationales are constrained to the length of SIS rationales.

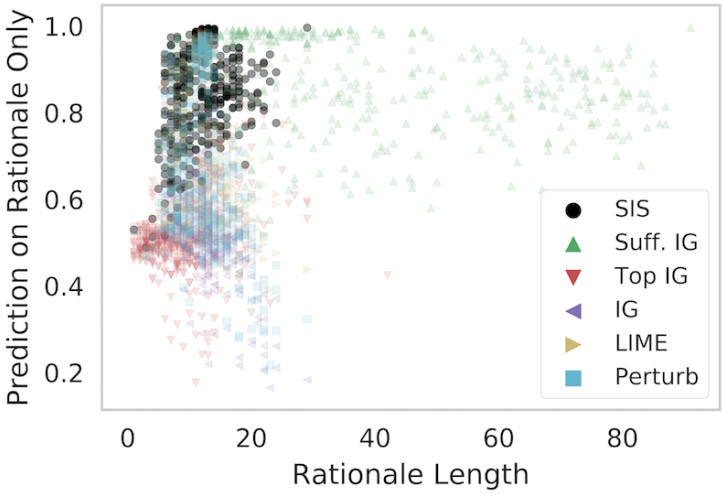


Figure 2-15: Prediction on rationale only (all other bases masked) vs. rationale length (number of bases) for various methods in the TF binding task.

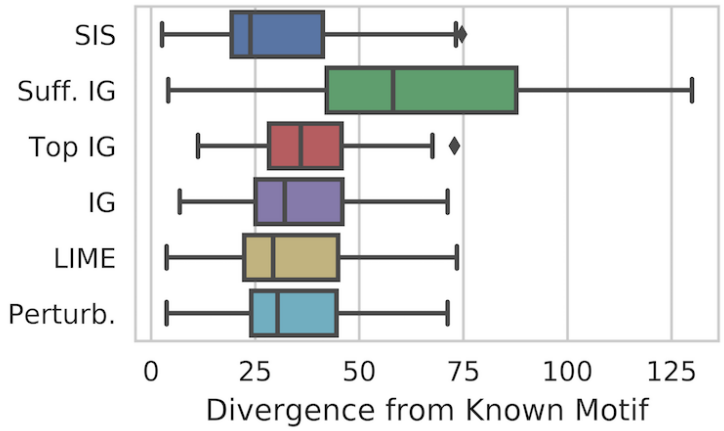


Figure 2-16: KL divergence between JASPAR motifs (known ground truth) and rationales found via various methods. Shown are results for 422 TF datasets (each one summarized by median divergence).

.

with "N" (unknown) bases to the length of a full input sequence (101 bases) and optimally aligned with the known motif[4] according to the likelihood criterion. The aligned motif is then also padded to the same length, and we compute the divergence between between the rationale R and known motif M as:

$$\text{Div}(R, M) = \sum_i D_{\text{KL}}(R_i || M_i) \tag{2.3}$$

where $D_{\text{KL}}(R_i||M_i) = \sum_j R_i(j) \log \frac{R_i(j)}{M_i(j)}$ is the Kullback-Leibler (KL) divergence from M_i to R_i, and M_i and R_i are distributions over bases (`A`, `C`, `G`, `T`) at position i. Note that as R and M become more dissimilar, $\text{Div}(R, M)$ increases. We ensure $M_{ij} > 0 \; \forall \; i, j$ so D_{KL} is always finite. Figure 2-16 shows the divergence of rationales produced by **SIScollection** is significantly lower than that of rationales identified using other methods (Wilcoxon $p \leq$ 1e$-$5 in all cases). SIS is thus more effective at uncovering these underlying biological principles than the alternative methods we explored.

[4]A JASPAR motif is a $n \times 4$ right stochastic matrix M. The columns represent the ACGT DNA bases and the rows a DNA sequence. It represents the marginal probability of the base j at position i being present with probability M_{ij}. The unknown base "N" receives uniform 1/4 probability for each of ACGT. An example JASPAR motif is shown in Figure 2-20.

2.6 MNIST Digit Classification

In this section, we apply SIS to interpret a 10-way convolutional neural network (CNN) classifier trained on the MNIST handwritten digits data (LeCun et al., 1998). In addition to interpreting the classifier's decisions on correctly classified examples, we see how SIS can be further employed to understand the basis for the CNN's misclassifications (Section 2.6.2). In this application, we also observe the effect of local minima in the backward selection phase of the SIS procedure and show how our method facilitates minimality of the resulting rationales (Section 2.6.3).

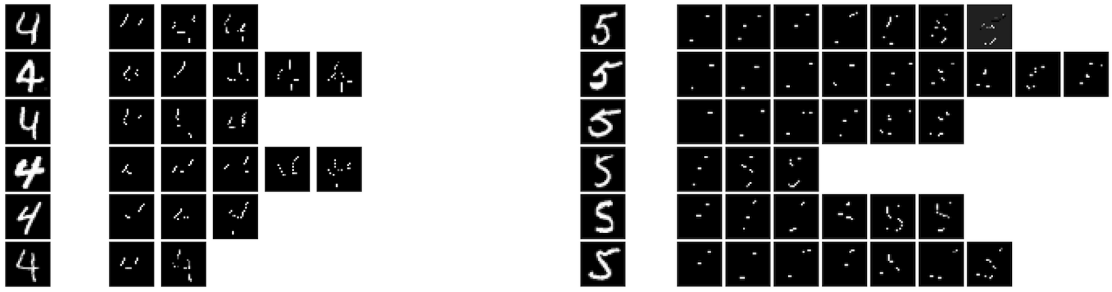
2.6.1 Dataset and Model Details

The MNIST database of handwritten digits contains 60k training images and 10k test images (LeCun et al., 1998). All images are 28x28 grayscale, and we normalize them such that all pixel values are between 0 and 1. We train a simple 10-way CNN to classify the images using the same architecture as that provided in the Keras MNIST CNN example.[5] The architecture is defined as follows:

1. **Input**: (28 x 28 x 1) image, all values $\in [0, 1]$
2. **Convolutional Layer 1**: Applies 32 3x3 filters with ReLU activation
3. **Convolutional Layer 2**: Applies 64 3x3 filters, with ReLU activation
4. **Pooling Layer 1**: Performs max pooling with a 2x2 filter and dropout probability 0.25
5. **Dense Layer 1**: 128 neurons, with ReLU activation and dropout probability 0.5
6. **Dense Layer 2**: 10 neurons (one per digit class), with softmax activation

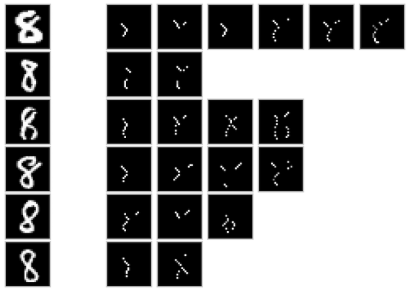
The Adadelta optimizer (Zeiler, 2012) is used to minimize cross-entropy loss on the training set. The final model achieves 99.7% accuracy on the train set and 99.1% accuracy on the held-out test set.

[5] `http://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py`

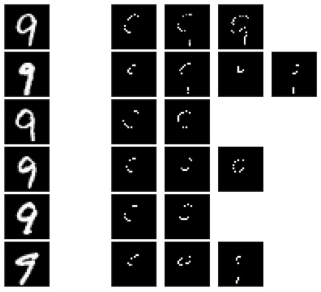


(a) Digit 4

(b) Digit 5



(c) Digit 8



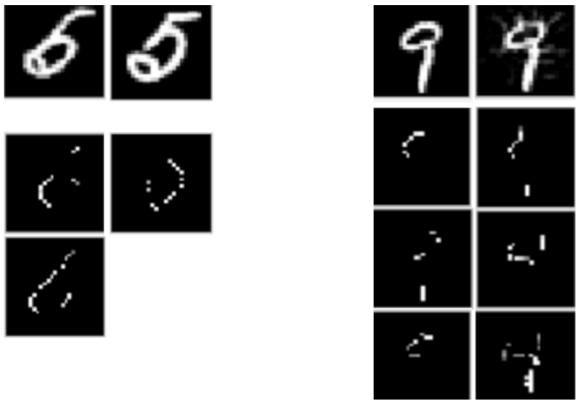
(d) Digit 9

Figure 2-17: Visualization of SIS-collections identified from MNIST digits **(a)** 4, **(b)** 5, **(c)** 8, and **(d)** 9 that are confidently classified by the CNN. For each class, six examples were chosen randomly. For each example, we show the original image (left) and the complete set of sufficient input subsets identified for that example (remaining images in each row). Each individual SIS depicted satisfies $f(\mathbf{x}_S) \geq 0.7$ for that class.

2.6.2 Applying SIS to Interpret Image Classifiers

When applying SIS to interpret the CNN's classification of MNIST handwritten digits, we only consider predicted probabilities for one class of interest at a time and always set $\tau = 0.7$ as the probability threshold for deciding that an image belongs to the class. We then extract the SIS-collections of all corresponding MNIST test set examples. Examples of the complete SIS-collection corresponding to randomly chosen digits are shown in Figure 2-17.

We also employ the SIS procedure to rationalize the CNN's misclassifications. We explore misclassifications of natural images in the MNIST test set as well as adversarially modified images. Figure 2-18a shows two (unmodified) MNIST digits whose true labels are 5 but which are misclassified by the CNN as 6 and 0, respectively. The SIS-collections depicted for these digits immediately enable us to understand the basis for why the misclassifications occur.



(a) (b)

Figure 2-18: **(a)** SIS for digits 5 that are misclassified as 6 (1st column) and as 0 (2nd column). **(b)** SIS for correctly classified 9 (1st column) and when adversarially perturbed toward class 4 (2nd column).

Figure 2-18b illustrates how the SIS-collection drastically changes for an example of a correctly-classified 9 that has been adversarially manipulated (Carlini and Wagner, 2017b) to become confidently classified as the digit 4. Although a visual inspection of the perturbed image does not really reveal exactly how it has been manipulated, it becomes immediately clear from the SIS-collection for the adversarial image. The SIS-collections show that the perturbation modifies pixels in such a way that input patterns similar to the typical SIS-collection for a 4 (Figure 2-21) become embedded in the image. The adversarial manipulation was done using the Carlini-Wagner L_2 (CW2) attack[6] (Carlini and Wagner, 2017b) with a confidence parameter of 10. The CW2 attack tries to find the minimal change to the image, with respect to the L_2 norm, that will lead the image to be misclassified. Carlini and Wagner (2017a) demonstrate it to be one of the strongest extant adversarial attacks.

2.6.3 Local Minima in Backward Selection

Figure 2-19 demonstrates an example MNIST digit for which there exists a local minimum in the backward selection phase of our algorithm to identify the initial SIS. Note that if we were to terminate the backward selection as soon as predictions drop

[6]Implemented in the cleverhans library of Papernot et al. (2017)

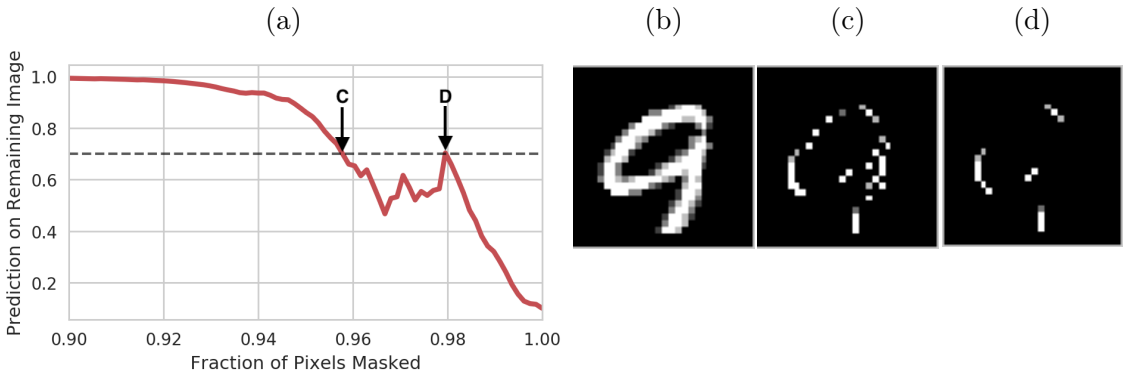


Figure 2-19: **(a)** Prediction on remaining image as pixels are masked during backward selection, when our CNN classifier is fed the MNIST digit in (b). The dashed line depicts the threshold $\tau = 0.7$. **(b)** Original image (class 9). **(c)** SIS if backward selection were to terminate the first time prediction on remaining image drops below 0.7, corresponding to point **C** in (a) (CNN predicts class 9 with probability 0.700 on this SIS). **(d)** Actual SIS produced by our **FindSIS** algorithm, corresponding to point **D** in (a) (CNN predicts class 9 with probability 0.704 on this SIS).

below the decision threshold, the resulting SIS would be overly large, violating our minimality criterion. It is also evident from Figure 2-19 that the smaller-cardinality SIS in (d), found after the initial local optimum in (c), presents a more interpretable input pattern that enables better understanding of the core motifs influencing our classifier's decisions. To avoid suboptimal results, it is important to run a complete backward selection sweep until the entire input is masked before building the SIS upward, as done in our **SIScollection** procedure (Section 2.2).

2.7 Clustering SIS for Global Insights

Identifying the different input patterns that justify a decision can help us better grasp the general operating principles of a model. To this end, we cluster all of the sufficient input subsets produced by our SIS method applied across a large number of examples that receive the same decision by a particular model. Here, we cluster the sufficient input subsets using DBSCAN (Ester et al., 1996), a widely applicable algorithm that only requires specifying pairwise distances between the SIS. This approach allows us to choose a suitable distance metric between sufficient input subsets depending on

the particular domain.

2.7.1 Clustering SIS from Sentiment Predictors

We first cluster the sufficient input subsets found across held-out[7] beer reviews (Test fold in Table 2.1) that received positive aroma predictions from our LSTM model (model details in Section 2.4.1). The distance between two SIS is taken here as the Jaccard (intersection over union) distance between their bag of words representations, S_1 and S_2:

$$D(S_1, S_2) = 1 - \frac{S_1 \cap S_2}{S_1 \cup S_2} \tag{2.4}$$

Table 2.3 shows three resulting clusters containing phrases that the LSTM has learned to associate with positive aromas in the absence of other context. The full clustering for SIS from beer reviews with strong positive predicted sentiment can be found in Table A.1 (strong negative predicted sentiment in Table A.2).

2.7.2 Clustering SIS from TF Binding Classifiers

We next apply our clustering procedure to the sufficient input subsets found by our method across all test-set DNA sequences which the CNN model (Section 2.5.1) predicts would be bound by some transcription factor (see Section 2.5). In this application, the pairwise distance between two sufficient input subsets is taken to be the Levenshtein (edit) distance between the string representations of the masked sequences (where non-SIS characters are masked with `N` as in Section 2.5.1).

Figure 2-20 shows the clusters for a particular transcription factor (MAFF) for which two SIS clusters were found, aligned with the known motif from JASPAR (Mathelier et al., 2016) for this TF (discussion of JASPAR motifs in Section 2.5.3). Additional SIS in each of the clusters are given in Table 2.4. Notably, we find that despite contiguity not being enforced in our algorithm, each cluster is comprised of short sequences that clearly capture different aspects of the underlying DNA motif

[7]For experiments involving clustering and/or comparing different models, we use examples drawn from the Test fold (instead of Annotation fold, see Table 2.1) to consider a larger number of examples.

Table 2.3: Three clusters of SIS extracted from beer reviews with positive CNN aroma predictions. Each row shows four most frequent unique SIS in a cluster (each SIS shown as ordered word list with text-positions omitted). Each unique SIS can be present many times in one cluster.

Cluster	**SIS #1**	**SIS #2**	**SIS #3**	**SIS #4**
C_1	smell amazing wonderful	nice wonderful nose	wonderful amazing	amazing amazing
C_2	grapefruit mango pineapple	pineapple grapefruit pineapple grapefruit	hops grapefruit pineapple floyds	mango pineapple incredible
C_3	creme brulee brulee	creme brulee decadent	incredible creme brulee	creme brulee exceptional

known to bind this TF. This result suggests that when the models are expected to behave according to some underlying scientific principles (e.g., those governing DNA transcription factor binding, as captured by the motif), the SIS clustering approach presented here is able to recover them. Had the motif not been known *a priori*, our approach would have enabled us to gain insight into which DNA sequence positions are critical for DNA-TF binding to occur.

2.7.3 Clustering SIS from MNSIT Digit Classifiers

Finally, we apply our clustering methodology the sufficient input subsets found across all MNIST test set examples that are confidently identified by the CNN (Section 2.6) as a particular class. To cluster SIS from the image data, we compute the pairwise distance between two sufficient input subsets S_1 and S_2 as the energy distance (Rizzo and Székely, 2016) between two distributions over the image pixel coordinates that

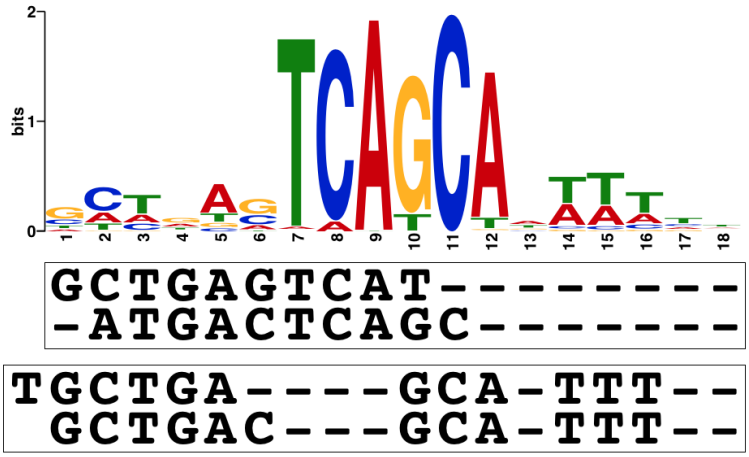


Figure 2-20: Known JASPAR motif (top) and alignment with cluster modes (bottom) for two SIS clusters identified by DBSCAN for one particular TF (MAFF). Additional items in clusters given in Table 2.4.

Table 2.4: Two clusters of SIS resulting from DBSCAN clustering on SIS stemming from CNN predicting binding for a particular transcription factor (MAFF). Most frequently-occurring SIS are each shown for each cluster. Frequency indicates number of times that SIS was observed.

SIS	Freq.
GCTGAGTCAT	197
ATGACTCAGC	185
GCTGAGTCA-C	83
GCTGAGTCAC	53
GCTGACTCAGCA	42

SIS	Freq.
TGCTGA--GCA-TTT	12
GCTGAC--GCA-TTT	8
TGCTGAC--GCA-TT	6
TGCTGAC--GCA-AA	5
TGCTGAC--GCA-AT	4

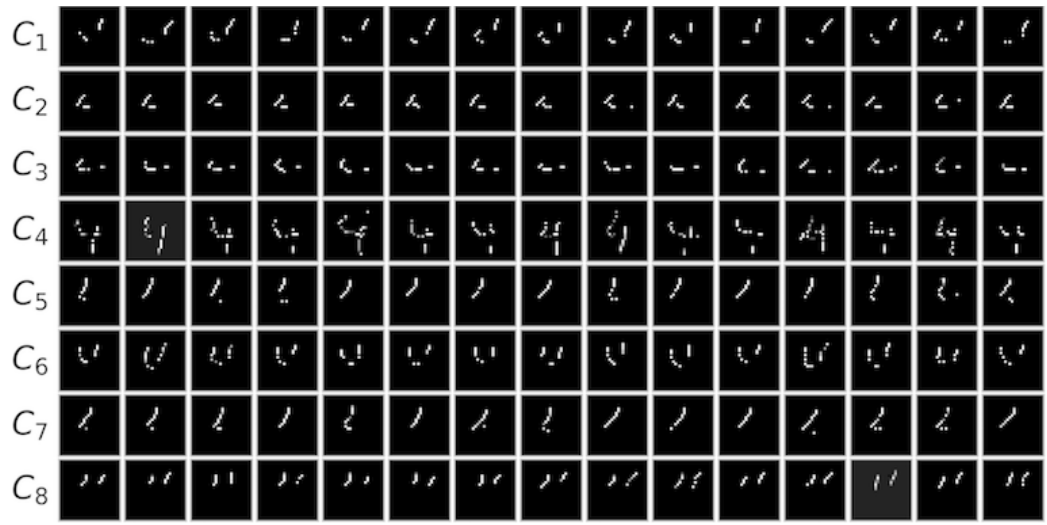


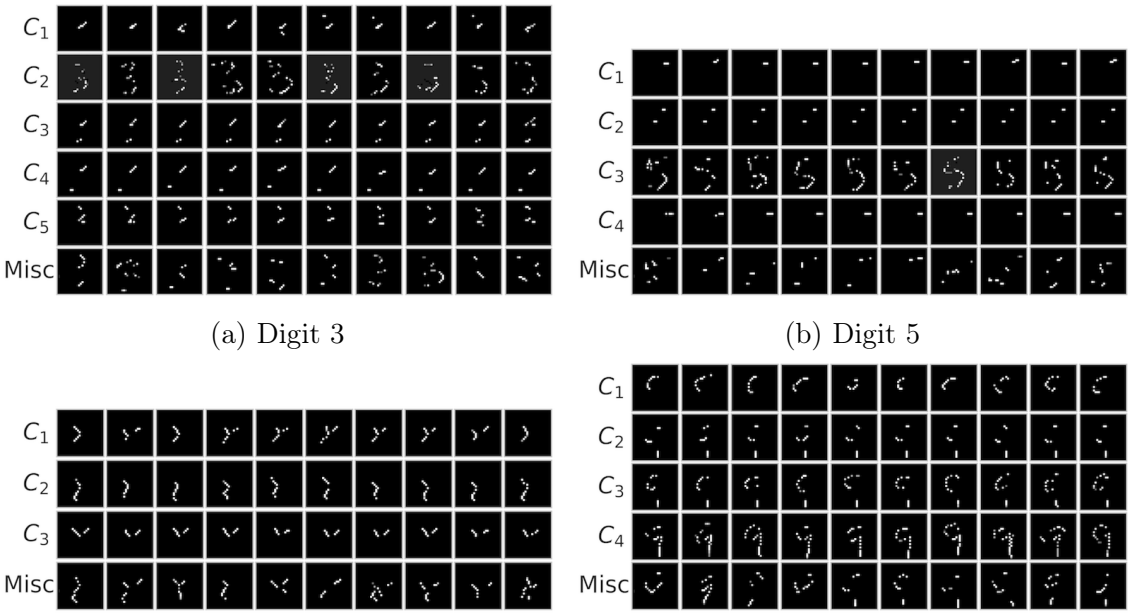
Figure 2-21: Eight clusters of SIS identified from examples of digit 4. Each row contains fifteen random SIS from a single cluster.

comprise the SIS, X_1 and $X_2 \in \mathbb{R}^2$:

$$D(X_1, X_2) = 2 \cdot \mathbb{E}\,||X_1 - X_2|| - \mathbb{E}\,||X_1 - X_1'|| - \mathbb{E}\,||X_2 - X_2'|| \geq 0 \qquad (2.5)$$

Here, X_i is uniformly distributed over the pixels that are selected as part of the SIS subset S_i, X_i' is an i.i.d. copy of X_i, and $||\cdot||$ represents the Euclidean norm. Unlike a Euclidean distance between images, our usage of the energy distance takes into account distances between the similar pixel coordinates that comprise each SIS. The energy distance offers a more efficiently computable integral probability metric than the optimal transport distance, which has been widely adopted as an appropriate measure of distance between images.

Figure 2-21 depicts the SIS clusters identified for digit 4. These clusters reveal distinct feature patterns learned by the CNN to distinguish digit 4 from other digits, which are clearly present in the vast majority of test set images confidently classified as a 4. For example, cluster C_8 depicts parallel slanted lines, a pattern that never occurs in other digits. We repeat this analysis for additional digit classes, and results are shown in Figure 2-22.



(a) Digit 3

(b) Digit 5

(c) Digit 8

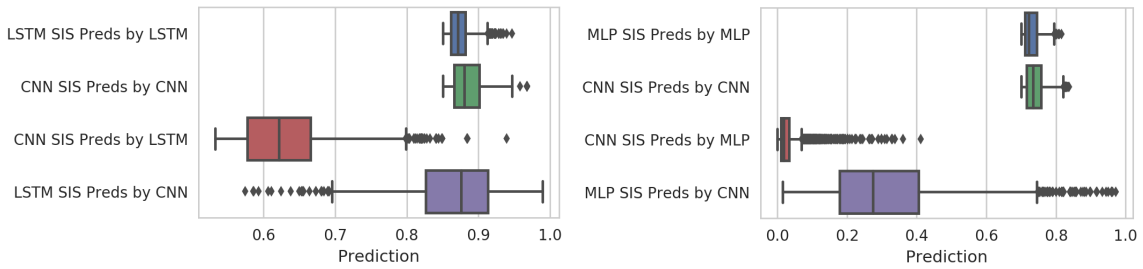
(d) Digit 9

Figure 2-22: Clustering all the SIS found for digits **(a)** 3, **(b)** 5, **(c)** 8, and **(d)** 9 under the CNN model. Each row contains images drawn from one cluster. The bottom row ("Misc") contains a sample of miscellaneous SIS not assigned to any cluster by DBSCAN.

2.8 Understanding Differences Between Models

The general insights revealed by our SIS clustering methodology can also be used to compare and contrast the operating behaviors of different models trained for the same task. In this section, we demonstrate this approach in two of our settings: training a text CNN to compare to our existing LSTM that predicts sentiment in beer reviews (Section 2.4) and training a simple feed-forward neural network to compare to our existing CNN to classify MNIST digits (Section 2.6).

Both networks exhibit similar accuracy in each task, so it is not immediately clear which model would be preferable to use in practice. We first determine whether the SIS extracted under one model are sufficient for the other model to arrive at the same prediction. Figure 2-23 shows the SIS extracted under one model are typically insufficient to receive the same decision from the other model, suggesting that these models base their positive predictions on different evidence. We next adopt our joint SIS clustering methodology to expose the differences in the SIS from each of the



(a) (b)

Figure 2-23: Predictions by one model on the SIS extracted from the other model in: **(a)** beer reviews with positive LSTM/CNN aroma predictions, and **(b)** MNIST digits confidently classified as 4 by CNN/MLP.

architectures in each of these applications in turn.

2.8.1 Understanding Differences Between Sentiment Predictors

In addition to the LSTM (see Section 2.4.1), we train a convolutional neural network (CNN) on the same sentiment analysis task (on the aroma aspect). The text CNN architecture is as follows:

1. **Input/Embeddings Layer**: Sequence with 500 timesteps, the word at each timestep is represented by a (learned) 100-dimensional embedding

2. **Convolutional Layer 1**: Applies 128 filters of window size 3 over the sequence, with ReLU activation

3. **Max Pooling Layer 1**: Max-over-time pooling, followed by flattening, to produce a $(128,)$ representation

4. **Dense**: 1 neuron (sentiment output), sigmoid activation

Note that a new set of embeddings is learned with the CNN. As with the LSTM model, we use Adam (Kingma and Ba, 2015) to minimize MSE on the training set. For the aroma aspect, this CNN achieves 0.016 (0.850), 0.025 (0.748), 0.026 (0.741), 0.014 (0.662) MSE (and Pearson ρ) on the Train, Validation, Test, and Annotation

sets, respectively. We note that this performance is similar to that from the LSTM (Table 2.1).

We apply our **SIScollection** procedure to extract the SIS-collections from all applicable test examples using the text CNN, as in Section 2.4. Figure 2-23a shows the predictions from one model (LSTM or CNN) when fed input examples that are SIS extracted with respect to the *other* model (for reviews predicted to have positive sentiment toward the aroma aspect). Since the word embeddings are model-specific, we embed each SIS using the embeddings of the model making the prediction (note that while the embeddings are different, the vocabulary is the same across the models).

In Table 2.5, we show five example clusters (and cluster composition) resulting from clustering the combined set of all sufficient input subsets extracted by the LSTM and CNN on reviews in the test set for which a model predicts positive sentiment toward the aroma aspect. The complete clustering on reviews receiving positive sentiment predictions is shown in Table A.3 (Table A.4 for reviews receiving negative sentiment predictions). These results suggest that the text CNN tends to learn localized (unigram/bigram) word patterns, while the LSTM identifies more complex multi-word interactions that seem more relevant to the target aroma value. Many SIS from the CNN are simply phrases with universally-positive sentiment, indicating this model may be less able to distinguish between positive sentiment toward aroma vs. other aspects such as appearance or palate.

2.8.2 Understanding Differences Between MNIST Classifiers

We next use SIS and our clustering procedure to understand and visualize differences in features learned by two different models trained on the same MNIST digit classification task. In addition to the previously described CNN model (see Section 2.6.1), we also trained a simple multilayer perceptron (MLP) on the same task. The MLP architecture is as follows:

1. **Input**: 784-dimensional (flattened) image, all values $\in [0, 1]$
2. **Dense Layer 1**: 250 neurons, ReLU activation, and dropout probability 0.2

Table 2.5: Joint clustering of the SIS from beer reviews predicted to have positive aroma by LSTM or CNN. Dashes are used in clusters with under four unique SIS. Percentages quantify SIS per cluster stemming from the LSTM.

Cluster	**LSTM**	**SIS #1**	**SIS #2**	**SIS #3**	**SIS #4**
C_1	0%	delicious	-	-	-
C_2	0%	very nice	-	-	-
C_3	20%	rich chocolate	very rich	chocolate complex	smells rich
C_4	33%	oak chocolate	chocolate raisins raisins oak bourbon	chocolate oak	raisins chocolate
C_5	70%	complex aroma	aroma complex peaches complex	aroma complex interesting cherries	aroma complex

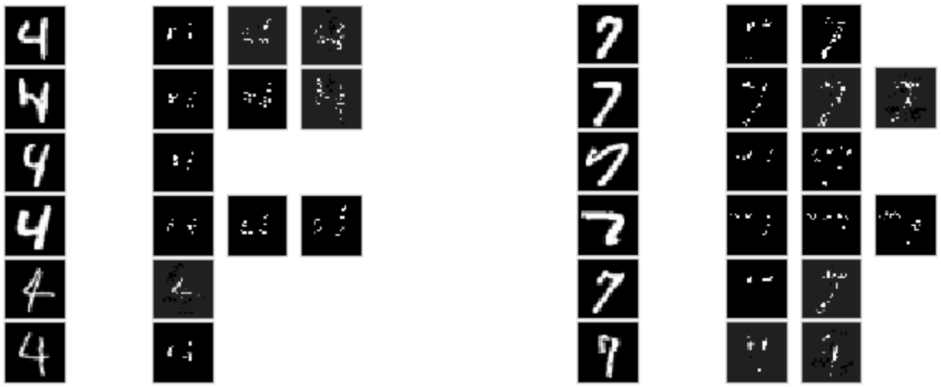
3. **Dense Layer 2**: 250 neurons, ReLU activation, and dropout probability 0.2

4. **Dense Layer 3**: 10 neurons (one per digit class), with softmax activation

As with the CNN, Adadelta (Zeiler, 2012) is used to minimize cross-entropy loss on the training set. The final MLP model achieves 99.7% accuracy on the train set and 98.3% accuracy on the test set, which is close to the performance of the CNN (see Section 2.6.1).

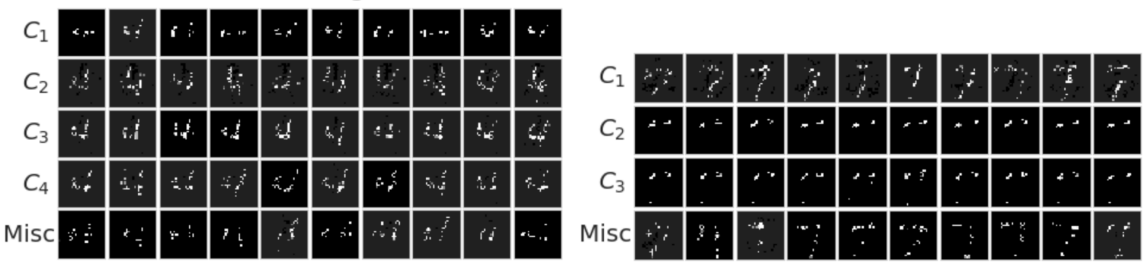
We apply the same procedure as in Section 2.6 to extract the SIS-collections from MNIST test images using the MLP. Figure 2-24 shows some examples of SIS-collections extracted for MNIST digits 4 and 8 from the MLP architecture. We also cluster the SIS-collections extracted from the MLP (as in Section 2.7.3). Clusters for two classes are shown in Figure 2-25.

To understand and visualize the differences between the features learned by each model to classify MNIST digits, we combine all SIS (from both models, for a particular class) and apply our joint SIS clustering procedure. In the resulting clustering (for digit 4 as shown in Figure 2-26), we list what percentage of the SIS in each cluster stem from the CNN vs. the MLP. Most clusters contain examples purely from a single model, indicating the two models have learned to associate different feature



(a) Digit 4 (b) Digit 7

Figure 2-24: Visualization of SIS-collections identified for MNIST digits **(a)** 4 and **(b)** 7 under the MLP model. For each class, six examples were chosen randomly. For each example, we show the original image (left) and the complete set of sufficient input subsets identified for that example (remaining images in each row). Note that each individual SIS satisfies $f(\mathbf{x}_S) \geq \tau$ for that class. Compare to the SIS extracted from the CNN architecture (Figure 2-17).



(a) Digit 4 (b) Digit 7

Figure 2-25: Clustering all the SIS identified by our method on digits **(a)** 4 and **(b)** 7 under the MLP model (Section 2.8.2). Each row contains images drawn from one cluster. The bottom row ("Misc") contains a sample of miscellaneous SIS not assigned to any cluster by DBSCAN. Compare to the SIS clustering from our CNN model (Figure 2-22).

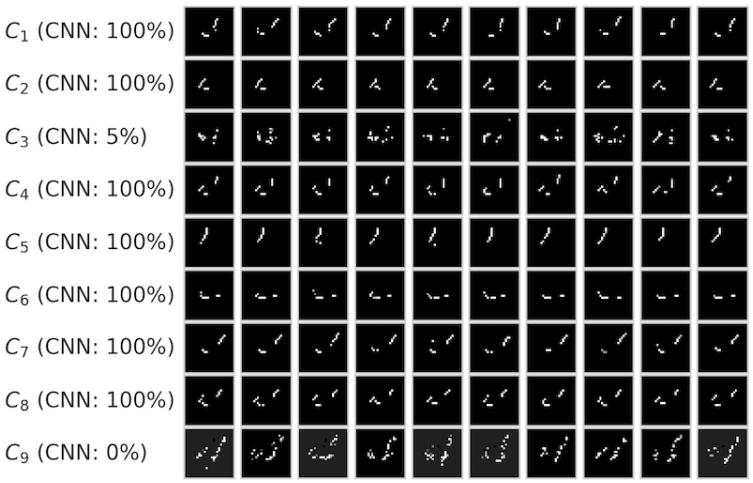


Figure 2-26: Jointly clustering the MNIST digit 4 SIS from CNN and MLP. We list the percentage of SIS in each cluster stemming from the CNN (rest from MLP).

patterns with class 4. Evidently, the CNN bases its confidence primarily on spatially-contiguous strokes comprising only a small portion of each image. Classifications by the MLP instead seem to be based on pixels located throughout the digit, demonstrating this model relies more on the global shape of the handwriting. Thus, this result suggests that the CNN may be more susceptible to mistaking other (non-digit) handwritten characters for 4s if they happen to share some of the same strokes.

Chapter 3

Overinterpretation by Deep Learning Classifiers

Well-founded decisions by machine learning (ML) systems are critical for high-stakes applications such as autonomous vehicles and medical diagnosis. Pathologies in models and their respective training datasets can result in unintended behavior during deployment if the systems are confronted with novel situations. For example, a medical image classifier for cancer detection attained high accuracy in benchmark test data, but was found to base decisions upon presence of rulers in an image (present when dermatologists already suspected cancer) (Patel, 2017). We define model *overinterpretation* to occur when a classifier finds strong class-evidence in regions of an image that contain no semantically salient features. Overinterpretation is related to overfitting, but overfitting can be diagnosed via reduced test accuracy. Overinterpretation can stem from true statistical signals in the underlying dataset distribution that happen to arise from particular properties of the data source (e.g., dermatologists' rulers). Thus, overinterpretation can be harder to diagnose as it admits decisions that are made by statistically valid criteria, and models that use such criteria can excel at benchmarks. We demonstrate overinterpretation occurs with unmodified subsets of the original images. In contrast to *adversarial examples* that modify images with extra information, overinterpretation is based on real patterns already present in the training data that also generalize to the test distribution. Hidden statistical signals

of benchmark datasets can result in models that overinterpret or do not generalize to new data from a different distribution. Computer vision (CV) research relies on datasets like CIFAR-10 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015) to provide standardized performance benchmarks. Here, we analyze the overinterpretation of popular CNN architectures on these benchmarks to characterize pathologies.

Revealing overinterpretation requires a systematic way to identify which features are used by a model to reach its decision. Feature attribution is addressed by a large number of interpretability methods, although they propose differing explanations for the decisions of a model. One natural explanation for image classification lies in the set of pixels that is sufficient for the model to make a confident prediction, even in the absence of information about the rest of the image. In the example of the medical image classifier for cancer detection, one might identify the pathological behavior by finding pixels depicting the ruler alone suffice for the model to confidently output the same classifications. In Chapter 2, we introduced Sufficient Input Subsets (SIS) as a framework to help humans interpret the decisions of black-box models. An SIS subset is a minimal subset of features (e.g., pixels) that suffices to yield a class probability above a certain threshold with all other features masked.

Here, we demonstrate that classifiers trained on CIFAR-10 and ImageNet can base their decisions on SIS subsets that contain few pixels and lack human understandable semantic content. Nevertheless, these SIS subsets contain statistical signals that generalize across the benchmark data distribution, and we are able to train classifiers on CIFAR-10 images missing 95% of their pixels and ImageNet images missing 90% of their pixels with minimal loss of test accuracy. Thus, these benchmarks contain inherent statistical shortcuts that classifiers optimized for accuracy can learn to exploit, instead of learning more complex *semantic* relationships between the image pixels and the assigned class label. While recent work suggests adversarially robust models base their predictions on more semantically meaningful features (Ilyas et al., 2019), we find these models suffer from overinterpretation as well. As we subsequently show, overinterpretation is not only a conceptual issue, but can actually harm overall classifier performance in practice. We find model ensembling and input dropout partially

mitigate overinterpretation, increasing the semantic content of the resulting SIS subsets. However, this mitigation is not a substitute for better training data, and we find that overinterpretation is a statistical property of common benchmarks. Intriguingly, the number of pixels in the SIS rationale behind a particular classification is often indicative of whether the image is correctly classified.

It may seem unnatural to use an interpretability method that produces feature attributions that look uninterpretable. However, we do not want to bias extracted rationales towards human visual priors when analyzing a model's pathologies, but rather faithfully report the features used by a model. To our knowledge, this is the first analysis showing one can extract nonsensical features from CIFAR-10 and ImageNet that intuitively should be insufficient or irrelevant for a confident prediction, yet are alone sufficient to train classifiers with minimal loss of performance. Our contributions include:

- We discover the pathology of overinterpretation and find it is a common failure mode of ML models, which latch onto non-salient but statistically valid signals in datasets (Section 3.3.1).

- We introduce Batched Gradient SIS, a new masking algorithm to scale SIS to high-dimensional inputs and apply it to characterize overinterpretation on ImageNet (Section 3.2.2).

- We provide a pipeline for detecting overinterpretation by masking over 90% of each image, demonstrating minimal loss of test accuracy, and establish lack of saliency in these patterns through human accuracy evaluations (Sections 3.2.3, 3.3.2, and 3.3.3).

- We show that the size of the feature set that the model relies on is inversely correlated with whether it is semantically meaningful via human evaluation (Section 3.3.3).

- We show misclassifications often rely on smaller and more spurious feature subsets suggesting overinterpretation is a serious practical issue (Section 3.3.4).

- We identify two strategies for mitigating overinterpretation (Section 3.3.5). We demonstrate that overinterpretation is caused by spurious statistical signals in training data, and thus training data must be carefully curated to eliminate overinterpretation artifacts.

Code for the experiments in this chapter is available at: `https://github.com/gifford-lab/overinterpretation`.

3.1 Related Work

While existing work has demonstrated numerous distinct flaws in deep image classifiers, we demonstrate a new distinct flaw, overinterpretation, previously undocumented in the literature. There has been substantial research on understanding dataset bias in CV (Torralba and Efros, 2011; Tommasi et al., 2017) and the fragility of image classifiers deployed outside benchmark settings. We extend our work on sufficient input subsets (SIS) (Chapter 2) with the introduction of the Batched Gradient SIS method, and we use this method to show that ImageNet sufficient input subset pixels for training and testing often exist at image borders. Many alternative interpretability methods also aim to understand models by extracting *rationales* (pixel-subsets) that provide positive evidence for a class (Fong et al., 2019; Samek et al., 2016; Agarwal and Nguyen, 2020; Dhurandhar et al., 2018), and we adopt SIS throughout this work as a particularly straightforward method for producing such rationales. This prior work (including SIS) is limited to understanding models and does not use the enhanced understanding of models to identify overinterpretation. We contrast the issue of overinterpretation against other previously known model flaws below:

- Image classifiers have been shown to be fragile when objects from one image are transplanted in another image (Rosenfeld et al., 2018), and can be biased by object context (Shetty et al., 2019; Singh et al., 2020b). In contrast, overinterpretation differs because we demonstrate that highly sparse, unmodified subsets of pixels

in images suffice for image classifiers to make the same predictions as on the full images.

- Lapuschkin et al. (2019) demonstrate that DNNs can learn to rely on spurious signals in datasets, including source tags and artificial padding, but which are still human-interpretable. In contrast, the patterns we identify are minimal collections of pixels in images that are semantically meaningless to humans (they do not comprise human-interpretable parts of images). We demonstrate such patterns generalize to the test distribution suggesting they arise from degenerate signals in popular benchmarks, and thus models trained on these datasets may fail to generalize to real-world data.

- CNNs in particular have been conjectured to pick up on localized features like texture instead of more global features like object shape (Gatys et al., 2017; Geirhos et al., 2019). Brendel and Bethge (2019) show CNNs trained on natural ImageNet images may rely on local features and, unlike humans, are able to classify texturized images, suggesting ImageNet alone is insufficient to force DNNs to rely on more causal representations. Our work demonstrates another source of degeneracy of popular image datasets, where sparse, unmodified subsets of training images that are meaningless to humans can enable a model to generalize to test data. We provide one explanation for why ImageNet-trained models may struggle to generalize to out-of-distribution data.

- Geirhos et al. (2018) discover that DNNs trained on distorted images fail to generalize as well as human observers when trained under image distortions. In contrast, overinterpretation reveals a different failure mode of DNNs, whereby models latch onto spurious but statistically valid sets of features in undistorted images. This phenomenon can limit the ability of a DNN to generalize to real-world data even when trained on natural images.

- Other work has shown deep image classifiers can make confident predictions on nonsensical patterns (Nguyen et al., 2015), and the susceptibility of DNNs to ad-

versarial examples or synthetic images has been widely studied (Goodfellow et al., 2015; Madry et al., 2018; Ilyas et al., 2019). However, these adversarial examples synthesize artificial images or modify real images with auxiliary information. In contrast, we demonstrate overinterpretation of unmodified subsets of actual training images, indicating the patterns are already present in the original dataset. We further demonstrate that such signals in training data actually generalize to the test distribution and that adversarially robust models also suffer from overinterpretation.

- Hooker et al. (2019) found sparse pixel subsets suffice to attain high classification accuracy on popular image classification datasets, but evaluate interpretability methods rather than demonstrate spurious features or discover overinterpretation.

- Ghorbani et al. (2019) introduce principles and methods for human-understandable concept-based explanations of ML models. In contrast, overinterpretation differs because the features we identify are semantically meaningless to humans, stem from single images, and are not aggregated into interpretable concepts. The existence of such subsets stemming from unmodified subsets of images suggests degeneracies in the underlying benchmark datasets and failures of modern CNN models to rely on more robust and interpretable signals in training datasets.

- Geirhos et al. (2020) discuss the general problem of "shortcut learning" but do not recognize that 5% (CIFAR-10) or 10% (ImageNet) spurious pixel-subsets are statistically valid signals in these datasets, nor characterize pixels that provide sufficient support and lead to overinterpretation.

- In natural language processing (NLP), Feng et al. (2018) explored model pathologies using a similar technique, but did not analyze whether the semantically spurious patterns relied on are a statistical property of the dataset. Other work has demonstrated the presence of various spurious statistical shortcuts in major NLP benchmarks, showing this problem is not unique to CV (Niven and Kao, 2019).

3.2 Methods

3.2.1 Datasets and Models

CIFAR-10 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015) have become two of the most popular image classification benchmarks. Most image classifiers are evaluated by the CV community based on their accuracy in one of these benchmarks. We also use the CIFAR-10-C dataset (Hendrycks and Dietterich, 2019) to evaluate the extent to which our CIFAR-10 models can generalize to out-of-distribution (OOD) data. CIFAR-10-C contains variants of CIFAR-10 test images altered by various corruptions (e.g., Gaussian noise, motion blur). Where computing sufficient input subsets on CIFAR-10-C images, we use a uniform random sample of 2000 images across the entire CIFAR-10-C set. Additional results on CIFAR-10.1 v6 (Recht et al., 2018) are presented in Table B.4. We use the ILSVRC2012 ImageNet dataset (Russakovsky et al., 2015).

For CIFAR-10, we explore three common CNN architectures: a deep residual network with depth 20 (ResNet20) (He et al., 2016a), a v2 deep residual network with depth 18 (ResNet18) (He et al., 2016b), and VGG16 (Simonyan and Zisserman, 2015). We train these networks using cross-entropy loss optimized via SGD with Nesterov momentum (Sutskever et al., 2013) and employ standard data augmentation strategies (He et al., 2016b) (Appendix B.2). After training many CIFAR-10 networks individually, we construct four different ensemble classifiers by grouping various networks together. Each ensemble outputs the average prediction over its member networks (specifically, the arithmetic mean of their logits). For each of three architectures, we create a corresponding homogeneous ensemble by individually training five networks of that architecture. Each network has a different random initialization, which suffices to produce substantially different models despite having been trained on the same data (Osband et al., 2016). Our fourth ensemble is heterogeneous, containing all 15 networks (5 replicates of each of 3 distinct CNN architectures).

For ImageNet, we use a pre-trained Inception v3 model (Szegedy et al., 2016) that achieves 22.55% and 6.44% top-1 and top-5 error (Paszke et al., 2019) Additional

results from an ImageNet ResNet50 are presented in Appendix B.6.

3.2.2 Discovering Sufficient Features

CIFAR-10. We interpret the feature patterns learned by CIFAR-10 CNNs using the Sufficient Input Subsets (SIS) procedure, which produces rationales (SIS subsets) of a black-box model's decision-making (introduced in Chapter 2). SIS subsets are minimal subsets of input features (pixels) whose values alone suffice for the model to make the same decision as on the original input. Let $f_c(x)$ denote the probability that an image x belongs to class c. An SIS subset S is a minimal subset of pixels of x such that $f_c(x_S) \geq \tau$, where τ is a prespecified confidence threshold and x_S is a modified input in which all information about values outside S are masked. We mask pixels by replacement with the mean value over all images (equal to zero when images have been normalized), which is presumably least informative to a trained classifier (Section 2.2). SIS subsets are found via a local backward selection algorithm applied to the function giving the confidence of the predicted (most likely) class.

ImageNet. We scale the SIS backward selection procedure to ImageNet with the introduction of Batched Gradient SIS, a gradient-based method to find sufficient input subsets on high-dimensional inputs. The sufficient input subsets discovered by Batched Gradient SIS are guaranteed to be sufficient, but may be larger than those discovered by the original exhaustive SIS algorithm. Here we find small SIS subsets with Batched Gradient SIS (Figure B-15). Rather than separately masking every remaining pixel at each iteration to find the pixel whose masking least reduces f, we use the gradient of f with respect to the input pixels $\mathbf{x}$ and mask M, $\nabla_M f(\mathbf{x} \odot (1 - M))$, to order pixels (via a single backward pass). Instead of masking only one pixel per iteration, we mask larger subsets of $k \geq 1$ pixels per iteration. Given p input features, our Batched Gradient FindSIS procedure finds each SIS subset in $\mathcal{O}(\frac{p}{k})$ evaluations of ∇f (as opposed to $\mathcal{O}(p^2)$ evaluations of f in FindSIS (Section 2.2)). The complete Batched Gradient SIS algorithm is presented in Appendix B.1.

3.2.3 Detecting Overinterpretation

We produce sparse variants of all train and test set images retaining 5% (CIFAR-10) or 10% (ImageNet) of pixels in each image. Our goal is to identify sparse pixel-subsets that contain feature patterns the model identifies as strong class-evidence as it classifies an image. We identify pixels to retain based on sorting by SIS BackSelect (Section 2.2, CIFAR-10) or our Batched Gradient BackSelect procedure (ImageNet). These backward selection (BS) pixel-subset images contain the final pixels (with their same RGB values as in the original images) while all other pixels' values are replaced with zero. Note that we apply backward selection to the function giving the confidence of the *predicted* class from the original model to prevent adding information about the true class for misclassified images, and we use the true labels for training/evaluating models on pixel-subsets. As backward selection is applied locally on each image, the specific pixels retained differ across images.

We train new classifiers on solely these pixel-subsets of training images and evaluate accuracy on corresponding pixel-subsets of test images to determine whether such pixel-subsets are statistically valid for generalization in the benchmark. We use the same training setup and hyperparameters (Section 3.2.1) without data augmentation of training images (results with data augmentation in Table B.1). We consider a model to overinterpret its input when these signals can generalize to test data but lack semantic meaning (Section 3.2.4).

3.2.4 Human Classification Benchmark

To evaluate whether sparse pixel-subsets of images can be accurately classified by humans, we asked four participants to classify images containing various degrees of masking. We randomly sampled 100 images from the CIFAR-10 test set (10 images per class) that were correctly and confidently ($\geq$ 99% confidence) classified by our models, and for each image, kept only 5%, 30%, or 50% of pixels as ranked by backward selection (all other pixels masked). Backward selection image subsets are sampled across our three models. Since larger subsets of pixels are by construction

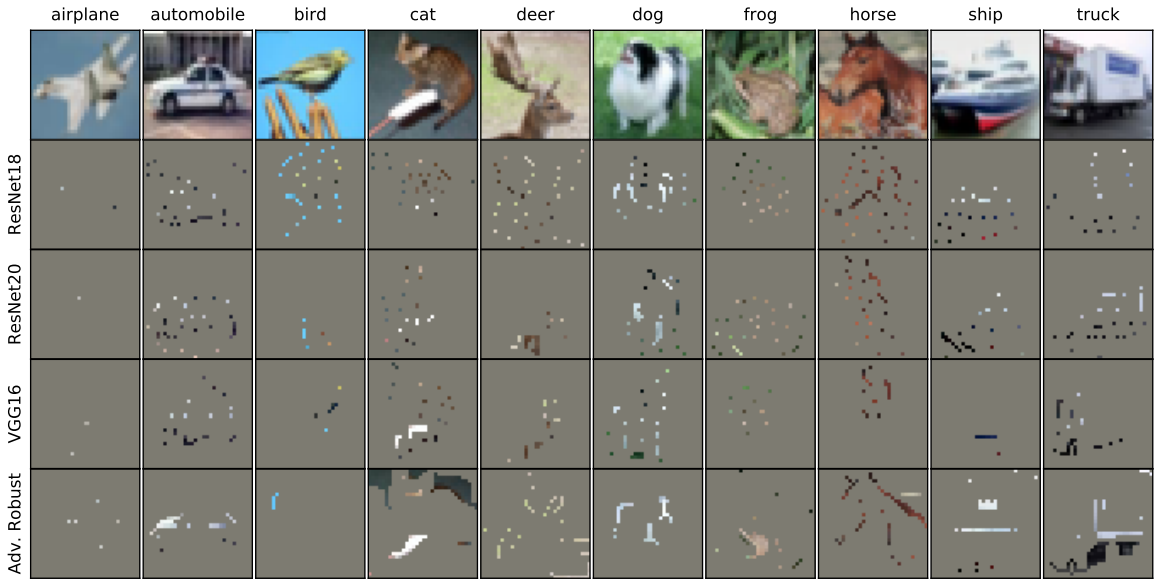


Figure 3-1: Sufficient input subsets (SIS) for a sample of CIFAR-10 test images (top). Each SIS image shown below is classified by the respective model with $\geq 99\%$ confidence.

supersets of smaller subsets identified by the same model, we presented each batch of 100 images in order of increasing subset size and shuffled the order of images within each batch. Users were asked to classify each of the 300 images as one of the 10 classes in CIFAR-10 and were not provided training images. The same task was given to each user (and is shown in Appendix B.5).

3.3 Results

3.3.1 CNNs Classify Images Using Spurious Features

CIFAR-10. Figure 3-1 shows example SIS subsets (threshold 0.99) from CIFAR-10 test images (additional examples in Appendix B.3). These SIS subset images are confidently and correctly classified by each model with $\geq 99\%$ confidence toward the predicted class. We observe these SIS subsets are highly sparse and the average SIS size at this threshold is $< 5\%$ of each image (Figure 3-2), suggesting these CNNs confidently classify images that appear nonsensical to humans (Section 3.3.3), leading to concern about their robustness and generalizability. We also find that SIS size can

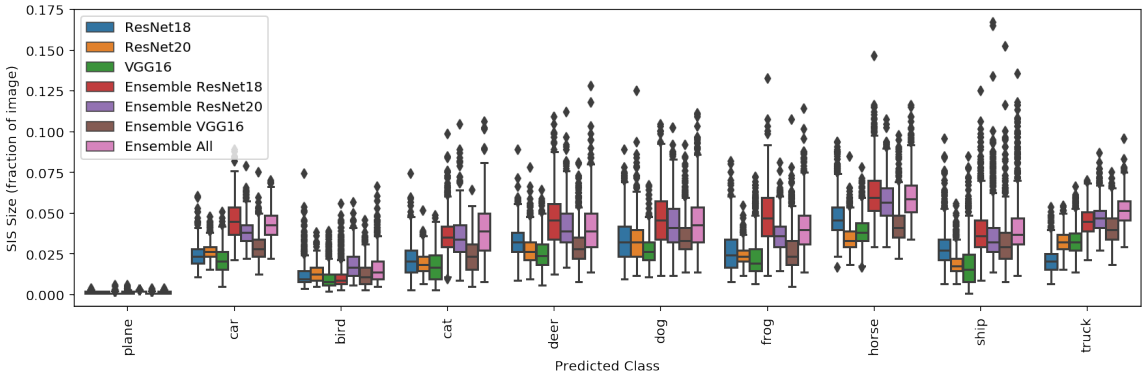
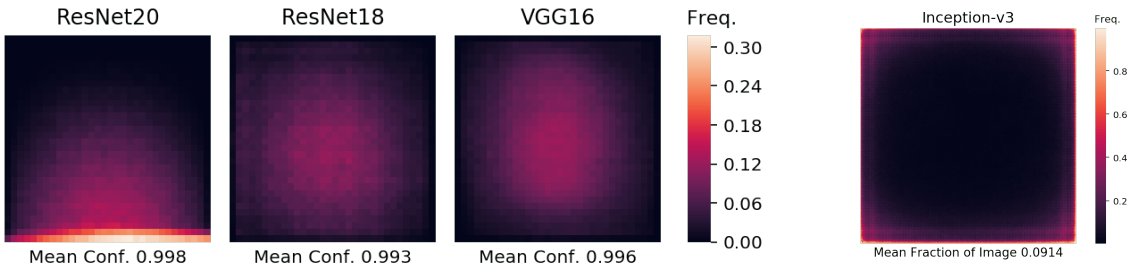


Figure 3-2: Distribution of SIS size per predicted class by CIFAR-10 models computed on all CIFAR-10 test set images classified with $\geq$ 99% confidence (SIS confidence threshold 0.99).

differ significantly by predicted class (Figure 3-2).

We retain 5% of pixels in each image using local backward selection and mask the remaining 95% with zeros (Section 3.2.3) and find models trained on full images classify these pixel-subsets as accurately as full images (Table 3.1). Figure 3-3a shows the pixel locations and confidence of these 5% pixel-subsets across all CIFAR-10 test images. We found the concentration of pixels on the bottom border for ResNet20 is a result of tie-breaking during SIS backward selection (Appendix B.4). Moreover, the CNNs are more confident on these pixels subsets than on full images: the mean drop in confidence for the predicted class between original images and these 5% subsets is -0.035 (std dev. $= 0.107$), -0.016 (0.094), and -0.012 (0.074) computed over all CIFAR-10 test images for our ResNet20, ResNet18, and VGG16 models, respectively, suggesting severe overinterpretation (negative values imply greater confidence on the 5% subsets). We find pixel-subsets chosen via backward selection are significantly more predictive than equally large pixel-subsets chosen uniformly at random from each image (Table 3.1).

We also find SIS subsets confidently classified by one model do not transfer to other models. For instance, 5% pixel-subsets derived from CIFAR-10 test images using one ResNet18 model (which classifies them with 94.8% accuracy) are only classified with 25.8%, 29.2%, and 27.5% accuracy by another ResNet18 replicate, ResNet20, and VGG16 models, respectively, suggesting there exist many different statistical



(a) CIFAR-10 (b) ImageNet

Figure 3-3: Heatmaps of pixel locations comprising pixel-subsets. Frequency indicates fraction of subsets containing each pixel. **(a)** 5% pixel-subsets across CIFAR-10 test set for each model. Mean confidence indicates confidence on 5% pixel-subsets. **(b)** Sufficient input subsets (confidence threshold 0.9) across ImageNet validation images from Inception v3.

patterns that a flexible model might learn to rely on, and thus CIFAR-10 image classification remains a highly underdetermined problem. Training classifiers that make predictions for the right reasons may require clever regularization strategies and architecture design to ensure models favor salient features over spurious pixel subsets.

While recent work has suggested semantics can be better captured by models that are robust to adversarial inputs that fool standard neural networks via human-imperceptible modifications to images (Madry et al., 2018; Santurkar et al., 2019), we explore a wide residual network that is adversarially robust for CIFAR-10 classification (Madry et al., 2018) and find evidence of overinterpretation (Figure 3-1). This finding suggests adversarial robustness alone does not prevent models from overinterpreting spurious signals in CIFAR-10.

We also ran Batched Gradient SIS on CIFAR-10 and found edge-heavy sufficient input subsets for CIFAR-10 (Appendix B.4). These heatmap differences are a result of the different valid equivalent sufficient input subsets found by the two SIS discovery algorithms. However, since all sufficient input subsets are validated with a model and guaranteed to be sufficient for classification at the specified threshold, the heatmaps are accurate depictions of what is sufficient for the model to classify images at the threshold. Overinterpretation is independent of the SIS algorithm used because both algorithms produce human-uninterpretable sufficient subsets as shown

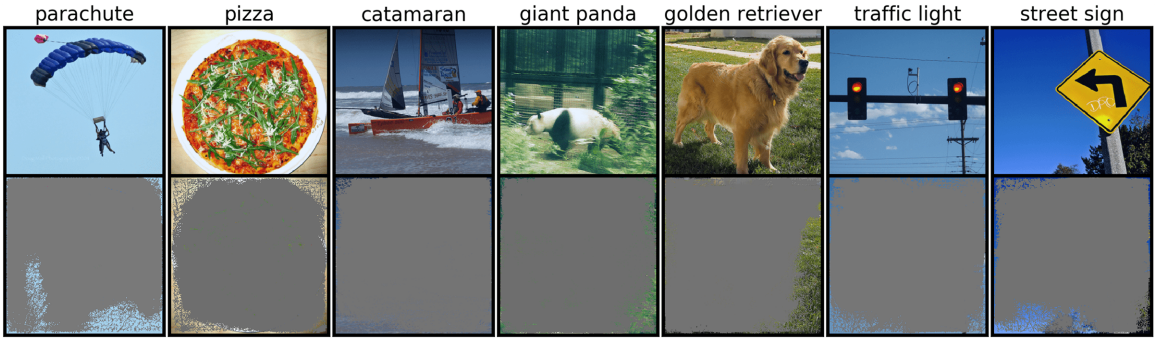


Figure 3-4: Sufficient input subsets (threshold 0.9) for example ImageNet validation images. The bottom row shows the corresponding images with all pixels outside of each SIS subset masked but are still classified by the Inception v3 model with $\geq 90\%$ confidence.

in the examples.

ImageNet. We find models trained on ImageNet images suffer from severe overinterpretation. Figure 3-4 shows example SIS subsets (threshold 0.9) found via Batched Gradient SIS on images confidently classified by the pre-trained Inception v3 (additional examples in Figures B-12–B-14). These SIS subsets appear visually nonsensical, yet the network classifies them with $\geq 90\%$ confidence. We find SIS pixels are concentrated outside of the actual object that determines the class label. For example, in the "pizza" image, the SIS is concentrated on the shape of the plate and the background table, rather than the pizza itself, suggesting the model could generalize poorly on images containing different circular items on a table. In the "giant panda" image, the SIS contains bamboo, which likely appeared in the collection of ImageNet photos for this class. In the "traffic light" and "street sign" images, the SIS consists of pixels in sky, suggesting that autonomous vehicle systems that may depend on these models should be carefully evaluated for overinterpretation pathologies.

Figure 3-3b shows SIS pixel locations from a random sample of 1000 ImageNet validation images. We find concentration along image borders, suggesting the model relies heavily on image backgrounds and suffers from severe overinterpretation. This is a serious problem as objects determining ImageNet classes are often located near image centers, and thus this network fails to focus on salient features. We found the

mean fraction of an image required for classification with $\geq 90\%$ confidence is only 0.0914, and mean SIS size differs significantly by predicted class (Figure B-16).

3.3.2 Sparse Subsets are Real Statistical Patterns

The overconfidence of CNNs for image classification (Guo et al., 2017) may lead one to wonder whether the observed overconfidence on semantically meaningless SIS subsets is an artifact of calibration rather than true statistical signals in the dataset. We train models on 5% pixel-subsets of CIFAR-10 training images found via backward selection (Section 3.2.3). We find models trained solely on these pixel-subsets can classify corresponding test image pixel-subsets with minimal accuracy loss compared to models trained on full images (Table 3.1), and thus these 5% pixel-subsets are valid statistical signals in training images that generalize to the test distribution. As a baseline to the 5% pixel-subsets identified by backward selection, we create variants of all images where the 5% pixel-subsets are selected at random from each image (rather than by backward selection) and use the same random pixel-subsets for training each new model. Models trained on random subsets have significantly lower test accuracy compared to models trained on 5% pixel-subsets from backward selection (Table 3.1). We observe, however, that random 5% subsets of images still capture enough signal to predict roughly 5 times better than blind guessing, but do not capture nearly enough information for models to make accurate predictions.

We found that the 5% backward selection pixel-subsets did not contain model-specific features, and thus reflected valid predictive signals regardless of the model architecture employed for subset discovery. Our hypothesis was that 5% pixel-subsets discovered with one architecture would provide robust performance when used to train and evaluate a second architecture. We found this hypothesis supported for all six pairs of subset discovery and train-test architectures evaluated (Table B.2). These results demonstrate that the highly sparse subsets found via backward selection offer a valid predictive signal in the CIFAR-10 benchmark exploited by models to attain high test accuracy.

We observe similar results on ImageNet. Inception v3 trained on 10% pixel-subsets

of ImageNet training images achieves 71.4% top-1 accuracy (mean over 5 runs) on the corresponding pixel-subset ImageNet validation set (Table B.7). Additional ImageNet results for Inception v3 and ResNet50, including training and evaluation on random pixel-subsets and pixel-subsets of different architectures, are provided in Table B.7.

3.3.3 Humans Struggle to Classify Sparse Subsets

We find a strong correlation between the fraction of unmasked pixels in each image and human classification accuracy ($R^2 = 0.94$, Figure B-11). Human accuracy on 5% pixel-subsets of CIFAR-10 images (mean = 19.2%, std dev = 4.8%, Table B.6) is significantly lower than on original, unmasked images (roughly 94% (Karpathy, 2011)), though greater than random guessing, presumably due to correlations between labels and features such as color (e.g., blue sky suggests airplane, ship, or bird).

However, CNNs (even when trained on full images and achieve accuracy on par with human accuracy on full images) classify these sparse image subsets with very high accuracy (Table 3.1), indicating benchmark images contain statistical signals that are not salient to humans. Models solely trained to minimize prediction error may thus latch onto these signals while still accurately generalizing to test data, but may behave counterintuitively when fed images from a different source that does not share these exact statistics. The strong correlation between the size of CIFAR-10 pixel-subsets and the corresponding human classification accuracy suggests larger subsets contain more semantically salient content. Thus, a model whose decisions have larger corresponding SIS subsets presumably exhibits less overinterpretation than one with smaller SIS subsets, as we investigate in Section 3.3.4.

3.3.4 SIS Size is Related to Model Accuracy

Given that smaller SIS contain fewer salient features according to human classifiers, models that justify their classifications based on sparse SIS subsets may be limited in terms of attainable accuracy, particularly in out-of-distribution settings. Here, we investigate the relationship between a single model's predictive accuracy and the

size of the SIS subsets in which it identifies class-evidence. We draw no conclusions between models as they are uncalibrated (additional results of SIS from calibrated models are presented in Appendix B.4). For each of our three classifiers, we compute the average SIS size increase for correctly classified images as compared to incorrectly classified images (expressed as a percentage). We find SIS subsets of correctly classified images are consistently significantly larger than those of misclassified images at all SIS confidence thresholds for both CIFAR-10 test images (Figure 3-5) and CIFAR-10-C OOD images (Figure B-3). This is especially striking given model confidence is uniformly lower on the misclassified inputs (Figure B-4). Lower confidence would normally imply a larger SIS subset at a given confidence level, as one expects fewer pixels can be masked before the model's confidence drops below the SIS threshold. Thus, we can rule out overall model confidence as an explanation of the smaller SIS of misclassified images. This result suggests the sparse SIS subsets we identify are not just a curiosity, but may be leading to poor generalization on real images.

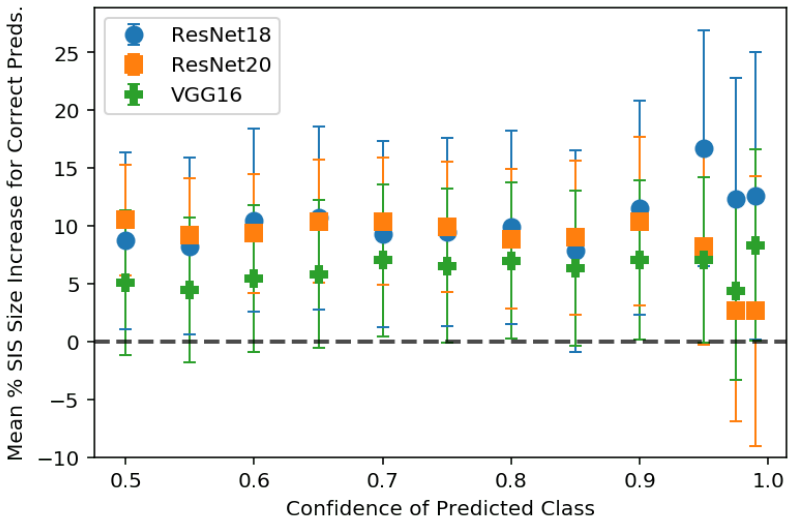


Figure 3-5: Percentage increase in mean SIS size of correctly classified compared to misclassified CIFAR-10 test images. Positive values indicate larger mean SIS size for correctly classified images. Error bars indicate 95% confidence interval for the difference in means.

3.3.5 Mitigating Overinterpretation

Ensembling. Model ensembling is known to improve classification performance (Goh et al., 2001; Ju et al., 2018). As we found pixel-subset size to be strongly correlated with human pixel-subset classification accuracy (Section 3.3.3), our metric for measuring how much ensembling may alleviate overinterpretation is the increase in SIS subset size. We find ensembling uniformly increases test accuracy as expected but also increases the SIS size (Figure 3-6), hence mitigating overinterpretation.

We conjecture the cause of both the increase in the accuracy and SIS size for ensembles is the same. We observe that SIS subsets are generally not transferable from one model to another — i.e., an SIS for one model is rarely an SIS for another (Section 3.3.1). Thus, different models rely on different independent signals to arrive at the same prediction. An ensemble bases its prediction on multiple such signals, increasing predictive accuracy and SIS subset size by requiring simultaneous activation of multiple independently trained feature detectors. We find SIS subsets of the ensemble are larger than the SIS of its individual members (examples in Figure B-2).

Input Dropout. We apply input dropout (Srivastava et al., 2014) to both train and test images. We retain each input pixel with probability $p = 0.8$ and set the values of dropped pixels to zero. We find a small decrease in CIFAR-10 test accuracy for models regularized with input dropout though find a significant ($\sim 6\%$) increase in OOD test accuracy on CIFAR-10-C images (Table 3.1, Figure B-5). Figure 3-6 shows a corresponding increase in SIS subset size for these models, suggesting input dropout applied at train and test time helps to mitigate overinterpretation. We conjecture that random dropout of input pixels disrupts spurious signals that lead to overinterpretation.

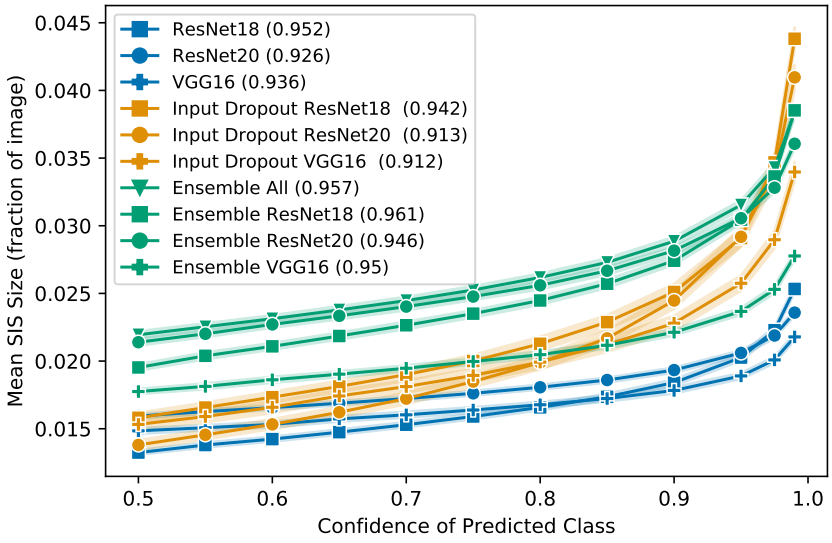


Figure 3-6: Mean SIS size on CIFAR-10 test images as SIS threshold varies. SIS size indicates fraction of pixels necessary for model to make the same prediction at each confidence threshold. Model accuracies are shown in the legend. 95% confidence intervals are shaded around each mean.

Table 3.1: Accuracy of CIFAR-10 classifiers trained and evaluated on full images, 5% backward selection (BS) pixel-subsets, and 5% random pixel-subsets. Where possible, accuracy is reported as mean ± standard deviation (%) over five runs. For training on BS subsets, we run BS on all images for a single model of each type and average over five models trained on these subsets. Additional results on CIFAR-10.1 are presented in Table B.4.

Model	Train On	Evaluate On	CIFAR-10 Test Acc.	CIFAR-10-C Acc.
ResNet20	Full Images	Full Images	92.52 ± 0.09	69.44 ± 0.52
		5% BS Subsets	92.48	70.65
		5% Random	9.98 ± 0.03	10.02 ± 0.01
	5% BS Subsets	5% BS Subsets	92.49 ± 0.02	70.58 ± 0.03
	5% Random	5% Random	50.25 ± 0.19	44.04 ± 0.33
	Input Dropout (Full)	Input Dropout (Full)	91.02 ± 0.25	75.46 ± 0.74
ResNet18	Full Images	Full Images	95.17 ± 0.21	75.08 ± 0.20
		5% BS Subsets	94.76	75.15
		5% Random	10.08 ± 0.15	10.08 ± 0.07
	5% BS Subsets	5% BS Subsets	94.96 ± 0.04	75.25 ± 0.05
	5% Random	5% Random	51.27 ± 0.82	45.24 ± 0.45
	Input Dropout (Full)	Input Dropout (Full)	94.15 ± 0.26	80.35 ± 0.39
VGG16	Full Images	Full Images	93.69 ± 0.12	74.14 ± 0.45
		5% BS Subsets	93.27	73.95
		5% Random	10.02 ± 0.18	9.97 ± 0.18
	5% BS Subsets	5% BS Subsets	92.60 ± 0.08	73.27 ± 0.18
	5% Random	5% Random	53.66 ± 1.96	46.88 ± 1.27
	Input Dropout (Full)	Input Dropout (Full)	91.09 ± 0.15	80.43 ± 0.24
Ensemble (ResNet18)	Full Images	Full Images	96.07	77.00
		5% Random	9.98	10.01

Chapter 4

Computational Design of Peptide Vaccines with n-times Coverage

In this chapter, we introduce the EvalVax and OptiVax framework for the computational evaluation and design of peptide vaccines. In contrast to the traditional vaccine strategies that deliver protein subunits or contain live-attenuated pathogen to elicit antibody responses, peptide vaccines are focused on activation of T cells against specific short antigenic fragments. Peptide vaccines deliver short peptide epitopes that are presented on the surface of antigen presenting cells (APCs) by Major Histocompatibility Complex (MHC) class I or class II molecules that are recognized by T cells to engage a cellular immune response against the peptides. Peptide vaccines can mitigate potential allergenic responses that could be triggered by the greater diversity of antigens found in larger protein subunits (Li et al., 2014). In addition, peptide vaccines can focus immune responses toward epitopes that are conserved across pathogenic variants, unlike larger protein vaccines that may induce immune responses against mutable antigens. Peptide vaccines are in development for diseases including HIV (Arunachalam et al., 2020), HPV (Kenter et al., 2009), and malaria (Nardin et al., 2000), as well as in cancer where peptide vaccines can deliver personalized neoantigens that are present in patients' tumors (Hu et al., 2018).

An important consideration in the development of peptide vaccines is the selection of a compact vaccine payload. Ideally, the peptides should provide broad pop-

ulation coverage considering the MHC allele diversity in the human population, and for an optimal immune response, the vaccine should activate both $CD8^+$ and $CD4^+$ T cells (Zhang et al., 2009). For pathogenic targets, peptides should be conserved across pathogenic variants to provide pan-variant immunity. Importantly, our evaluation and design framework introduces the metric of *n-times population coverage*, in which each individual in the population is ideally covered by at least n vaccine peptide-HLA hits. We define a *peptide-HLA hit* as a (peptide, HLA allele) pair for which the peptide is predicted to be displayed by the HLA allele and immunogenic in the individual. While it might be assumed that an individual will be covered by a vaccine if they display a single peptide, our n-times coverage framework provides the following advantages over 1-times coverage to increase the robustness of the vaccine:

1. When an individual displays multiple peptides, their immune system activates and expands more than one set of T cell clonotypes that are poised to fight viral infection (Sekine et al., 2020; Schultheiß et al., 2020; Grifoni et al., 2020b).

2. The peptides that are immunogenic vary from one individual to another, and thus having multiple peptides displayed increases the probability at least one will be strongly immunogenic (Croft et al., 2019).

3. If a virus evolves and changes its peptide composition, using multiple peptides reduces the chance of viral escape (Wibmer et al., 2021).

For a peptide to be effective in a vaccine to induce cellular immunity it must first bind within the groove of a MHC class I or class II molecule, and secondly, it must be immunogenic, that is, it must activate T cells when it is bound by MHC proteins and displayed. Immunogenicity is therefore dependent on the sequence of the peptide displayed, the protein sequences of an individual's MHC genes, and the affinity between the two. A challenge for the design of peptide vaccines is the diversity of human MHC gene alleles that each have specific preferences for the peptide sequences they will display. The Human Leukocyte Antigen (HLA) loci, located within the MHC, encode the HLA class I and class II molecules; an individual's HLA type describes

the alleles they carry at each three classical class I loci (HLA-A, HLA-B, and HLA-C) and three class II loci (HLA-DR, HLA-DQ, and HLA-DP).

To create effective vaccines it is necessary to consider the HLA allelic frequency in the target population, as well as linkage disequilibrium between HLA genes to discover a set of peptides that is likely to be robustly displayed. Human populations that originate from different geographies have differing frequencies of HLA alleles, and these populations exhibit linkage disequilibrium between HLA loci that result in population specific haplotype frequencies. However, previous computational peptide vaccine design and evaluation methods do not utilize the distribution of HLA haplotypes in a population, and thus cannot accurately assess the coverage provided by a vaccine. Present population-based methods, like iVax (Moise et al., 2015) and SARS-CoV-2 specific efforts (Fast et al., 2020), do not take into account haplotypes and rare HLA allelic combinations. The IEDB Population Coverage Tool (Bui et al., 2006) estimates peptide-HLA binding coverage and the distribution of peptides displayed for a given population but assumes independence between different loci and thus does not consider linkage disequilibrium.

Here, we utilize human HLA haplotype frequencies of three major populations, those self-reporting as having White, Black, or Asian ancestry, to computationally compute population coverage of SARS-CoV-2 peptides with high predicted HLA binding affinity for inclusion in MHC class I or II vaccine formulations. We examined 4,690 geographically sampled SARS-CoV-2 genomes to exclude peptides with undesired mutation rates. Recent advances in machine learning have produced models that can predict the presentation of peptides by hundreds of allelic variants of both class I and class II MHC molecules (Zeng and Gifford, 2019; Jurtz et al., 2017; O'Donnell et al., 2018; Jensen et al., 2018; Peters et al., 2020). These models are evaluated on their ability to accurately predict data unobserved during their training on hundreds of HLA alleles. Given that different models may be more or less accurate for different sequence families and can make idiosyncratic errors, we use an ensemble of models for vaccine design. We evaluate completed designs using eleven models to provide a conservative evaluation of vaccine peptide presentation. Our vaccine evaluation met-

rics (EvalVax) can be used independently of vaccine optimization to evaluate existing vaccines as we demonstrate.

Using conservative metrics of peptide-HLA binding, we find that our optimization methods (OptiVax) provide both a higher likelihood of peptide display as well as a larger number of associated peptides than other published SARS-CoV-2 peptide vaccine designs with fewer than 150 peptides. Our proposed SARS-CoV-2 MHC class I vaccine formulations provide 93.21% predicted population coverage with at least five vaccine peptide-HLA hits on average per person ($\geq$ 1 peptide 99.91%) with all vaccine peptides perfectly conserved across 4,690 geographically sampled SARS-CoV-2 genomes. Our proposed MHC class II vaccine formulations provide 97.21% predicted coverage with at least five vaccine peptide-HLA hits on average per person with all peptides having observed mutation probability ≤ 0.001. We also show OptiVax can be used to augment SARS-CoV-2 spike (S) protein vaccine designs to increase their population coverage.

The methods we present in this chapter are focused on the computational design of optimized vaccine payloads, and these payloads can be delivered through a variety of platforms, including synthesized peptides, viral vectors, nucleic acids, and can be encapsulated in nanoparticles (Paston et al., 2021; Pardi et al., 2018). In Chapter 5, we evaluate our n-times coverage vaccine design approach in a SARS-CoV-2 animal challenge study, and our vaccine is delivered by an mRNA-LNP molecule that encodes the vaccine payload.

Code and data for the experiments in this chapter are available at: `https://github.com/gifford-lab/optivax`. HLA haplotype frequency data are available at: `http://dx.doi.org/10.17632/cfxkfy9zp4.1`.

4.1 Methods

4.1.1 Framework Overview

Our approach to vaccine design uses combinatorial optimization to select peptides to achieve specific population level objectives. We provide two methods for peptide vaccine evaluation: EvalVax-Unlinked, which considers HLA allele frequencies assuming independence between HLA loci, and EvalVax-Robust, which considers haplotype frequencies and computes population coverage at minimum levels of high scoring peptide-HLA combinations per individual. We employ these evaluation methods as objective functions for peptide vaccine formulation by combinatorial optimization in OptiVax-Unlinked and OptiVax-Robust. In our framework, vaccine design proceeds by (1) starting with an initial proteome, filtering out peptides with undesired properties, (2) scoring which peptides will be presented and thus are potentially immunogenic, and (3) selecting an optimized set of candidate peptides given the frequency of HLA haplotypes or HLA alleles in a target population. Our filtering steps eliminate peptides that are predicted to be glycosylated, peptides that are expected to drift in sequence and thus cause vaccine escape, peptides that are cleaved, and peptides that are identical to peptides in the human proteome. Vaccine peptides can be drawn from the entire proteome or from specific proteins of interest. An overview of our system is shown in Figure 4-1.

Once candidate peptides are tested, any that are not immunogenic in the context of the restricting HLA allotype can be eliminated from the candidate peptide pool. Draft vaccine designs containing non-immunogenic peptides can be revised to eliminate them, and the reduced vaccine design can be re-evaluated with EvalVax to see if the design still meets performance criteria. If not, the vaccine design process can be repeated with the revised candidate pool. Immunogenicity data can be incorporated into the peptide scoring process that is used for both vaccine design and evaluation as shown in italics in Figure 4-1.

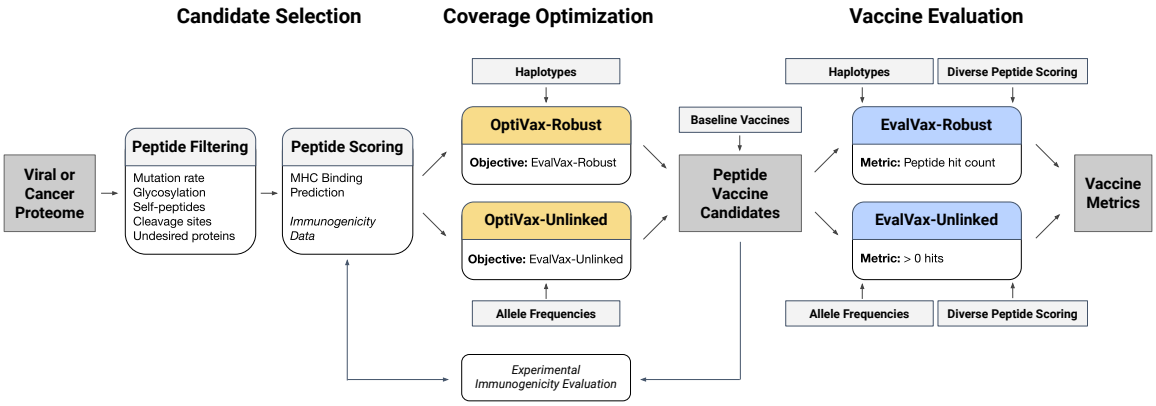


Figure 4-1: The OptiVax and EvalVax machine learning system for combinatorial vaccine optimization and evaluation. The methods can be used to design new peptide vaccines, evaluate existing vaccines, or augment existing vaccine designs. Peptides are scored by machine learning and immunogenicity data for population coverage optimization and evaluation.

4.1.2 Proteome to Candidate Vaccine Peptides

Given a target proteome as input, we identify all potential T cell epitopes for inclusion in a vaccine. We extract peptides of length 8–10 inclusive for consideration of MHC class I binding (Rist et al., 2013) and peptides of length 13–25 inclusive for class II binding (Chicz et al., 1992) by using sliding windows of each size over the entire proteome. While peptides presented by MHC class I molecules can occasionally be longer than 10 residues (Trolle et al., 2016), we conservatively limit our search to length 8–10 since MHC class I presented peptides are predominately 8–10 residues in length (Rist et al., 2013).

Using this sliding window approach, we created peptide sets from the SARS-CoV-2 (COVID-19) and SARS-CoV (Human SARS coronavirus) proteomes. SARS-CoV-2 was processed to discover relevant peptides for a vaccine, and SARS-CoV was processed to reveal common peptides between the two viruses during evaluation. The SARS-CoV-2 proteome is comprised of four structural proteins (E, M, N, and S) and at least six additional ORFs encoding nonstructural proteins, including the SARS-CoV-2 protease (Finkel et al., 2020; Zhang et al., 2020a). We obtained the SARS-CoV-2 viral proteome from GISAID (Elbe and Buckland-Merrett, 2017) sequence entry Wuhan/IPBCAMS-WH-01/2019, the first documented case. We used

Nextstrain (Hadfield et al., 2018) to identify open reading frames (ORFs) and translate the sequence. Our sliding windows on SARS-CoV-2 resulted in 29,403 candidate peptides for MHC class I and 125,593 candidate peptides for MHC class II. We obtained the SARS-CoV proteome from UniProt (Consortium, 2019) under Proteome ID UP000000354. For SARS-CoV, our procedure creates 29,661 and 126,711 unique peptides for MHC class I and class II, respectively.

4.1.3 Peptide Filtering

Removal of Highly Mutable Peptides. We eliminate peptides that are observed to mutate above an input threshold rate to improve coverage over all SARS-CoV-2 variants and reduce the chance that the virus will mutate and escape vaccine-induced immunity in the future. When possible, we select peptides that are observed to be perfectly conserved across all observed SARS-CoV-2 viral genomes. Peptides that are observed to be perfectly conserved in thousands of examples may be functionally constrained to evolve slowly or not at all. If functional data are available, they can be used to supplement observed viral genome mutation rates by increasing mutation rates over functionally non-constrained residues.

For SARS-CoV-2, we obtained the most up to date version of the GISAID database (Elbe and Buckland-Merrett, 2017) (as of 2:02pm EST May 13, 2020, see Table S4: GISAID acknowledgements) and used Nextstrain (Hadfield et al., 2018) to remove genomes with sequencing errors, translate the genome into proteins, and perform multiple sequence alignments (MSAs). We retrieved 24468 sequences from GISAID, and 19288 remained after Nextstrain quality processing. After quality processing, Nextstrain randomly sampled 34 genomes from every geographic region and month to produce a representative set of 5142 genomes for evolutionary analysis. Nextstrain definition of a "region" can vary from a city (e.g., "Shanghai") to a larger geographical district. Spatial and temporal sampling in Nextstrain is designed to provide a representative sampling of sequences around the world.

The 5142 genomes sampled by Nextstrain were then translated into protein sequences and aligned. We eliminated viral genome sequences that had a stop codon,

a gap, an unknown amino acid (because of an uncalled nucleotide in the codon), or had a gene that lacked a starting methionine, except for ORF1b which does not begin with a methionine. This left a total of 4690 sequences that were used to compute peptide level mutation probabilities. For each peptide, the probability of mutation was computed as the number of non-reference peptide sequences observed divided by the total number of peptide sequences observed.

Removal of Cleavage Regions. SARS-CoV-2 contains a number of post-translation cleavage sites in ORF1a and ORF1b that result in a number of nonstructural protein products. Cleavage sites for ORF1a and ORF1b were obtained from UniProt (Consortium, 2019) under entry P0DTD1. In addition, a furin-like cleavage site has been identified in the spike protein (Wang et al., 2020; Coutard et al., 2020). This cleavage occurs before peptides are loaded in the endoplasmic reticulum for class I or endosomes for class II. Any peptide that spans any of these cleavage sites is removed from consideration. This removes 3,739 peptides out of the 154,996 we consider across windows 8–10 (class I) and 13–25 (class II) ($\sim$2.4%).

Removal of Glycosylated Peptides. Glycosylation is a post-translational modification that involves the covalent attachment of carbohydrates to specific motifs on the surface of the protein. We eliminate all peptides that are predicted to have N-linked glycosylation as it can inhibit MHC class I peptide loading and T cell recognition of peptides (Wolfert and Boons, 2013; Wrapp et al., 2020). In addition, we do not know how well existing peptide prediction methods function on glycosylated peptides. The use of peptides that are natively glycosylated in a virus would likely require that vaccine peptides be identically glycosylated to enable T cell recognition of vaccine primed memory. The use of non-glycosylated vaccine peptides in this case has resulted in vaccine failures (Wolfert and Boons, 2013).

We identified peptides that may be glycosylated with the NetNGlyc N-glycosylation prediction server (Gupta et al., 2004). We verified these predictions for the spike protein by ensuring they were in the same locations as those found using experimental

data of spike N-glycosylation from Cryo-EM (Walls et al., 2020) and tandem mass spectrometry (Zhang et al., 2020b). A majority of the potential N-glycosylation sites (16 out of 22) were identified in both experimental studies, and further supported by homologous regions with glycosylation found in SARS-CoV (Walls et al., 2020). We found that all 22 experimentally identified real or likely N-glycosylation sites from the SARS-CoV-2 spike protein were predicted to be glycosylated with non-zero probability by NetNGlyc. Therefore, we eliminated all peptides where NetNGlyc predicted a non-zero N-glycosylation probability in any residue. This resulted in the elimination of 18,957 of the 154,996 peptides considered ($\sim$12%).

Removal of Self-epitopes. T cells are selected to ignore peptides derived from the normal human proteome, and thus we remove any self-peptides from consideration for a vaccine. In addition, it is possible that a vaccine might stimulate the adaptive immune system to react to a self-peptide that was presented at an abnormally high level, which could lead to an autoimmune disorder. All peptides from SARS-CoV-2 were scanned against the entire human proteome downloaded from UniProt (Consortium, 2019) under Proteome ID UP000005640. A total of 48 exact peptide matches (46 8-mers, two 9-mers) were discovered and eliminated from consideration.

Removal of Undesired Proteins. OptiVax can design vaccines using peptides from specific viral or oncogene proteins of interest by removing peptides from undesired proteins from the candidate pool. Grifoni et al. (2020b) tested T cell responses from COVID-19 convalescent patients and found that peptides from the S, M, and N proteins of SARS-CoV-2 produce the dominant $CD4^+$ and $CD8^+$ responses when compared to other SARS-CoV-2 proteins. We used OptiVax to produce additional SARS-CoV-2 vaccines comprised of peptides drawn from only the S, M, and N proteins (Section 4.2.2).

4.1.4 Computational Models for Candidate Peptide Scoring

Computational Peptide-HLA Prediction Models. For a peptide vaccine to be effective, its constituent peptides need to be displayed, and thus a computational vaccine design must be built upon a solid predictive foundation of what peptides will be displayed by each HLA allele. Incorrect predictions could lead to failure of a pre-clinical or clinical trial at great human cost. To this end we are concerned with the precision (true positives / all positives) of our predictions such that we maximize the chance that a peptide predicted to be displayed will in fact be displayed. We are less concerned with our ability to recall all of the peptides that will be displayed as long as we have a set of suitable size that will be displayed. We reduce the risk of false positives by employing multiple computational methods to predict peptide-HLA binding.

All models take as input a (HLA, peptide) pair and output predicted peptide-HLA binding affinity (IC50) in nanomolar units. For both MHC class I and class II models, we consider peptides to be binders if the predicted HLA binding affinity is $\leq$ 50 nM (Sette et al., 1994), providing a conservative threshold to increase the probability of peptide display. For MHC class I design, we use an ensemble that outputs the mean predicted binding affinity of NetMHCpan-4.0 (Jurtz et al., 2017) and MHCflurry 1.6.0 (O'Donnell et al., 2020a, 2018). We find this ensemble increases the precision of binding affinity estimates over the individual models on available SARS-CoV-2 experimental data (Section 4.2.1). For MHC class II design, we use NetMHCIIpan-4.0 (Reynisson et al., 2020b). For evaluation, we use our ensemble estimate of binding (MHC class I), as well as use binding predictions from multiple prediction algorithms (MHC class I: NetMHCpan-4.0 (Jurtz et al., 2017), NetMHCpan-4.1 (Reynisson et al., 2020a), MHCflurry 1.6.0 (O'Donnell et al., 2020a), PUFFIN (Zeng and Gifford, 2019); MHC class II: NetMHCIIpan-3.2 (Jensen et al., 2018), NetMHCIIpan-4.0 (Reynisson et al., 2020b), PUFFIN (Zeng and Gifford, 2019)) to ensure that all methods agree that we have a good peptide vaccine. Where our methods require a probability of peptide-HLA binding (as in Equation 4.5), affin-

ity predictions are capped at 50000 nM and transformed into $[0, 1]$ using a logistic transformation, $1 - \log_{50000}(\text{aff})$, where larger values correspond to greater likelihood of eliciting an immunogenic response (Sette et al., 1994; Buus et al., 2003; Nielsen et al., 2003). The ≤ 50 nM binding affinity threshold corresponds to a threshold of ≥ 0.638 after logistic transformation.

Our computational predictions of peptide display specify the supporting HLA alleles, thus enabling immunogenicity testing of peptides on HLA matched individuals. When available, these data can be used to eliminate peptide support by particular HLA alleles when the peptides are found to be non-immunogenic (Figure 4-1).

Comparison of Binding Affinity and Rank Predictions. NetMHCpan-4.0 (Jurtz et al., 2017) and NetMHCIIpan-4.0 (Reynisson et al., 2020b) output predicted binding affinity (BA), percentile rank of predicted BA compared to a set of random natural peptides, and percentile rank of an eluted ligand (EL) score compared to a set of random natural peptides. Default parameters for these methods suggest EL percentile rank thresholds of 0.5% and 2% rank for classifying peptides as strong and weak binders, respectively, for MHC class I and thresholds of 2% and 10% for strong and weak binders, respectively, for MHC class II.

To score peptides for our vaccine designs, we use a 50 nM predicted binding affinity threshold. We found binders selected with this criterion are also considered binders under alternative criteria based on percentile rank and increased precision on available SARS-CoV-2 experimental data (Section 4.2.1). Across our set of all candidate SARS-CoV-2 MHC class I peptides, we found 91.0% of peptide-HLA hits with ≤ 50 nM predicted binding affinity by NetMHCpan-4.0 were also considered binders using BA percentile rank $\leq 0.5\%$ (100.0% have BA percentile rank $\leq 2\%$). Using percentile rank for EL scores, 67.6% of peptide-HLA hits with ≤ 50 nM predicted binding affinity have EL percentile rank $\leq 0.5\%$ (92.6% have EL percentile rank $\leq 2\%$). Across all candidate SARS-CoV-2 MHC class II peptides, we found 86.1% of peptide-HLA hits with ≤ 50 nM predicted binding affinity by NetMHCIIpan-4.0 were also considered binders using BA percentile rank $\leq 2\%$ (100.0% have BA percentile rank $\leq$

10%). Using percentile rank for EL scores, 26.1% of peptide-HLA hits with $\leq$ 50 nM predicted binding affinity have EL percentile rank $\leq$ 2% (63.1% have EL percentile rank $\leq$ 10%).

Binders selected using percentile rank metrics were generally not considered binders under a 50 nM predicted binding threshold. Across our set of all candidate SARS-CoV-2 MHC class I peptides, we found 17.5% of peptide-HLA hits with EL percentile rank $\leq$ 0.5% have $\leq$ 50 nM predicted binding affinity by NetMHCpan-4.0. Across all candidate SARS-CoV-2 MHC class II peptides, we found 11.3% of peptide-HLA hits with EL percentile rank $\leq$ 2.0% have $\leq$ 50 nM predicted binding affinity by NetMHCIIpan-4.0.

4.1.5 HLA Population Frequency Computation

When we compute the probability of vaccine coverage over a population, we use complementary methods that assume either independence or linkage between allele frequencies in genomically proximal HLA loci. In EvalVax-Unlinked, we assume independence and use HLA allelic frequencies for 2392 class I alleles and 280 class II alleles across 15 geographic regions from the dbMHC database (Helmberg et al., 2004) obtained from the IEDB Population Coverage Tool (Bui et al., 2006). For each geographic region, we normalize the frequencies within each locus. If the sum of the raw frequencies exceeds one we normalize them to one, and if the sum of the raw frequencies is less than one the missing frequency is made up by a placeholder allele that receives no binding. In EvalVax-Robust, we assume linkage and use observed haplotype frequencies of HLA-A, HLA-B, and HLA-C loci for class I computations, or observed haplotype frequencies of HLA-DP, HLA-DQ, and HLA-DR for class II computations. We observed a total of 2138 distinct haplotypes for the HLA class I locus that include 230 different HLA-A, HLA-B, and HLA-C HLA alleles. We observed a total of 1711 distinct haplotypes for the HLA class II locus that include 280 different HLA-DP, HLA-DQ, and HLA-DR HLA alleles. We have independent haplotype frequency measurements for three populations self-reporting as having White (European), Black (African), or Asian ancestry.

HLA class I and class II haplotype frequencies were inferred using high resolution typing of individuals from distinct racial background. We estimated HLA class I haplotypes from HLA-A,-B, and -C genotypes of 2886 individuals of Black ancestry (46 distinct HLA-A alleles, 70 distinct HLA-B alleles, 40 distinct HLA-C alleles), 2327 individuals of White ancestry (38 distinct HLA-A alleles, 64 distinct HLA-B alleles, 34 distinct HLA-C alleles) and 1653 individuals of Asian ancestry (25 distinct HLA-A alleles, 51 distinct HLA-B alleles, 25 distinct HLA-C alleles). HLA class II haplotypes were estimated based on DR, DQ, DP genotypes of 2474 individuals of Black ancestry (10 distinct HLA-DPA1 alleles, 45 distinct HLA-DPB1 alleles, 14 distinct HLA-DQA1 alleles, 21 distinct HLA-DQB1 alleles, 38 distinct HLA-DRB1 alleles), 1857 individuals of White ancestry (7 distinct HLA-DPA1 alleles, 29 distinct HLA-DPB1 alleles, 18 distinct HLA-DQA1 alleles, 21 distinct HLA-DQB1 alleles, 41 distinct HLA-DRB1 alleles) and 1675 individuals of Asian ancestry (7 distinct HLA-DPA1 alleles, 28 distinct HLA-DPB1 alleles, 16 distinct HLA-DQA1 alleles, 16 distinct HLA-DQB1 alleles, 36 distinct HLA-DRB1 alleles). For each racial background, HLA class I and class II haplotypes were inferred using Hapferret[1], an implementation of the Expectation-Maximization algorithm (Excoffier and Slatkin, 1995). A total of 1200, 779, and 440 class I and 920, 537, and 502 class II haplotype frequencies were derived in Black, White, and Asian populations, respectively.

4.1.6 EvalVax Population Coverage Objectives

EvalVax-Robust. EvalVax-Robust computes the distribution of per individual peptide-HLA binding hits over a given population. It accounts for the significant linkage disequilibrium (LD) between HLA loci and uses haplotype frequencies for population coverage estimates. We expect that a vaccine will be more effective if more of its peptides are displayed by an individual's HLA molecules, and thus EvalVax-Robust computes the probability of having at least N predicted peptide-HLA binding hits for each individual in the population.

Assuming for each of the HLA-A,B,C loci there are M_A, M_B, M_C alleles respec-

[1] `https://github.com/nilsboar/hap-ferret`

tively, for a given haploid $A_iB_jC_k$, the haplotype frequency is defined as $G(i,j,k)$ and $\sum_{i=1}^{M_A}\sum_{j=1}^{M_B}\sum_{k=1}^{M_C} G(i,j,k) = 1$. We assume independence of inherited haplotypes and compute the frequency of a diploid genotype as:

$$F_{i_1j_1k_1i_2j_2k_2} = F(A_{i_1}B_{j_1}C_{k_1}, A_{i_2}B_{j_2}C_{k_2}) = G(i_1,j_1,k_1)G(i_2,j_2,k_2) \tag{4.1}$$

For each allele a, $e(a)$ denotes the number of peptides predicted to bind to the allele with ≤ 50 nM affinity, which we call the number of peptide-HLA hits. Then for each possible diploid genotype we compute the total number of peptide-HLA hits of the genotype as the sum of $e(a)$ of the unique alleles in the genotype (there can be 3–6 unique alleles depending on the zygosity of each locus):

$$C_{i_1j_1k_1i_2j_2k_2} = C(A_{i_1}B_{j_1}C_{k_1}, A_{i_2}B_{j_2}C_{k_2}) = \sum_{a\in\{A_{i_1},B_{j_1},C_{k_1}\}\cup\{A_{i_2},B_{j_2},C_{k_2}\}} e(a) \tag{4.2}$$

We then compute the frequency of having exactly k peptide-HLA hits in the population as:

$$P(n=k) = \sum_{i_1=1}^{M_A}\sum_{j_1=1}^{M_B}\sum_{k_1=1}^{M_C}\sum_{i_2=1}^{M_A}\sum_{j_2=1}^{M_B}\sum_{k_2=1}^{M_C} F_{i_1j_1k_1i_2j_2k_2}\mathbb{1}\{C_{i_1j_1k_1i_2j_2k_2} = k\} \tag{4.3}$$

We define the population coverage objective function for EvalVax-Robust as the probability of having at least N peptide-HLA hits in the population, where the threshold N is set to the minimum number of displayed vaccine peptides desired:

$$P(n \geq N) = \sum_{k=N}^{\infty} P(n=k) \tag{4.4}$$

When we evaluate metrics on a world population, we equally weight population coverage estimations over three population groups (White, Black, and Asian) as the final objective function. In addition to the probability of having at least N peptide-HLA hits per individual, we also evaluate the expected number of per individual peptide-HLA hits in the population, which provides insight on how well the vaccine is

displayed on average. The EvalVax objective can be used independently to evaluate peptide vaccine population coverage (Figure 4-1).

EvalVax-Unlinked. When haplotype frequencies are not available for a population, we can evaluate a vaccine using HLA allele frequencies that assume independence and compute the probability that at least one peptide binds to any of the alleles at any of the loci. To encourage a diverse set of peptides to bind to a single HLA allele, we use the predicted binding probability of a peptide to an allele instead of using a binary indicator of binding. This permits multiple peptides to contribute to the probability score at each allele. Considering K loci $\{L_1, ..., L_K\}$, for each locus there are M_k alleles $a_1, ..., a_{M_k}$ and the allele frequency is defined as $G_k(a_i)$ and $\sum_{i=1}^{M_k} G_k(a_i) = 1$. Given a set of N peptides $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$, for each allele (of locus L_k) the predicted binding probability to peptide P_n is $e_k^n(a_i)$. Assuming no competition between peptides, the probability that allele a_i ends up having at least one peptide bound is:

$$e_k(a_i) = 1 - \prod_{n=1}^{N}(1 - e_k^n(a_i)) \tag{4.5}$$

We define the diploid frequency of alleles as $F_k(a_i, a_j) = G_k(a_i)G_k(a_j)$, and we conservatively assume that a homozygous diploid locus does not improve the chance of peptide presentation over a single copy of the locus. Thus, the probability that a diploid genotype has at least one peptide bound is defined as:

$$B_k(a_i, a_j) = \begin{cases} 1 - (1 - e_k(a_i))(1 - e_k(a_j)), & \text{if} \quad i \neq j \\ e_k(a_i), & \text{if} \quad i = j \end{cases} \tag{4.6}$$

Therefore, the probability that a person in the given population displays at least one peptide in the set $\mathcal{P}$ at a particular locus L_k is calculated by:

$$F_k(\mathcal{P}) = \sum_{i=1}^{M_k}\sum_{j=1}^{M_k} F_k(a_i, a_j)B_k(a_i, a_j) \tag{4.7}$$

To combine different loci assuming no linkage disequilibrium, the probability that a person in the given population has at least one locus that binds to at least one peptide from $\mathcal{P}$ is defined as:

$$P(\mathcal{P}) = 1 - \prod_{k=1}^{K}(1 - F_k(\mathcal{P})) \tag{4.8}$$

which is the evaluation metric for EvalVax-Unlinked.

We conservatively only consider peptides with predicted binding affinity ≤ 50 nM. We set values of $e_k^n(a_i)$ weaker than 50 nM predicted binding affinity to zero. This constraint on peptide binding operates in addition to the peptide filtering steps described in Section 4.1.3. When we evaluate vaccines on a world population, we equally weight population coverage estimates over 15 geographic regions (see Figure 4-4b) as the final objective function.

4.1.7 OptiVax

Coverage optimization is performed by OptiVax using beam search to efficiently select an optimal subset of peptides that maximizes a desired population coverage objective. Starting from an empty set, OptiVax iteratively expands solutions in the beam by adding one peptide at a time, and keeps the top k solutions over all possible expansions in the beam. We use a beam size of $k = 10$ for MHC class I and $k = 5$ for MHC class II.

OptiVax-Robust. OptiVax-Robust uses beam search to find a minimal set of peptides that reaches a target population coverage probability at a threshold of n predicted peptide-HLA hits for each individual. We start from an empty set of peptides and $n = 0$, and iteratively expand the solution by one peptide at a time and retain the top k solutions until the population coverage probability for the current n reaches the target population coverage probability threshold for that n. We then repeat the same process for $n + 1$. If it not possible to reach the target population coverage probability threshold for n then the current coverage is accepted and we repeat the process for $n + 1$. At the expense of increased computational cost, beam search im-

proves upon greedy optimization by considering k possible solutions at each step. During each iteration, the population coverage probability threshold at the present n controls the robustness of coverage. Increasing the target population coverage probability increases the difficulty of the optimization task. The iterative process stops when the target population coverage at the desired n is achieved. In early rounds of optimization, OptiVax uses a high population coverage probability to provide better individual coverage. In subsequent rounds, the target population coverage probability is reduced on a fixed schedule.

In Liu et al. (2022), this combinatorial optimization problem is formally defined as the MAXIMUM n-TIMES COVERAGE problem, which is shown to be NP-complete and not submodular. The beam search procedure described above is formalized as the MARGINALGREEDY algorithm (Liu et al., 2022).

OptiVax-Unlinked. OptiVax-Unlinked optimizes the EvalVax-Unlinked objective function (Section 4.1.6) that considers HLA allele frequencies at each HLA locus independently. OptiVax-Unlinked uses beam search to find a minimal set of peptides that reaches a desired population coverage probability that each individual on average displays at least one vaccine peptide.

Peptide Sequence Diversity Constraints. During optimization, we add sequence diversity constraints between vaccine peptides to avoid selection of overlapping peptide sequences in a vaccine formulation. This issue arises because sliding a window over a proteome produces overlapping sequences that are very similar in HLA binding characteristics. When any version of OptiVax selects a peptide during optimization, it eliminates from further consideration all unselected peptides that are within three (MHC class I) or five (MHC class II) edits on a sequence distance metric from the selected peptide. The distance metric computation aligns two peptides not allowing gaps and mismatches and the distance metric is the sum of the lengths of any end overhangs where the opposite peptide sequence is absent.

4.2 Results

4.2.1 Validation of ML Models on Experimental Stability Data

We evaluate peptide-HLA binding predictions on a set of experimentally assessed SARS-CoV-2 peptides whose peptide-HLA complex stability was assessed in vitro across 11 MHC allotypes (5 HLA-A, 1 HLA-B, 4 HLA-C, 1 HLA-DRB1) (Prachar et al., 2020). Prachar et al. (2020) suggests peptides with at least 60% of the stability of a reference peptide in a NeoScreen assay are likely high affinity binders. For MHC class I alleles, the dataset contains 912 unique peptide-HLA pairs, of which 185 peptides are considered stable ($\geq$ 60% stability). For MHC class II, the dataset contains 93 total peptides, of which 22 are stable. We use our computational models to predict peptide-HLA binding and evaluate them using various binding criteria against the experimental peptide stability measurement (Table 4.1). We compare classification performance using different binding criteria (Section 4.1.4) and find in general that classifying binders using predicted binding affinity maximizes AUROC, and a 50 nM binding affinity threshold maximizes precision (Table 4.1). We find our mean ensemble of NetMHCpan-4.0 and MHCflurry further improves classification AUROC and precision over the individual models for predicting MHC class I epitopes. On MHC class II data, NetMHCIIpan-4.0 achieves AUROC 0.848 and precision 0.625 using a 500 nM threshold (Table 4.1). While NetMHCIIpan-4.0 with a 50 nM threshold does not identify any peptides in this dataset as binders, we use this stricter threshold in our vaccine designs as it is more conservative and less likely to admit false positive binders. In general, we find performance of PUFFIN with a 50 nM binding threshold comparable to alternative methods on both MHC class I and class II data and use PUFFIN as part of our vaccine design evaluation (Section 4.2.6). We further calibrated ML models using experimental immunogenicity data from convalescent COVID-19 patients, and results are presented in Liu et al. (2021).

Table 4.1: Classification performance of computational methods for predicting peptide-MHC binding evaluated on experimental SARS-CoV-2 peptide stability data (Prachar et al., 2020) across 11 MHC allotypes (5 HLA-A, 1 HLA-B, 4 HLA-C, 1 HLA-DRB1). Ensemble outputs the mean predicted binding affinity of NetMHCpan-4.0 and MHCflurry (Section 4.1.4). Classification performance of peptide-MHC scoring models was calculated using scikit-learn (Pedregosa et al., 2011). AUROC and average precision are computed using raw predictions, and the remaining metrics are computed using binarized predictions based on the respective binding criteria. (BA = binding affinity, EL = eluted ligand)

MHC	Model	Binding Criterion	AUROC	Precision	Sensitivity	Specificity	Avg. Precision
Class I	NetMHCpan-4.0	BA $\leq$ 50 nM	0.845	0.516	0.714	0.829	0.486
	NetMHCpan-4.0	BA $\leq$ 500 nM	0.845	0.308	0.968	0.446	0.486
	NetMHCpan-4.0	BA % Rank $\leq$ 0.5	0.746	0.249	0.968	0.257	0.416
	NetMHCpan-4.0	BA % Rank $\leq$ 2	0.746	0.212	1.000	0.054	0.416
	NetMHCpan-4.0	EL % Rank $\leq$ 0.5	0.757	0.256	0.930	0.312	0.479
	NetMHCpan-4.0	EL % Rank $\leq$ 2	0.757	0.214	0.989	0.077	0.479
	NetMHCpan-4.1	BA $\leq$ 50 nM	0.853	0.504	0.719	0.820	0.499
	NetMHCpan-4.1	BA $\leq$ 500 nM	0.853	0.304	0.984	0.428	0.499
	NetMHCpan-4.1	EL % Rank $\leq$ 0.5	0.776	0.278	0.903	0.403	0.490
	NetMHCpan-4.1	EL % Rank $\leq$ 2	0.776	0.219	0.989	0.103	0.490
	MHCflurry 1.6.0	BA $\leq$ 50 nM	0.724	0.404	0.422	0.842	0.411
	PUFFIN	BA $\leq$ 50 nM	0.768	0.526	0.492	0.887	0.485
	PUFFIN	BA $\leq$ 500 nM	0.768	0.272	0.870	0.406	0.485
	Ensemble	Mean BA $\leq$ 50 nM	0.862	0.683	0.514	0.939	0.650
Class II	NetMHCIIpan-4.0	BA $\leq$ 50 nM	0.848	0.000	0.000	1.000	0.762
	NetMHCIIpan-4.0	BA $\leq$ 500 nM	0.848	0.625	0.682	0.873	0.762
	NetMHCIIpan-4.0	EL % Rank $\leq$ 2	0.908	1.000	0.182	1.000	0.785
	NetMHCIIpan-4.0	EL % Rank $\leq$ 10	0.908	0.789	0.682	0.944	0.785
	NetMHCIIpan-3.2	BA $\leq$ 50 nM	0.766	1.000	0.045	1.000	0.544
	NetMHCIIpan-3.2	BA $\leq$ 500 nM	0.766	0.253	0.909	0.169	0.544
	NetMHCIIpan-3.2	BA % Rank $\leq$ 2	0.766	0.380	0.864	0.563	0.536
	NetMHCIIpan-3.2	BA % Rank $\leq$ 10	0.766	0.253	1.000	0.085	0.536
	PUFFIN	BA $\leq$ 50 nM	0.704	0.667	0.091	0.986	0.430
	PUFFIN	BA $\leq$ 500 nM	0.704	0.275	0.864	0.296	0.430

4.2.2 OptiVax-Robust Vaccine Designs for SARS-CoV-2

MHC class I. We selected an optimized set of peptides from all SARS-CoV-2 proteins using OptiVax-Robust and the EvalVax-Robust objective function. We limited our candidates to peptides with length 8–10 and excluded peptides that have been observed with any mutation or are predicted to have non-zero probability of glycosylation (Section 4.1.2). For computation of the objective function, we use the mean predicted IC50 values from our NetMHCpan-4.0 and MHCflurry ensemble to obtain reliable binding affinity predictions for evaluation and optimization. After peptide

filtering, we had 378 candidate peptides. With OptiVax-Robust optimization, we designed a vaccine with 19 peptides that achieves 99.39% EvalVax-Unlinked coverage and 99.91% EvalVax-Robust coverage over three ethnic groups (Asian, Black, White) with at least one peptide-HLA hit per individual. This set of peptides also provides 93.21% coverage with at least 5 peptide-HLA hits and 67.75% coverage with at least 8 peptide-HLA hits (Figure 4-2, Table 4.2). The population level distribution of the number of peptide-HLA hits in White, Black, and Asian populations is shown in Figure 4-2, where the expected number of peptide-HLA hits is 9.358, 8.515, and 10.206, respectively.

MHC class II. We limited our candidates to peptides with length 13–25 and excluded peptides that have been observed with mutation probability greater than 0.001 or are predicted to have non-zero glycosylation probability. We use the predicted binding affinity from NetMHCIIpan-4.0 for optimization and evaluation. After peptide filtering, we had 7977 candidate peptides. With OptiVax-Robust optimization, we designed a vaccine with 19 peptides that achieves 90.76% EvalVax-Unlinked coverage and 99.67% EvalVax-Robust coverage over three ethnic groups (Asian, Black, White) with at least one peptide-HLA hit per individual. This set of peptides also provides 97.21% coverage with at least 5 peptide-HLA hits and 88.48% coverage with at least 8 peptide-HLA hits (Figure 4-2, Table 4.2). The population level distribution of the number of peptide-HLA hits per individual in White, Black, and Asian populations is shown in Figure 4-2, where the expected number of of peptide-HLA hits is 16.635, 15.708, 11.000, respectively.

Vaccine designs with S, M, N proteins only. We also used OptiVax-Robust to design vaccines for MHC class I and class II based solely upon peptides from the S, M, and N proteins of SARS-CoV-2 and evaluated the resulting vaccine performance. Grifoni et al. (2020b) found that peptides from the S, M, and N proteins produced the majority of the $CD4^+$ (86%) and $CD8^+$ (60%) T cell response in 20 convalescent COVID-19 patients. Since Grifoni et al. (2020b) used megapool based assays, it is

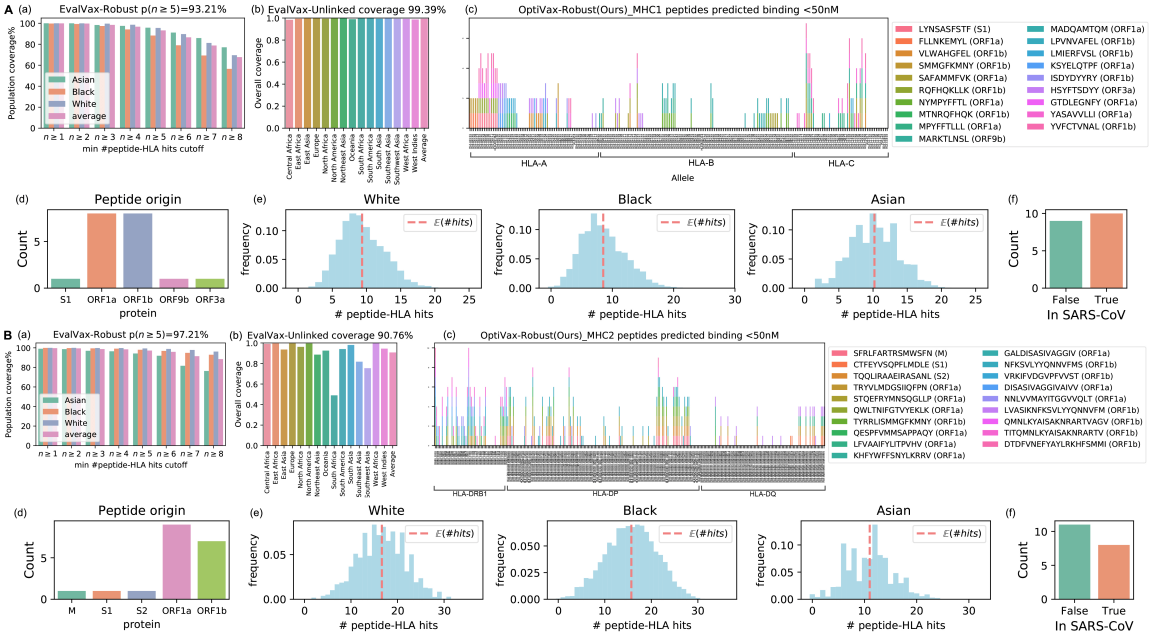


Figure 4-2: SARS-CoV-2 OptiVax-Robust selected peptide vaccine set for (A) MHC class I and (B) MHC class II. (a) EvalVax-Robust population coverage at different per-individual number of peptide-HLA hit cutoffs for populations self-reporting as having White, Black, or Asian ancestry and average values. (b) EvalVax-Unlinked population coverage on 15 geographic regions and averaged population coverage. (c) Binding of vaccine peptides to each of the available alleles in MHC I and II. (d) Peptide viral protein origins. (e) Distribution of the number of per-individual peptide-HLA hits in populations self-reporting as having White, Black, or Asian ancestry. (f) Vaccine peptide presence in SARS-CoV.

not possible to use their data to identify individual peptides that are immunogenic.

As shown in Table 4.2, our SMN only MHC class I vaccine with 26 peptides achieves 98.15% coverage over three ethnic groups (Asian, Black, White) with at least one average peptide-HLA hit per individual. There were an average of at least five peptide hits in 67.37% of the population, and the expected per-individual number of hits for White, Black, and Asian populations are 5.313, 5.643, and 6.448, respectively. The OptiVax-Robust MHC class II SMN only vaccine with 22 peptides achieves 98.57% coverage with an average of at least one peptide-HLA hit per individual. There were an average of at least five peptide hits in 85.37% of the population, and the expected per-individual number of hits in White, Black, and Asian populations are 11.309, 9.693 and 7.053, respectively. The detailed vaccine designs are in

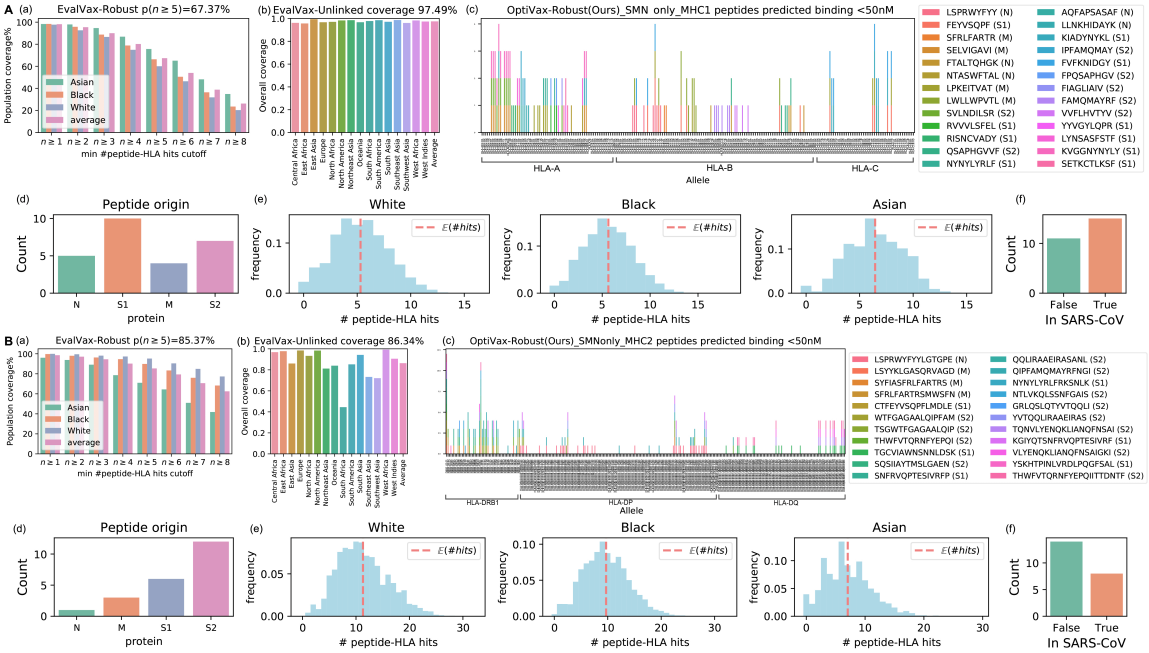


Figure 4-3: OptiVax-Robust designed peptide vaccine using peptides from the SARS-CoV-2 S, M, and N proteins only. (A) Results for MHC class I and (B) MHC class II. (a) EvalVax-Robust population coverage at different minimum number of peptide-HLA hit cutoffs. (b) EvalVax-Unlinked population coverage. (c) Binding of vaccine peptides to each of the available alleles in MHC I and II. (d) Peptide viral protein origins. (e) Distribution of the number of per-individual peptide-HLA hits in populations self-reporting as having White, Black, or Asian ancestry. (f) Vaccine peptide presence in SARS-CoV.

Figure 4-3. We observed that it is more difficult to optimize vaccines with S, N, and M proteins only. We expect this is because we have fewer candidate peptides to cover all of our haplotype combinations.

4.2.3 OptiVax-Unlinked Vaccine Designs for SARS-CoV-2

MHC class I. We limited our candidates to peptides with length 8–10 and zero predicted probability of glycosylation. We also excluded peptides that have been observed with any mutation. We use the mean predicted binding affinity values from our ensemble of NetMHCpan-4.0 and MHCflurry on 2392 HLA class I alleles to obtain reliable binding affinity predictions for evaluation and optimization. After peptide filtering, we had 472 candidate peptides. With OptiVax-Unlinked optimization, we

designed a vaccine with 19 peptides that achieves 99.79% EvalVax-Unlinked population coverage (averages over 15 geographic regions). As shown in Figure 4-4, the 19 vaccine peptides bind to a diverse range of alleles across the HLA-A/B/C loci. Even though less effective than OptiVax-Robust at providing a higher number of expected individual peptide-HLA hits in the population, the OptiVax-Unlinked peptide set still achieves high coverage on EvalVax-Robust metrics (99.99% for $p(n \geq 1)$, 89.15% for $p(n \geq 5)$, 49.59% for $p(n \geq 8)$). The expected per-individual number of peptide-HLA hits for the design is 7.340, 6.899, and 8.971 for White, Black, and Asian populations, respectively (Table 4.2).

MHC class II. We excluded peptides that have been observed with a mutation probability greater than 0.001 or are predicted to have non-zero probability of being glycosylated. We use the predicted binding affinity from NetMHCIIpan-4.0 for optimization and initial evaluation. After peptide filtering, we had 7966 candidate peptides. With OptiVax-Unlinked, we designed a vaccine with 19 peptides that achieves 91.67% EvalVax-Unlinked population coverage (averaged over 15 geographic regions). As shown in Figure 4-4, the 19 vaccine peptides bind to a diverse range of alleles across the HLA-DRB/DP/DQ loci. Even though less effective than OptiVax-Robust on providing a high predicted number of average peptide-HLA hits in the population, the OptiVax-Unlinked peptide set still achieves high coverage on EvalVax-Robust metrics (99.67% for $p(n \geq 1)$, 95.94% for $p(n \geq 5)$, 83.30% for $p(n \geq 8)$). The expected per-individual number of peptide-HLA hits for the design is 14.366, 12.711, and 9.657 for White, Black, and Asian populations, respectively (Table 4.2).

4.2.4 EvalVax Evaluation of Public SARS-CoV-2 Vaccine Designs

We used EvalVax to evaluate peptide vaccines and megapools proposed by other publications (Lee and Koohy, 2020; Fast et al., 2020; Poran et al., 2020; Bhattacharya et al., 2020; Baruah and Bose, 2020; Abdelmageed et al., 2020; Ahmed et al., 2020; Srivastava et al., 2020; Herst et al., 2020; Vashi et al., 2020; Akhand et al., 2020; Mitra et al.,

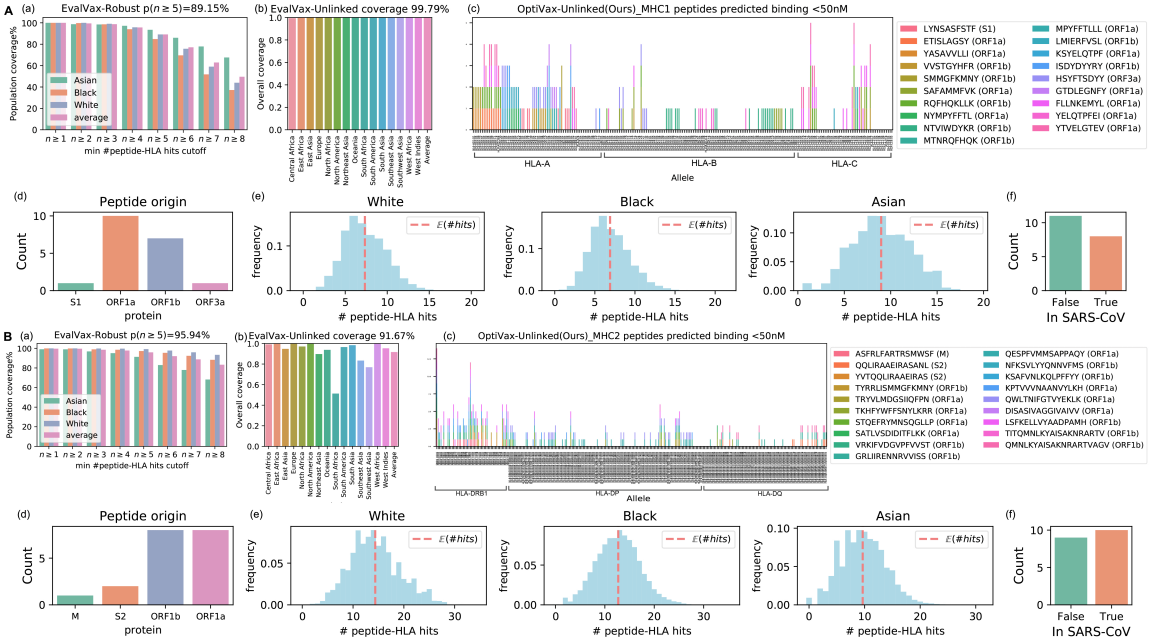


Figure 4-4: OptiVax-Unlinked selected SARS-CoV-2 optimal peptide vaccine set for (A) MHC class I and (B) MHC class II. (a) EvalVax-Robust population coverage at different per-individual number of peptide-HLA hits cutoffs for populations self-reporting as having White, Black, or Asian ancestry and average value. (b) EvalVax-Unlinked population coverage on 15 geographic regions and averaged population coverage. (c) Binding of vaccine peptides to each of the available alleles in MHC I and II. (d) Peptide viral protein origins. (e) Distribution of the number of per-individual peptide-HLA hits in populations self-reporting as having White, Black, or Asian ancestry. (f) Vaccine peptide presence in SARS-CoV.

2020; Khan et al., 2020; Banerjee et al., 2020; Ramaiah and Arumugaswami, 2020; Gupta et al., 2020; Saha and Prasad, 2020; Tahir ul Qamar et al., 2020; Singh et al., 2020a; Yarmarkovich et al., 2020; Grifoni et al., 2020a; Nerli and Sgourakis, 2020; Yazdani et al., 2020; Ismail et al., 2020) on metrics including EvalVax-Unlinked and EvalVax-Robust population coverage at different per-individual number of peptide-HLA hits thresholds, expected per-individual number of peptide-HLA hits in White, Black, and Asian populations, percentage of peptides that are predicted to be glycosylated, peptides observed to mutate with probability greater than 0.001, or peptides that sit on known cleavage sites. We define *normalized coverage* as the mean expected per-individual number of peptide-HLA hits for a vaccine divided by the number of peptides in the vaccine. Details of the baseline vaccine designs are presented in the

Key Resources Table of Liu et al. (2020a).

We evaluate whole protein vaccines by first converting them into the non-redundant peptides they display in a given haplotype population. Using a windowing strategy to enumerate all peptides in a whole protein vaccine produces a large number of overlapping redundant peptides that will cause EvalVax to provide optimistic and unrealistic vaccine metrics. We accomplish protein vaccine representation by using OptiVax to create a vaccine design from the entire protein vaccine payload without any limitation on the number of peptides in the vaccine. OptiVax eliminates highly redundant peptides during design and chooses the largest set of peptides that maximizes population coverage (Section 4.1.7). For example, EvalVax predicts SARS-CoV-2 S protein vaccines will have $n \geq 5$ MHC class II peptide hits in 95.99% of the population on average when simple windowing is employed resulting in 16315 redundant peptides, and 82.72% of the population when non-redundant S is used resulting in its representation as 102 peptides that are not glycosylated, and have a mutation probability of ≤ 0.001 (Table 4.2).

Figures 4-5 and 4-6 show the comparison between OptiVax-Robust designed MHC class I and class II vaccines at all vaccine sizes (top solution in the beam up to the given vaccine size) from 1–35 peptides (blue curves) and baseline vaccines (red crosses) proposed by other publications. We observe superior performance of OptiVax-Robust designed vaccines on all evaluation metrics at all vaccine sizes for both MHC class I and class II. Most baselines achieve reasonable coverage at $n \geq 1$ peptide hits. However, many fail to show a high probability of higher hit counts, indicating a lack of predicted redundancy if a single peptide is not displayed. We also evaluate randomly selected peptide sets of size 19 from predicted binders of MHC class I and II, where a binder is defined as a peptide predicted to bind with ≤ 50 nM to more than 5 of the alleles in the MHC class. We found that a set of random binders can achieve greater coverage than some of the proposed vaccines we use as baselines.

Table 4.2 summarizes EvalVax results for all baselines with a vaccine peptide count less than 150 peptides. We also evaluated an average of 200 random designs for MHC class I or class II that are comprised of 19 random peptides predicted to

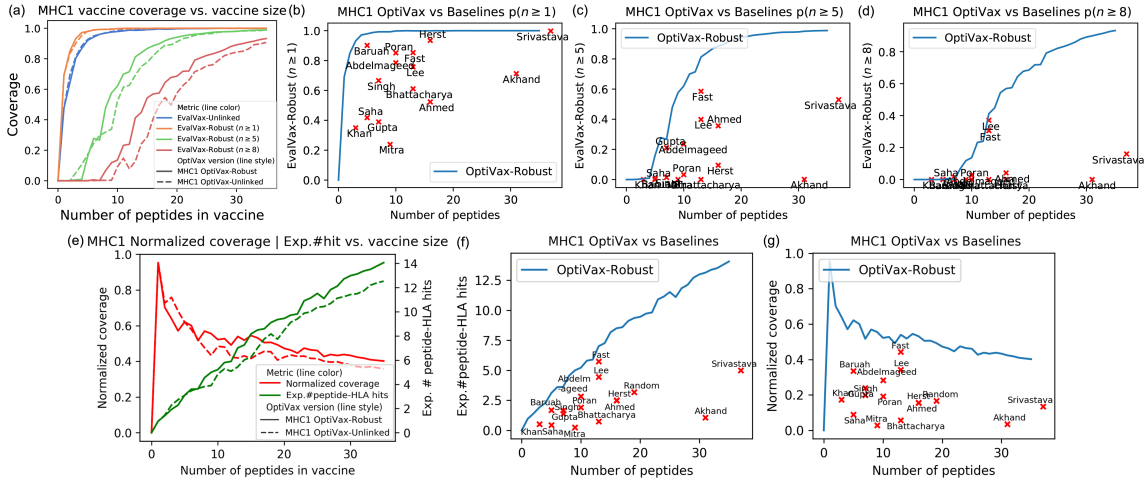


Figure 4-5: EvalVax population coverage evaluation, expectation of per individual number of peptide-HLA hits and normalized coverage for MHC class I SARS-CoV-2 vaccines. (a) EvalVax population coverage for OptiVax-Unlinked and OptiVax-Robust proposed vaccine at different vaccine sizes. (b) EvalVax-Robust population coverage with $n \geq 1$ peptide-HLA hits per individual, OptiVax-Robust performance is shown by the blue curve and baseline performance is shown by red crosses (labeled by name of first author). (c) EvalVax-Robust population coverage with $n \geq 5$ peptide-HLA hits. (d) EvalVax-Robust population coverage with $n \geq 8$ peptide-HLA hits. (e) Expected number of peptide-HLA hits vs. peptide vaccine size for OptiVax-Robust and OptiVax-Unlinked, and normalized coverage (hits / vaccine size) at different vaccine size. (f) Comparison of OptiVax-Robust and baselines on expected number of peptide-HLA hits. OptiVax-Robust performance is shown by the blue curve and baseline performance is shown by red crosses. (g) Comparison between OptiVax-Robust and baselines on normalized coverage.

bind with $\leq$ 50 nM to more than 5 of the alleles in the MHC class. We found that the baseline methods all provide less coverage than OptiVax derived sets, and some contain peptides predicted to be glycosylated or have a high observed mutation probability (Table 4.2). We also observe some baselines contain peptides that cross cleavage sites or overlap with self-peptides.

4.2.5 OptiVax Augmentation of SARS-CoV-2 S Protein Vaccines

When predicted population coverage for a whole protein vaccine is found insufficient, OptiVax can perform optimized augmented vaccine design to suggest additional pep-

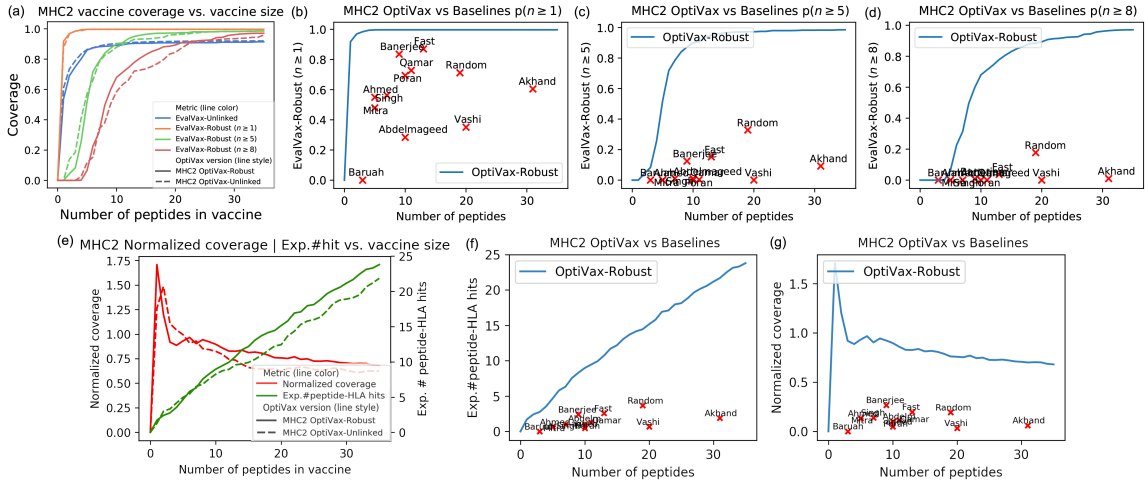


Figure 4-6: EvalVax population coverage evaluation, expectation of per individual number of peptide-HLA hits and normalized coverage for MHC class II SARS-CoV-2 vaccines. (a) EvalVax population coverage for OptiVax-Unlinked and OptiVax-Robust proposed vaccine at different vaccine sizes. (b) EvalVax-Robust population coverage with $n \geq 1$ peptide-HLA hits per individual, OptiVax-Robust performance is shown by the blue curve and baseline performance is shown by red crosses (labeled by name of first author). (c) EvalVax-Robust population coverage with $n \geq 5$ peptide-HLA hits. (d) EvalVax-Robust population coverage with $n \geq 8$ peptide-HLA hits. (e) Expected number of peptide-HLA hits vs. peptide vaccine size for OptiVax-Robust and OptiVax-Unlinked, and normalized coverage (hits / vaccine size) at different vaccine size. (f) Comparison of OptiVax-Robust and baselines on expected number of peptide-HLA hits. OptiVax-Robust performance is shown by the blue curve and baseline performance is shown by red crosses. (g) Comparison between OptiVax-Robust and baselines on normalized coverage.

tides to add to an existing formulation. In this mode, we use OptiVax to compute the non-redundant displayed peptide set for a protein vaccine, and then use this as the initial set of peptides for OptiVax design. OptiVax then adds additional non-redundant peptides to this initial set to increase population coverage.

We used OptiVax-Robust augmentation to add 26 peptides to the SARS-CoV-2 S protein vaccine to increase the predicted MHC class II population coverage for $n \geq 5$ peptide hits from 82.72% to 98.73%. For MHC class I, OptiVax augmentation added 16 peptides to the SARS-CoV-2 S protein vaccine to increase the predicted population coverage for $n \geq 5$ peptide hits from 97.4% to 99.9% (Table 4.2). OptiVax vaccine designs, non-redundant peptide sets, and vaccine augmentations are presented in Supplemental Table S3 of Liu et al. (2020a). Additional population coverage gap

results and SARS-CoV-2 protein subunit augmentation vaccine designs are presented in Liu et al. (2021).

4.2.6 EvalVax Vaccine Evaluation Using Alternative Prediction Models

We evaluated all Table 4.2 vaccine designs using eleven independent peptide-HLA binding prediction methods to ensure that the performance observed in Table 4.2 is consistent across prediction methods. For MHC class I prediction, we validated using seven methods: NetMHCpan-4.0; NetMHCpan-4.1; MHCflurry 1.6.0; PUFFIN; the mean of NetMHCpan-4.0 and MHCflurry 1.6.0 with a 50 nM threshold on predicted affinity; and NetMHCpan-4.0 and NetMHCpan-4.1 with a 99.5% threshold on EL ranking. For MHC class II prediction, we validated using four different methods: NetMHCIIpan-3.2 and NetMHCIIpan-4.0, each with either a 50 nM threshold on predicted affinity or a 98% threshold on EL ranking. The result of all eleven EvalVax evaluation metrics for all Table 4.2 designs are presented in Supplemental Table S2 of Liu et al. (2020a). We find all eleven methods used for evaluation show Table 4.2 is a conservative estimate of vaccine performance.

Table 4.2: Comparison of existing baselines, S-protein peptides, and OptiVax designed peptide vaccines (using all SARS-CoV-2 or S/M/N proteins only) on various population coverage evaluation metrics and vaccine quality metrics (percentage of peptides with mutation rate > 0.001 or with non-zero probability of being glycosylated). S-protein includes all possible S-protein peptides of lengths 8–10 (MHC class I) and 13–25 (MHC class II). Non-redundant peptide sets are a result of OptiVax analysis of non-redundant displayed peptides. The table is sorted by EvalVax-Robust $p(n \geq 1)$. Random subsets are generated 200 times. The binders used for generating random subsets are defined as peptides that are predicted to bind with ≤ 50 nM to more than 5 of the alleles.

Peptide Set	Vaccine Size	EvalVax-Unlinked	EvalVax-Robust *p(n≥1)*	EvalVax-Robust *p(n≥5)*	EvalVax-Robust *p(n≥8)*	Exp. # Peptide-HLA Hits / Vaccine Size	Exp. # Peptide-HLA Hits (White)	Exp. # Peptide-HLA Hits (Black)	Exp. # Peptide-HLA Hits (Asian)	Peptides Glycosylated	Peptides Mutation Rate > 0.001	On Cleavage Site	Protein Origins	In SARS-CoV
MHC Class I Peptide Vaccine Evaluation														
OptiVax Augmented Non-redundant S-protein	126 + 16	100.00%	100.00%	99.97%	99.27%	20.50%	27.20	27.68	32.44	0.00%	0.00%	0.00%	M, N, ORF1a, ORF1b, ORF3a, S1, S2	30.28%
S-protein	3795	99.96%	100.00%	99.17%	98.29%	0.91%	30.84	32.14	41.13	15.57%	29.99%	0.63%	S1, S2	29.30%
OptiVax-Unlinked	19	99.79%	99.99%	89.15%	49.59%	40.72%	7.34	6.90	8.97	0.00%	0.00%	0.00%	ORF1a, ORF1b, ORF3a, S1	42.11%
Non-redundant S-protein	126	99.84%	99.93%	97.37%	91.69%	16.82%	19.20	19.99	24.38	0.00%	0.00%	0.00%	S1, S2	27.78%
OptiVax-Robust	19	99.39%	99.91%	93.21%	67.75%	49.26%	9.36	8.52	10.21	0.00%	0.00%	0.00%	ORF1a, ORF1b, ORF3a, ORF9b, S1	52.63%
OptiVax-Robust – size 15	15	99.07%	99.89%	86.69%	54.36%	54.47%	8.17	7.20	9.14	0.00%	0.00%	0.00%	ORF1a, ORF1b, ORF9b, S1	53.33%
Non-redundant S1-subunit	68	99.18%	99.76%	86.53%	56.36%	12.23%	8.31	8.84	7.80	0.00%	0.00%	0.00%	S1	8.82%
Srivastava et al. (2020)	37	95.86%	99.75%	52.94%	16.00%	13.51%	5.37	4.99	4.64	8.11%	37.84%	0.00%	E, M, N, ORF10, ORF1a, ORF1b, ORF3a, ORF6, ORF7a, ORF7b, ORF8, S1	45.95%
OptiVax-Robust – S/M/N only	26	97.49%	98.15%	67.37%	26.24%	22.31%	5.31	5.64	6.45	0.00%	0.00%	0.00%	M, N, S1, S2	57.69%
Herst et al. (2020)	52	90.89%	95.82%	56.52%	19.99%	9.88%	5.20	4.44	5.77	7.69%	34.62%	0.00%	N	55.77%
Herst et al. (2020) – top 16	16	80.41%	93.46%	9.47%	0.03%	15.73%	2.75	2.60	2.20	12.50%	12.50%	0.00%	N	68.75%
Random subset of binders	19	81.04%	90.33%	25.02%	4.58%	16.74%	3.01	2.83	3.70	0.00%	29.89%	0.00%	N/A	40.37%
Baruah and Bose (2020)	5	71.91%	90.10%	0.55%	0.00%	33.60%	1.93	1.44	1.67	0.00%	40.00%	0.00%	S1, S2	40.00%
Fast et al. (2020)	13	78.66%	85.29%	58.51%	30.56%	44.25%	5.59	4.98	6.69	7.69%	30.77%	0.00%	E, M, N, ORF1a, S1, S2	23.08%
Poran et al. (2020)	10	69.12%	85.13%	3.21%	0.01%	19.23%	1.68	1.72	2.37	0.00%	30.00%	0.00%	ORF1a, ORF1b, ORF3a, ORF8, S1	20.00%
Vashi et al. (2020)	51	68.63%	80.80%	1.52%	0.00%	3.12%	1.90	1.70	1.17	11.76%	43.14%	5.88%	S1, S2	5.88%
Abdelmageed et al. (2020)	10	66.91%	78.49%	23.49%	2.72%	28.34%	2.93	2.50	3.07	10.00%	10.00%	0.00%	E	80.00%
Lee and Koohy (2020)	13	64.96%	75.75%	39.82%	37.09%	34.15%	4.77	3.69	4.86	0.00%	7.69%	0.00%	E, N, ORF1a, ORF1b, S2	53.85%
Akhand et al. (2020)	31	49.46%	71.24%	0.08%	0.00%	3.47%	1.09	1.11	1.02	3.23%	35.48%	0.00%	E, M, N, S1	41.94%
Singh et al. (2020a)	7	53.91%	66.59%	1.38%	0.00%	19.87%	1.34	1.30	1.53	0.00%	28.57%	0.00%	E, M, N, S1, S2	71.43%
Bhattacharya et al. (2020)	13	44.56%	61.09%	0.00%	0.00%	5.67%	0.79	0.69	0.73	23.08%	46.15%	7.69%	S1, S2	23.08%
Ahmed et al. (2020)	16	45.25%	52.30%	35.61%	4.15%	15.57%	2.56	2.18	2.73	12.50%	25.00%	0.00%	N, S2	100.00%
Saha and Prasad (2020)	5	29.90%	41.77%	0.00%	0.00%	8.86%	0.56	0.36	0.41	0.00%	20.00%	0.00%	S1	20.00%
Gupta et al. (2020)	7	30.23%	38.91%	21.08%	1.41%	23.92%	1.32	0.55	3.15	0.00%	42.86%	0.00%	S1, S2	14.29%
Khan et al. (2020)	3	27.14%	34.98%	0.00%	0.00%	17.33%	0.76	0.56	0.24	0.00%	66.67%	0.00%	S1, S2	0.00%
Mitra et al. (2020)	9	13.97%	23.86%	0.00%	0.00%	2.83%	0.15	0.08	0.54	22.22%	11.11%	0.00%	S1, S2	11.11%
MHC Class II Peptide Vaccine Evaluation														
OptiVax-Unlinked	19	91.67%	99.67%	95.94%	83.30%	64.45%	14.37	12.71	9.66	0.00%	0.00%	0.00%	M, ORF1a, ORF1b, S2	52.63%
OptiVax-Robust	19	90.76%	99.67%	97.21%	88.48%	76.04%	16.64	15.71	11.00	0.00%	0.00%	0.00%	M, ORF1a, ORF1b, S1, S2	42.11%
OptiVax Augmented Non-redundant S-protein	102 + 26	91.65%	99.67%	98.73%	97.27%	26.81%	43.79	36.06	23.12	0.00%	0.00%	0.00%	M, ORF1a, ORF1b, S1, S2	29.69%
Ramaiah and Arumugaswami (2020)	134	87.28%	98.88%	90.20%	83.97%	25.18%	45.04	38.25	17.93	20.15%	44.78%	0.00%	E, M, N, S1, S2	30.60%
S-protein	16315	89.80%	98.76%	95.99%	95.73%	2.22%	492.82	385.60	208.34	30.01%	57.50%	1.43%	S1, S2	16.06%
OptiVax-Robust – S/M/N only	22	86.34%	98.57%	85.37%	62.49%	42.51%	11.31	9.69	7.05	0.00%	0.00%	0.00%	M, N, S1, S2	36.36%
Non-redundant S-protein	102	84.91%	98.56%	82.72%	77.19%	16.61%	23.54	17.04	10.23	0.00%	0.00%	0.00%	S1, S2	28.43%
Non-redundant S1-subunit	53	77.14%	95.81%	63.43%	41.82%	16.33%	13.07	8.74	4.16	0.00%	0.00%	0.00%	S1	3.77%
Random subset of binders	19	72.41%	93.61%	58.67%	32.40%	31.59%	7.72	6.49	3.79	0.00%	63.79%	0.00%	N/A	23.55%
Fast et al. (2020)	13	67.29%	86.99%	15.24%	3.69%	19.69%	3.65	2.26	1.77	30.77%	38.46%	0.00%	E, M, N, ORF1a, S1, S2	0.00%
Banerjee et al. (2020)	9	56.73%	83.51%	12.49%	0.66%	26.65%	3.16	2.35	1.68	22.22%	44.44%	0.00%	S1, S2	55.56%
Tahir ul Qamar et al. (2020)	11	39.44%	72.75%	0.27%	0.00%	11.62%	1.84	1.46	0.53	0.00%	72.73%	0.00%	E, M, N, ORF10, ORF6, ORF7a, ORF8	36.36%
Poran et al. (2020)	10	42.30%	69.37%	0.00%	0.00%	9.83%	1.47	0.91	0.57	20.00%	90.00%	0.00%	ORF1a, ORF1b, ORF3a, S2	20.00%
Akhand et al. (2020)	31	43.90%	60.45%	9.22%	1.01%	6.08%	2.53	2.54	0.59	3.23%	48.39%	0.00%	E, M, N, S1	29.03%
Singh et al. (2020a)	7	41.48%	56.29%	0.96%	0.00%	14.02%	1.44	1.11	0.39	0.00%	28.57%	0.00%	E, M, N, S1, S2	71.43%
Ahmed et al. (2020)	5	27.69%	54.96%	0.00%	0.00%	13.08%	0.74	0.72	0.51	0.00%	20.00%	0.00%	N, S2	100.00%
Mitra et al. (2020)	5	25.46%	47.92%	0.04%	0.00%	13.14%	0.90	0.58	0.49	60.00%	20.00%	0.00%	S1, S2	0.00%
Vashi et al. (2020)	20	20.78%	35.12%	0.04%	0.00%	3.36%	0.96	0.62	0.44	15.00%	35.00%	5.00%	S1, S2	0.00%
Abdelmageed et al. (2020)	10	19.15%	28.40%	0.96%	0.00%	4.79%	0.92	0.27	0.24	60.00%	70.00%	0.00%	E	30.00%
Baruah and Bose (2020)	3	0.00%	0.00%	0.00%	0.00%	0.00%	0.00	0.00	0.00	66.67%	100.00%	0.00%	S1	0.00%

Chapter 5

Pan-variant COVID-19 Vaccine Challenge Study

In this chapter, we evaluate a vaccine designed using the OptiVax framework (Chapter 4) in an animal challenge study. We adopt a humanized HLA-A*02:01 transgenic mouse model and immunize mice with a T cell vaccine ("MIT-T-COVID") containing the HLA-A*02:01 subset of our human population vaccine design and additional epitopes we predict to be presented by the mouse MHC class II allele H-2-IAb. Our results highlight the importance of the n-times coverage strategy and demonstrate the role of $CD8^+$ and $CD4^+$ T cells in mediating viral clearance following challenge with the SARS-CoV-2 Beta variant.

Current strategies for COVID-19 vaccine design utilize one or more SARS-CoV-2 Spike protein subunits to primarily activate the humoral arm of the adaptive immune response to produce neutralizing antibodies to the Spike receptor binding domain (RBD) (Walsh et al., 2020; Baden et al., 2021). Vaccination to produce neutralizing antibodies is a natural objective, as neutralizing antibodies present an effective barrier to the viral infection of permissive cells by binding to the RBD and thus blocking cellular entry via the ACE2 receptor. However, the strategy of focusing on Spike as the sole vaccine target has proven problematic as Spike rapidly evolves to produce structural variants that evade antibody-based acquired immunity from vaccination or infection with previous viral variants (Tregoning et al., 2021; Willett et al., 2022). Compared

to the original Wuhan variant, novel viral variants are arising that are more contagious (Zhang et al., 2021), and infectious to a broader range of host species (Shuai et al., 2021). Thus, vaccine designers are pursuing a stream of novel Spike variant vaccines. Multivalent Spike vaccines and bivalent booster vaccines provide protection against multiple known variants of concern (VOCs) of SARS-CoV-2 but are not necessarily protective against unknown future variants (Martinez et al., 2021). Mosaic RBD nanoparticles that display disparate SARS-CoV RBDs have been found to produce effective neutralizing antibodies against both SARS-CoV-1 and SARS-CoV-2 (Cohen et al., 2022), but the robustness of mosaic RBD protection against possible future Spike mutations depends upon conserved Spike structural epitopes.

The vaccine approach we present depends upon conserved T cell epitopes drawn from the entire viral proteome for protection against future variants. Since T cell epitopes can originate from any part of the viral proteome, they can be drawn from portions of the proteome that are evolutionarily stable and immunogenic. The predication of epitope stability can be accomplished by historical analysis of thousands of viral variants (Section 4.1.3), structural analysis (Nathan et al., 2021), or the functional analysis of mutations lethal to the virus. We have used a set of highly stable epitope candidates to design a T cell vaccine that covers a broad range human MHC class I and class II haplotypes. Our T cell vaccine design proceeds by vaccine epitope selection that optimizes population coverage where every vaccinated individual is predicted to experience on average multiple immunogenic peptide-HLA hits (Chapter 4).

Code for the experiments in this chapter is available at: `https://github.com/gifford-lab/MIT-T-COVID-vaccine`. Original data collected during this study have been deposited to Mendeley Data: `http://dx.doi.org/10.17632/p4c823jzxz.1`.

5.1 Methods

5.1.1 n-times Coverage Vaccine Design

The MIT-T-COVID vaccine realizes an n-times coverage objective by encoding multiple epitopes for each target MHC class I and II diplotype to (1) expand diverse sets of T cell clonotypes to fight viral infection, (2) accommodate variations in epitope immunogenicity between vaccinees, and (3) reduce the chances that viral evolution will lead to immune system escape (Chapter 4). The MIT-T-COVID vaccine consists of eight MHC class I epitopes and three MHC class II epitopes (Figure 5-1A, Table 5.1, Appendix C.1). The MHC class I and II vaccine peptides are encoded into a single mRNA construct for delivery with the same Acuitas LNP delivery platform that is used by the Pfizer-BioNTech Comirnaty® vaccine (Figure 5-1A). The eight MIT-T-COVID MHC class I epitopes are the HLA-A*02:01 subset of the MHC class I *de novo* MIRA only vaccine design of Liu et al. (2021) that used combinatorial optimization to select vaccine epitopes to maximize n-times population coverage over HLA haplotype frequencies (Chapter 4). For inclusion in the assembled construct, the eight MHC class I vaccine peptides were randomly shuffled, and alternate peptides were flanked with five additional amino acids at each terminus as originally flanked in the SARS-CoV-2 proteome. Selected epitopes were flanked to test if flanking enhanced or impaired epitope presentation. The three MHC class II epitopes were selected by considering all SARS-CoV-2 proteome windows of length 13–25 and selecting conserved peptides that are predicted to be displayed by the mouse MHC class II allele H-2-IAb (details in Appendix C.1). The mRNA construct encodes a secretion signal sequence at the N-terminus and an MHC class I trafficking signal (MITD) at the C-terminus (Kreiter et al., 2008). Peptide sequences are joined by non-immunogenic glycine/serine linkers (Sahin et al., 2017). The construct also included control peptides for HLA-A*02:01 and H-2-IAb (CMV pp65: NLVPMVATV for HLA-A*02:01 and Human CD74: KPVSKMRMATPLLMQAL for H-2-IAb) (Vita et al., 2019).

A

MRVTAPRTLILLLSGALALTETWAGS G G S G G G G S G G **YLYALVYFL** G G S G G G G S G G **RSKNPLLYDANYFLCWHTN**
G G S G G G G S G G **FVDGVPFVV** G G S G G G G S G G **AYYVGYLQPRTFLLKYNEN** G G S G G G G S G G
FLNRFTTTL G G S G G G G S G G **RLTKYTMADLVYALRHFDE** G G S G G G G S G G **SIIAYTMSL** G G S G G G G S G G
LLLFVTVYSHLLLVAAGLE G G S G G G G S G G **ATSRTLSYY** G G S G G G G S G G **KTFPPTEPK** G G S G G G G S G G
EEIAIILASFSASTSAFVETVKGLDY G G S G G G G S G G **KSILSPLYAFASEAARVVRSIFSRTL** G G S G G G G S G G
VDYGARFYFYTSKTTVASLINTLNDLGGSGGGGSGG**NLVPMVATV**GGSGGGGSGG**KPVSKMRMATPLLMQAL**GGSLGGGG
SGIVGIVAGLAVLAVVVIGAVVATVMCRRKSSGGKGGSYSQAASSDSAQGSDVSLTA

B

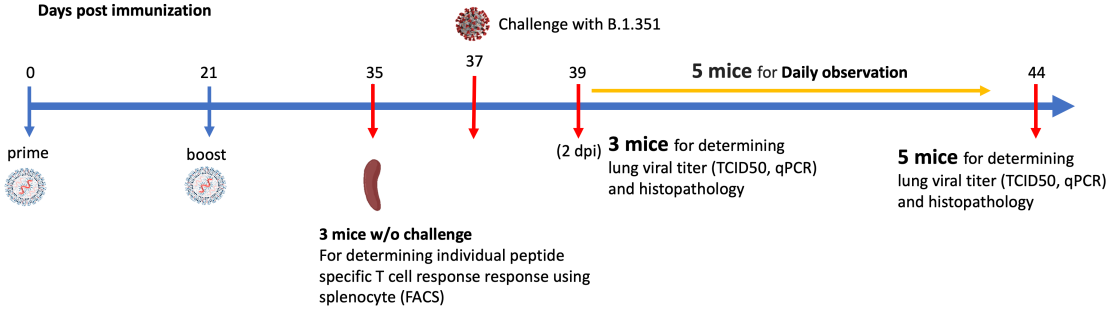


Figure 5-1: (A) Assembled vaccine construct containing a secretion signal sequence (red), peptides (bold) joined by non-immunogenic glycine/serine linkers, and an MHC class I trafficking signal (blue). (B) Study design.

5.1.2 SARS-CoV-2 Beta Variant Challenge Study

We immunized with three test articles: a negative control injection (PBS), the Pfizer-BioNTech Comirnaty® vaccine (wastage that was refrozen and then thawed), and the MIT-T-COVID vaccine (Figure 5-1B). We immunized HLA-A*02:01 human transgenic mice with these three vaccines, immunizing 11 age-matched male mice with each vaccine. Comirnaty® or MIT-T-COVID immunizations contained of 10 µg of mRNA. The mice were immunized at Day 0 and boosted at Day 21 (details in Appendix C.1). At Day 35 three mice from each group were sacrificed for immunogenicity studies, and at Day 37 the remaining eight mice were challenged intranasally (I.N.) with 5×10^4 $TCID_{50}$ of the SARS-CoV-2 B.1.351 variant. Challenged mice were subjected to daily monitoring for the onset of morbidity (i.e., weight changes and other signs of illness) and any mortality. At Day 39 (2 days post infection) three mice were taken from each group and sacrificed to determine viral burdens and to perform lung histopathology. At Day 44 (7 days post infection) the remaining mice were sacrificed to determine

viral burdens and to perform lung histopathology.

Additional method details are presented in Appendix C.

Table 5.1: MIT-T-COVID vaccine peptides, query peptides (vaccine immunogenicity), peptide origin, and appearance probability in HLA-A*02:01 convalescent COVID-19 patients in Snyder et al. (2020) (MIRA column) and Kared et al. (2021) (MCMT). Selected peptides were tested by other studies for their immunogenicity in convalescent COVID-19 patients whose HLA type included HLA-A*02:01. The study by Snyder et al. (2020) included 80 HLA-A*02:01 convalescent COVID-19 patients and tested peptides individually or in small pools with the Multiplex Identification of T-cell Receptor Antigen Specificity (MIRA) assay. Query peptides were first filtered to only consider those with predicted HLA-A*02:01 binding affinity less than or equal to 25 nM. The MIRA fraction is the number of individuals positive for a pool containing a query peptide divided by 80. The Kared et al. (2021) study evaluated 16 HLA-A*02:01 convalescent COVID-19 patients by mass cytometry–based multiplexed tetramer (MCMT) staining. The MCMT fraction is the number of individuals positive for a query peptide divided by 16.

MHC Class	Vaccine Peptide	Query Peptide	Query ID	Organism	Gene	Vaccine Peptide Start–End	Query Peptide Start–End	MIRA	MCMT
1	YLYALVYFL	YLYALVYFL	CD8-1	SARS-CoV-2	ORF3a	107–115	107–115	0.86	0.19
1	RSKNPLLYDANYFLCWHTN	LLYDANYFL	CD8-2	SARS-CoV-2	ORF3a	134–152	139–147	0.7	0.44
1	FVDGVPFVV	FVDGVPFVV	CD8-3	SARS-CoV-2	ORF1ab	4726–4734	4726–4734	0.68	
1	AYYVGYLQPRTFLLKYNEN	YLQPRTFLL	CD8-4	SARS-CoV-2	S	264–282	269–277	0.66	0.62
1	FLNRFTTTL	FLNRFTTTL	CD8-5	SARS-CoV-2	ORF1ab	3482–3490	3482–3490	0.53	
1	RLTKYTMADLVYALRHFDE	TMADLVYAL	CD8-6	SARS-CoV-2	ORF1ab	4510–4528	4515–4523	0.44	
1	SIIAYTMSL	SIIAYTMSL	CD8-7	SARS-CoV-2	S	691–699	691–699	0.59	
1	LLLFVTVYSHLLLVAAGLE	TVYSHLLLV	CD8-8	SARS-CoV-2	ORF3a	84–102	89–97	0.61	
1	ATSRTLSYY	ATSRTLSYY	CD8(−)-1	SARS-CoV-2	M	171–179	171–179		
1	KTFPPTEPK	KTFPPTEPK	CD8(−)-2	SARS-CoV-2	N	361–369	361–369		
1	NLVPMVATV	NLVPMVATV	A0201 (+)	CMV	pp65	495–503	495–503		
2	KSILSPLYAFASEAARVVR SIFSRTL	PLYAFASEAARVVRSI	CD4-1	SARS-CoV-2	ORF1ab	527–552	532–547		
2	VDYGARFYFYTSKTTVASL INTLNDL	RFYFYTSKTTVASLIN	CD4-2	SARS-Cov-2	ORF1ab	1416–1441	1421–1436		
2	EEIAIILASFSASTSAFVE TVKGLDY	ILASFSASTSAFVETV	CD4-3 (Female cohort only)	SARS-CoV-2	ORF1ab	471–496	476–491		
2	KPVSKMRMATPLLMQAL	KPVSKMRMATPLLMQAL	IAb (+)	Homo sapiens	CD74	102–118	102–118		

5.2 Results

5.2.1 MIT-T-COVID vaccine expands CD8$^+$ and CD4$^+$ SARS-CoV-2 specific T cells

Immunization with MIT-T-COVID vaccine expanded CD8$^+$ and CD4$^+$ T cells that expressed interferon gamma (IFN-γ) or tumor necrosis factor alpha (TNF-α) when queried by vaccine epitopes (Figures 5-2A–B). The observed variability of immunogenicity of MIT-T-COVID epitopes in convalescent COVID patients (Table 5.1) and in the present study supports our usage of multiple epitopes per MHC diplotype for n-times coverage. Immunization by Comirnaty® produced no significant T cell responses to vaccine epitopes (including a Spike epitope) when compared to PBS. CD8$^+$ T cells that are activated by the CD8-4 epitope (YLQPRTFLL, Spike 269–277) are expanded in animals immunized with the MIT-T-COVID vaccine (IFN-γ 1.32% $\pm$ 0.53% of CD8$^+$ T cells, $P = 0.0087$ vs. PBS; TNF-α 0.38% $\pm$ 0.09% of CD8$^+$ T cells, $P = 0.0149$ vs. PBS; Figure 5-2A). Similarly, the CD8-8 epitope (TVYSHLLLV, ORF3a 89–97) activated CD8$^+$ T cells that are expanded by the MIT-T-COVID vaccine (IFN-γ 0.60% $\pm$ 0.02% of CD8$^+$ T cells, $P = 0.001$ vs. PBS; TNF-α 0.25% $\pm$ 0.12% of CD8$^+$ T cells, $P = 0.015$ vs. PBS) and the CD4-2 epitope (RFYFYTSKTTVASLIN, ORF1ab 1421–1436) activated CD4$^+$ T cells are expanded by the MIT-T-COVID vaccine (TNF-α 0.078% $\pm$ 0.027%, $P = 0.058$ vs. PBS, $P = 0.010$ vs. Comirnaty®, IFN-γ not significant, Figure 5-2B). We also measured interleukin-2 (IL-2) expression and found the MIT-T-COVID vaccine significantly expanded CD4$^+$ T cells activated by the SARS-CoV-2 CD4$^+$ pool ($P = 0.0015$ vs. PBS, $P = 0.0013$ vs. Comirnaty®, Figure C-11). The lack of HLA-A*02:01 transgenic animal response to certain CD8$^+$ epitopes that were immunogenic in patients (Table 5.1) is consistent with past studies of transgenic mouse models (Kotturi et al., 2009), the variability of immunogenicity of epitopes between in-bred mice (Croft et al., 2019), and the potential of immunodominance of the immunogenic epitopes.

We immunized an additional cohort of female HLA-A*02:01 transgenic mice (Ap-

pendix C.1), and results are shown in Figure C-8. We also immunized female mice with synthetic peptides mixed with poly IC adjuvant and found no significant SARS-CoV-2 specific T cell responses compared to PBS controls (Figures C-9A–B, Appendix C.1). We found a significant increase in the number of effector and memory $CD8^+$ $CD44^+$ T cells in MIT-T-COVID and Comirnaty®-immunized mice, compared to those immunized with Peptide/poly IC or PBS (Figure C-9C).

T cells that lack IFN-γ responses can produce effective immune mediators (Nakiboneka et al., 2019). We found that the fraction of $CD4^+$ T cells that are $Foxp3^+$, designated regulatory T cells (Treg), were not expanded in Comirnaty® or MIT-T-COVID-immunized animals, suggesting that Treg cells that could induce tolerance were not expanded by immunization (Figure 5-2C).

5.2.2 MIT-T-COVID attenuates morbidity and prevents mortality

Upon viral challenge, both PBS and MIT-T-COVID-immunized animals exhibited a more than 10% weight loss by Day 3 (PBS mean weight reduction 11.386% ± 1.688%, MIT mean weight reduction 10.851% ± 0.641%), with the MIT-T-COVID-immunized animals beginning to recover from Day 4 onward (Figure 5-3A). The weight phenotype was mirrored by the clinical score phenotype, with PBS animals not recovering and MIT-T-COVID-immunized animals improving from Day 5 onward (Figure 5-3B, details in Appendix C.1). The Comirnaty® vaccine protected animals from significant weight loss and poor clinical scores.

When the yields of infectious progeny virus were measured at 2 days post infection (dpi), we noted that mice immunized with MIT-T-COVID vaccine had 6.706 ± 0.076 $\log_{10}$ $TCID_{50}$/g, compared to 7.258 ± 0.367 $\log_{10}$ $TCID_{50}$/g in the PBS control, representing a moderate 3.6-fold reduction in viral replication ($P = 0.046$; Figure 5-3D). In contrast, mice immunized with Comirnaty® had infectious viral titers that were below the detection limits at 2 dpi. Infectious viral titer was not significant for all test articles at 7 dpi (Figure 5-3D). Lung viral mRNA levels measured by qPCR

followed the same trends as infectious virus progeny in the lungs, with Comirnaty® showing no significant levels. Despite a reduced content of total and sub-genomic viral RNAs associated with MIT-T-COVID vaccine samples as assessed by qPCR, the reduction was insignificant, compared to those of PBS control (Figure C-2).

All the Comirnaty® and MIT-T-COVID-immunized mice survived to 7 dpi, when the study was terminated, with the difference in clinical scores of these two vaccine groups becoming insignificant (Figure 5-3C). In contrast, four of five PBS control mice had been euthanized because of weight loss $> 20\%$. The survival of all five animals immunized with Comirnaty® and the MIT-T-COVID, respectively, was significant, compared to that of PBS-immunized control ($P = 0.0053$, logrank test).

We noted that mice immunized with the Comirnaty® vaccine elicited substantial specific antibodies capable of neutralizing Beta and WA-1 variants of SARS-CoV-2 with 100% neutralizing titers (NT_{100}) of 896 ± 350 and fold 2048 ± 1887, respectively (Figure 5-3E). The higher neutralizing titer against WA-1 is expected as it is the strain matched to Comirnaty®. As expected for a peptide-based vaccine, no neutralizing titer could be readily detected in the serum from mice immunized with MIT-T-COVID vaccine or PBS. Total specific IgG/IgM antibodies were also measured by ELISA against a cell lysate prepared from WA-1-infected Vero E6 cells. Comirnaty® has a $\log_{10}$ IgG/IgM titer of 4.915 ± 0.213, while the $\log_{10}$ titer produced by PBS was 2.452 ± 0.321 and the MIT-T-COVID vaccine produced a $\log_{10}$ titer of 2.266 ± 0.291 (Figure 5-3F). The low but detectable titers in PBS and MIT-T-COVID vaccine-immunized mice may represent an early IgM response to viral infection.

5.2.3 MIT-T-COVID increases T cell infiltration of infected lungs

All lung samples were subjected to immunohistochemistry (IHC) staining for the SARS-CoV-2 spike protein (Figure C-3). We found specimens immunized with PBS exhibited extensive staining indicative of viral infection throughout the epithelium of both the bronchioles and the alveolar sacs, with the viral infection appearing more in-

tense at 2 dpi. Although viral infection is significantly reduced by 7 dpi, viral antigen was still readily detectable throughout alveoli. In comparison, specimens immunized with the MIT-T-COVID vaccine exhibited similarly extensive viral infection at 2 dpi throughout the bronchiolar and alveolar epithelia, albeit somewhat reduced in intensity. However, by 7 dpi, viral infection was significantly reduced in both extent and intensity, with brown puncta being detected only in a few alveoli scattered throughout the tissue. Contrasted with both PBS and MIT-T-COVID-immunized specimens, the Comirnaty®-immunized specimens exhibited significantly reduced viral infections at both 2 and 7 dpi. Apart from a single area at 7 dpi (see Figure C-4), viral antigen was undetected at both timepoints in Comirnaty®-immunized animals.

Paraffin-embedded and H&E-stained lung specimens of differentially immunized mice, harvested at 2 and 7 dpi, were subjected to histopathological examination (Figure C-5). We found that at 7 dpi mice immunized with MIT-T-COVID vaccine exhibited extensive lymphocytic infiltrations in perivascular regions and spaces from around bronchi, bronchioles, to alveoli. Fewer infiltrations were found in mice immunized with either Comirnaty® or PBS. Additionally, these infiltrations only localized at perivascular regions around bronchi and large bronchioles. Despite the less intensive and localized inflammatory infiltrates, we also noted widespread congestion, hemorrhage, and few foci of thromboembolism were exclusively observed within the lungs of Comirnaty®-immunized mice but not others (Figure C-5). We also noted the lung histopathology was milder but the same pattern at 2 dpi than those of 7 dpi (data not shown).

Lung specimens were subjected to IHC staining for $CD8^+$ and $CD4^+$ cells at both 2 and 7 dpi (Figures 5-4, C-6, and C-7). At 2 dpi, we found a significant increase in $CD8^+$ T cells infiltrating the lungs in mice immunized with MIT-T-COVID ($12.6\% \pm 5.91\%$ of all nucleated cells were $CD8^+$) or Comirnaty® ($12.4\% \pm 2.35\%$) compared to mice immunized with PBS (PBS mean $4.48\% \pm 1.99\%$; $P = 0.044$ vs. MIT-T-COVID, $P = 0.050$ vs. Comirnaty®; Figure 5-4A). At 7 dpi, we found a significant increase in $CD8^+$ T cells infiltrating the lungs in mice immunized with MIT-T-COVID ($24.0\% \pm 5.21\%$ of all nucleated cells were $CD8^+$) compared to mice immunized with

Comirnaty® (Comirnaty® mean 4.35% ± 2.20%, $P = 0.001$) or PBS (PBS mean 7.32% ± 4.88%, $P = 0.001$; Figure 5-4A). We observed a significant increase in $CD8^+$ T cells infiltrating the lungs in mice immunized with MIT-T-COVID between 2 and 7 dpi ($P = 0.0039$), and a significant decrease in $CD8^+$ T cells infiltrating the lungs in mice immunized with Comirnaty® between 2 and 7 dpi ($P = 0.044$; Figure 5-4A). At 7 dpi, we also found a significant increase in $CD4^+$ T cells infiltrating the lungs in mice immunized with MIT-T-COVID (7.09% ± 4.05%) compared to mice immunized with PBS (PBS mean 0.41% ± 0.39%; $P = 0.0062$; Figure 5-4B). In the unchallenged cohort of female mice, we found an increase in $CD8^+$ T cells infiltrating the lungs in mice immunized with MIT-T-COVID (1.12% ± 0.59% of all nucleated cells were $CD8^+$) or Comirnaty® (0.87% ± 0.56%) compared to mice immunized with PBS (0.09% ± 0.10%; $P = 0.001$ vs. MIT-T-COVID, $P = 0.003$ vs. Comirnaty®; Figure C-10).

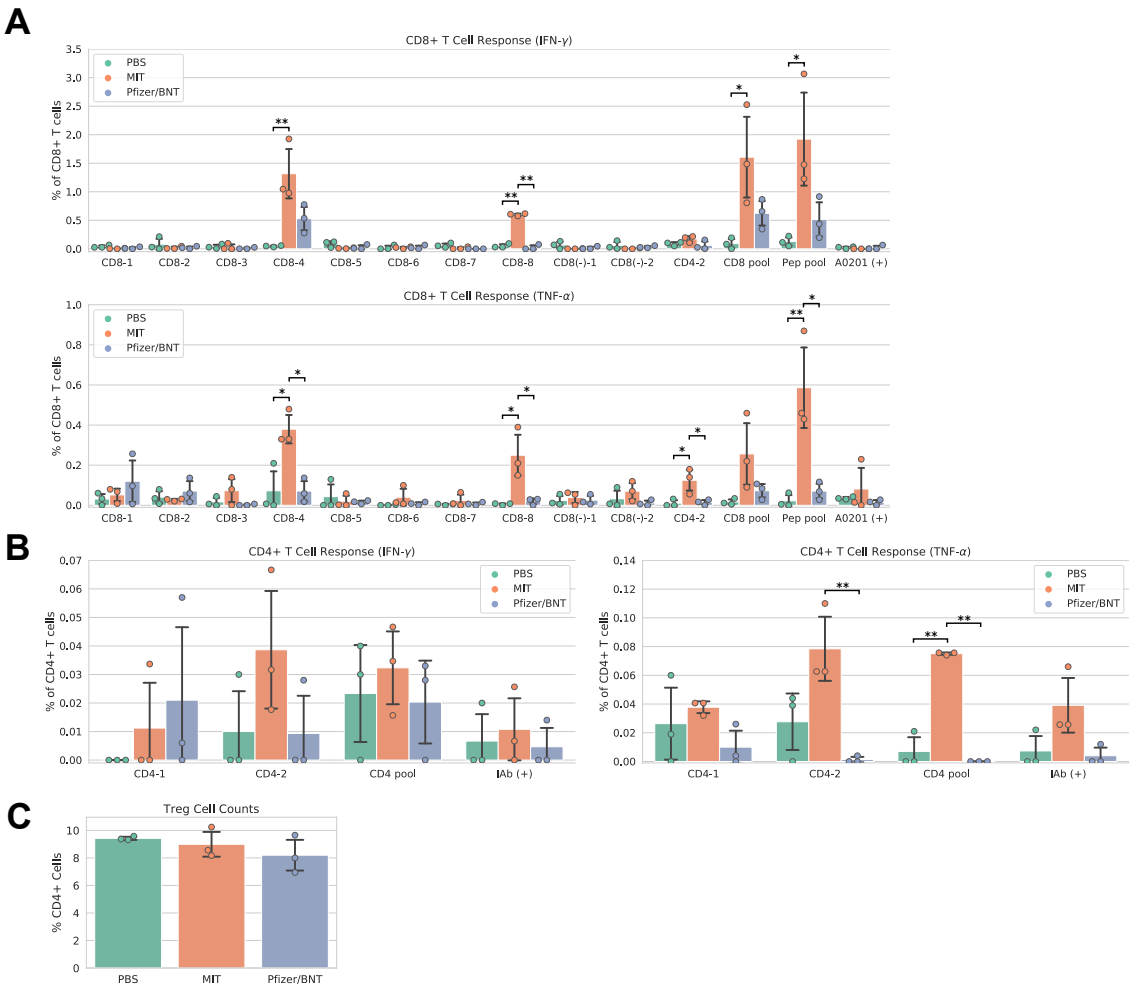


Figure 5-2: Vaccine immunogenicity. (A) CD8$^+$ T cell responses, (B) CD4$^+$ T cell responses, (C) Foxp3$^+$ regulatory T cells (Tregs) as a percentage of all CD4$^+$ cells. The CD8 pool includes MHC class I peptides CD8-1–CD8-8 (Table 5.1). The CD4 pool includes MHC class II peptides CD4-1 and CD4-2. The Pep pool includes all query peptides in Table 5.1 except CD4-3. Error bars indicate the standard deviation around each mean. P values were computed by one-way ANOVA with Tukey's test. $^*P < 0.05$, $^{**}P < 0.01$. See also Figure C-11.

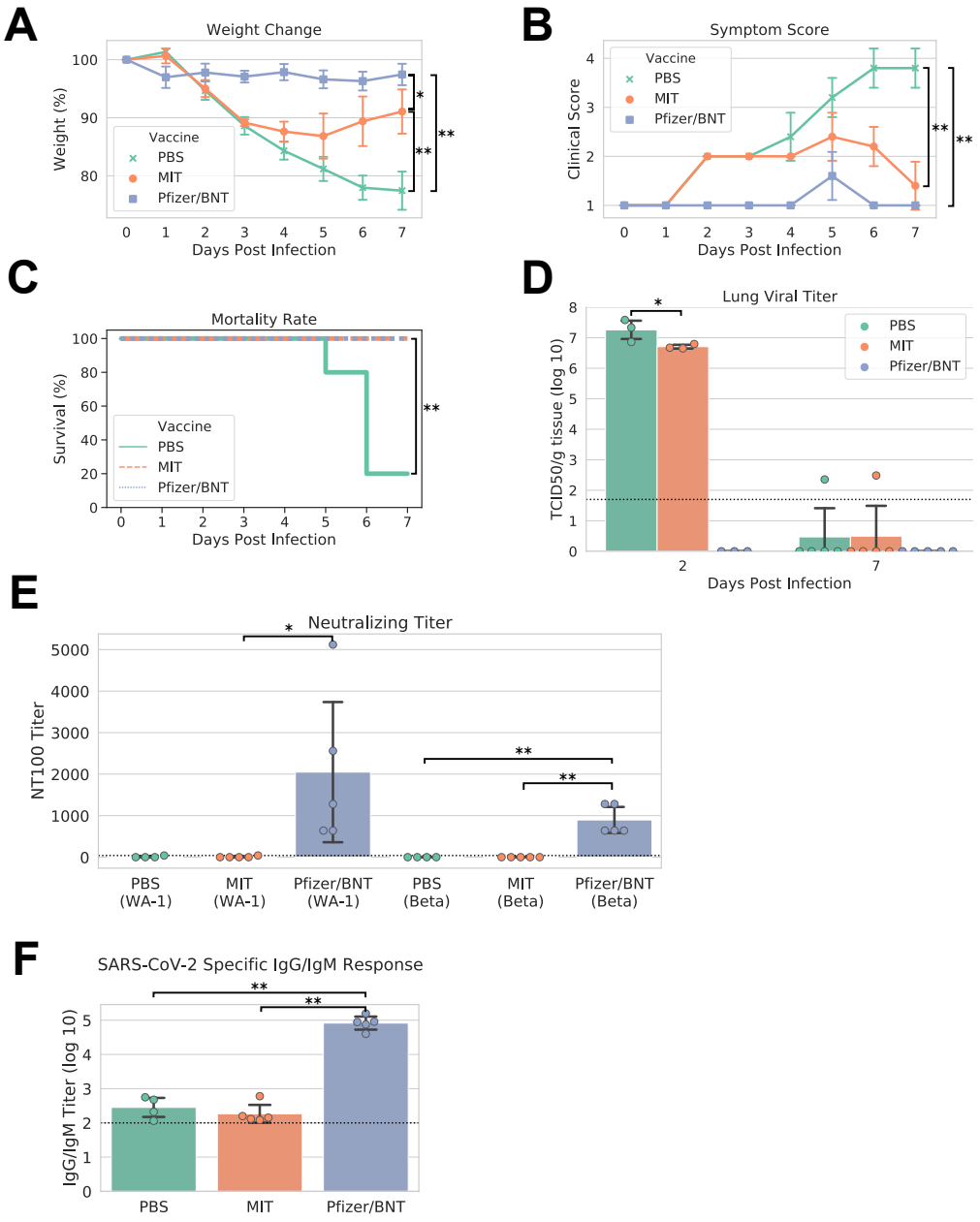


Figure 5-3: Study phenotypic data, lung viral titer, and vaccine antibody responses. (A) Weights vs. days post infection, (B) clinical scores vs. days post infection, (C) Kaplan-Meier mortality curve (mortality at 80% weight loss), (D) lung viral titer, (E) maximum serum dilution that provided 100% neutralization of viral infection in vitro, (F) IgG/IgM titer measured by ELISA against cell lysate infected with WA-1 SARS-CoV-2. Dotted lines in (D)–(F) indicate assay limits of detection. Error bars indicate the standard deviation around each mean. P values were computed by one-way ANOVA with Tukey's test except (C) P values were computed using the logrank test. $^{*}P < 0.05$ and $^{**}P < 0.01$.

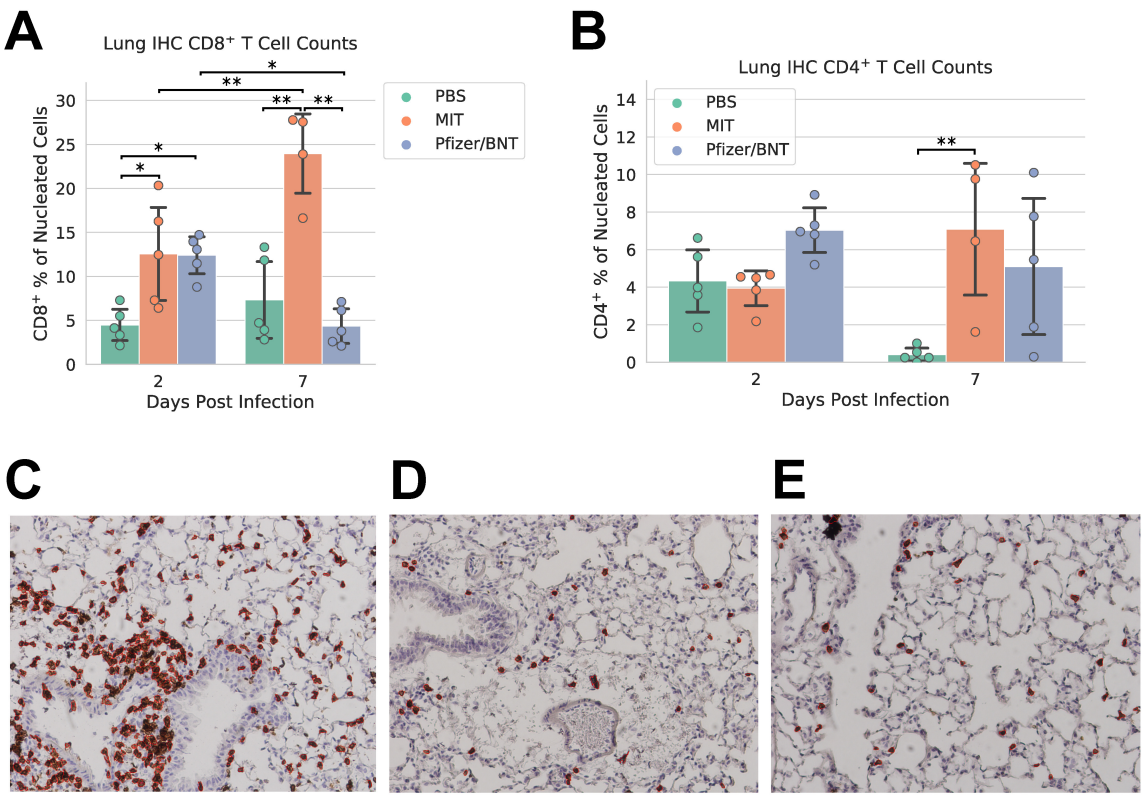


Figure 5-4: Lung immunohistochemistry for $CD8^+$ and $CD4^+$ cells. Counts of (A) $CD8^+$ and (B) $CD4^+$ T cells expressed as a percentage of all nucleated cells visible in each field from lung tissue. Example $CD8^+$ stain images at 7 dpi for (C) MIT-T-COVID, (D) Pfizer/BNT, and (E) PBS-immunized animals. Lung samples were subjected to IHC staining for CD8 (brown) with hematoxylin counterstain (blue). Images were taken at 10x magnification. Red outlines in (C)–(E) indicate $CD8^+$ cells identified and counted by CellProfiler software (Appendix C.1). Error bars indicate the standard deviation around each mean. P values were computed by two-way ANOVA with Tukey's test. $^*P < 0.05$ and $^{**}P < 0.01$. See also Figures C-6 and C-7.

5.3 Discussion

Here we find that a T cell vaccine ("MIT-T-COVID") that contains the human HLA-A*02:01 displayed subset of our COVID-19 T cell vaccine and additional mouse specific $CD4^+$ epitopes provides effective prophylaxis against the onset of SARS-CoV-2-induced morbidity and mortality caused by SARS-CoV-2 Beta infection in transgenic mice carrying HLA-A*02:01. The vaccine consists of 11 short T cell epitopes that are unchanged over 22 presently known SARS-CoV-2 variants of concern (VOCs) (Appendix C.1). We further demonstrate the MIT-T-COVID vaccine causes significant infiltration of $CD8^+$ and $CD4^+$ T lymphocytes in the lungs post infection. We chose the Beta variant for our challenge study as it is more pathogenic than recent variants (Halfmann et al., 2022) and its host range includes wild type mice (Shuai et al., 2021) since murine infection with SARS-CoV-2 variants that require ectopic human ACE2 expression results in fatal encephalitis (Kumari et al., 2021) which is not ideal for evaluating the efficacy of a T cell vaccine. We selected a highly pathogenic variant that models human disease to evaluate the effectiveness of our T cell vaccine in anticipation of possible future SARS-CoV-2 variants that result in severe disease. MIT-T-COVID vaccine epitopes delivered as peptides with a Poly(I:C) adjuvant failed to induce significant immune responses, supporting the value of mRNA-LNP delivery of the epitopes.

Existing Spike vaccines produce a T cell response that is thought to be important for vaccine effectiveness and durability (Moss, 2022). However, the T cell response induced by a Spike vaccine may be less effective in promoting durable immune responses than the T cell response induced by a pure T cell vaccine. Spike based T cell responses may contain epitopes that are subject to evolutionary change (de Silva et al., 2021; Redd et al., 2021) that dominate stable epitopes, or Spike epitopes may not be as well presented or immunogenic as epitopes drawn from a more diverse set of SARS-CoV-2 proteins. Thus, T cell augmentation strategies for Spike vaccines (Liu et al., 2021) and pure T cell vaccines will require further exploration to unravel the hierarchy of immune responses to their components, and how to structure a vaccine for

optimal prophylaxis. In patients with impaired antibody responses, T cell vaccines would eliminate the burden of non-immunogenic B cell epitopes (Benjamini et al., 2022) and provide immune protection, at least to a certain extent, against infection.

Multiple designs for T cell vaccines for SARS-CoV-2 have been proposed (Liu et al., 2021; Nathan et al., 2021; Heitmann et al., 2022; Pardieck et al., 2022) and are in clinical trials (NCT05113862, NCT0488536, NCT05069623, NCT04954469), but identifying the mechanisms behind the efficacy of pure T cell vaccines remains an open question. Substantial literature suggests that T cell responses are integral to the adaptive immunity to COVID-19 (Moss, 2022; Geers et al., 2021). For example, a study that ablated the B cell compartment of the immune system in Spike immunized mice found that $CD8^+$ T cells alone can control viral infection (Israelow et al., 2021). Pardieck et al. (2022) found that vaccination with a single mouse restricted $CD8^+$ T cell epitope conferred protection against mortality from the Leiden-0008/2020 SARS-CoV-2 variant (B.1) in K18-hACE2 transgenic mice, but unlike our study, required three doses for efficacy, did not engage a $CD4^+$ T cell response, did not identify significant T cell infiltration of the lungs, challenged with a lower viral dose (5000 PFU vs. our 5×10^4 $TCID_{50}$), and used a variant of SARS-CoV-2 that is not pathogenic in wild type mice (Shuai et al., 2021). In addition to specific antibody responses, COVID-19 vaccinees and convalescent patients possess SARS-CoV-2 specific $CD8^+$ and $CD4^+$ T cells, suggesting the contribution of the T cell compartment to the adaptive immunity to COVID-19 (Sekine et al., 2020), and clinical findings have revealed vaccine-induced T cell responses in B cell-deficient patients (Shree, 2022). It has also been reported that vaccination by WA (Wuhan) Spike in a mouse model failed to produce antibodies fully capable of neutralizing the SA (Beta) variant of SARS-CoV-2, yet immunized mice were protected against Beta strain challenge (Kingstad-Bakke et al., 2022). In addition, vaccination with T cell epitope-rich Nucleocapsid protein produced specific T cell responses thought to be causally associated with viral control (Matchett et al., 2021). Intranasal vaccination of mice with SARS-CoV-1 Nucleocapsid followed by challenge with 10^4 Plaque-forming units of SARS-CoV-1 prevented mortality in 75% of the mice (Zhao et al., 2016). Combined Spike and

Nucleocapsid vaccination improved viral control compared to Spike vaccination alone in preclinical models, while $CD8^+$ T cell depletion demonstrated the role of $CD8^+$ T cells in viral control and protection from weight loss (Hajnik et al., 2022).

The marginal IgG/IgM antibody titers that were not neutralizing elicited by mice immunized with the MIT-T-COVID vaccine indicates that the protective mechanism of the vaccine was likely based on a T cell response to the virus. The reduction in viral titer on day 2 shows that T cell responses were present early in infection. In the absence of neutralizing antibodies these responses were sufficient to rescue the immunized mice from the onset of mortality.

We expect that a T cell vaccine would provide prophylaxis, at least to certain extent, like what we have described against future SARS-CoV variants and strains that conserve the vaccine's epitopes. We chose Beta (B.1.351) for our challenge study as it has a severe phenotype in a wild-type mouse background (Shuai et al., 2021). Transgenic human ACE2 (hACE2) mice exhibit encephalitis post COVID infection that does not represent human pathology (Kumari et al., 2021) and thus might not be an ideal model for assessing the efficacy of T cell-based vaccines. Instead, we chose to evaluate the HLA-A*02:01 component of our vaccine design in a HLA-A*02:01 transgenic animal model given the predominance of HLA-A02 in the human population (Ellis et al., 2000).

The MIT-T-COVID vaccine induced a response where circulating T cells migrated rapidly and efficiently into the lung upon viral challenge, as evidenced by more intense and widespread lymphocytic infiltrations than other groups. Further, the degree of $CD8^+$ T cell infiltration of the lungs significantly increased in mice immunized with the MIT-T-COVID vaccine between days 2 and 7 post infection and compared to that elicited by unimmunized and challenged cohorts, thereby thwarting the concern over the involvement of resident T cells of the lung.

Another finding is that widespread congestion, hemorrhage, and few foci of thromboembolism were exclusively observed within the lungs of Comirnaty®-immunized mice. Although this finding is unrelated to the MIT-T-COVID vaccine, it suggests the possibility of immunization-induced side-effects or immunopathology of SARS-

CoV-2 vaccination and is consistent with previous reports in humans (Pavord et al., 2021; de Oliveira et al., 2022). The exact mechanism of pulmonary embolism in this case remains currently unknown and should be explored.

Cytotoxic T cells ($CD8^+$ T cells) can kill virally infected cells, and thus T cell vaccines might promote the long-term immunity to more effectively control Long COVID. Post-acute sequelae of COVID-19 (PASC, "Long COVID") causes persistent symptoms in $\sim$10% of people past twelve weeks of COVID infection (Rajan et al., 2021). Long COVID has been associated with the continued presence of Spike in the blood, suggesting that a tissue based viral reservoir remains in Long COVID patients (Swank et al., 2023).

Our results suggest that if one goal of a vaccine is protection against novel viral strains it may be appropriate to develop vaccines that permit symptomatic infection but protect against severe illness. Current regulatory criteria for licensing vaccines are solely based upon the prevention of symptomatic illness for vaccine matched viral strains. Further research on T cell vaccines may reveal novel vaccine designs that prevent severe illness while providing protection against viral variants as an additional tool in the fight against global pandemics.

Chapter 6

Discussion

The application of machine learning (ML) to biological problems often involves (1) training accurate models on data from biological experiments, where it is important that the models learn the underlying biological mechanisms to generalize to unseen data, and (2) the application of these models to a downstream task, such as antibody design or vaccine design. In this thesis, we introduce frameworks for ML model interpretability and for the computational design of peptide vaccines. Our model interpretability framework (SIS) can be used to evaluate black-box ML models, including those trained on biological data. Our vaccine design framework (OptiVax) adopts a ML model that predicts peptide-HLA binding and introduces a combinatorial optimization problem for the downstream task of peptide vaccine evaluation and design. Both framework designs are *flexible*, *faithful* to their respective underlying systems, and *conceptually straightforward* to facilitate their adoption.

In the Sufficient Input Subsets (SIS) interpretability framework (Chapter 2), we interpret black-box model decisions through minimal input patterns that alone provide sufficient evidence to justify a particular decision. We identify sufficient input subsets through a straightforward backward selection strategy applied locally to individual inputs. Intuitively, this procedure iteratively masks the least informative features whose removal causes the least loss of predicted confidence, preserving the most important features and preserving combinations of features present in the input that provide support for the prediction in combination (whereas the features may be

uninformative when presented independently). Interestingly, we also found that the confidence loss over the backward selection path may be non-monotonic, and thus to identify minimal sufficient input subsets, we remove and rank all features during backward selection and then build a minimal SIS in the reverse order. Our SIS framework can be broadly applied to a wide range of model classes, including set-valued inputs, and is faithful to the underlying models without requiring differentiability, additional training steps, or an auxiliary explanation model. Further, our framework design permits the flexible choice of decision confidence threshold and input mask, and SIS can be extended to accommodate sampling-based imputation methods (Rubin, 1976).

During evaluation of interpretation methods, it is important to avoid biasing results toward human priors of the ideal explanations or assume that the model is making decisions using semantically meaningful, rather than spurious, features. In our evaluation of SIS on natural language that measures alignment of rationales with human annotations, we first computed a metric that captures whether the human rationales contain sufficient predictive information on their own, before computing concordance between the SIS and human rationales. We also showed how the local explanations on individual decisions can be aggregated over many examples to gain insight into the model's global decision-making process and to contrast the behavior of different models trained on the same task. Given multiple models of comparable accuracy, we found that SIS-clustering can uncover critical operating differences, such as which model is more susceptible to spurious training data correlations or may generalize worse to counterfactual inputs that lie outside the data distribution.

Central to our SIS framework is the guarantee that each rationale provides sufficient evidence for confident prediction in absence of the rest of the features. Like Lei et al. (2016), we posit that rationales should be minimal and sufficient. Intriguingly, we found that rationales comprising salient features from widely used interpretability methods often do not contain sufficient support for confident classifications, or must contain a larger subset of features to meet the sufficiency threshold. Compared to alternative interpretability methods, we found SIS more effectively identified subsets of features that are both minimal *and* sufficient for confident classification. On bio-

logical data where the ground-truth for the model's behavior was known, we found subsets identified by SIS more accurately matched the underlying transcription factor motifs compared to other methods. Thus, we expect another use of our SIS framework is for scientific discovery — accurate ML models trained on experimental data can be interpreted to uncover the underlying principles. We applied SIS to deep neural networks that predict antibody-antigen binding enrichment, and SIS provided hypotheses about antibody CDR-H3 binding signatures that can be explored in future structural studies (Liu et al., 2020b). We also demonstrated comparing SIS interpretations to biological motifs in Carter et al. (2020). We used SIS to critique and compare deep neural networks trained to predict protein function from sequence (Bileschi et al., 2022), and SIS revealed differences in decision-making by different model classes.

Our SIS framework design also permits the flexible selection of SIS-finding algorithm. While the local backward selection strategy presented in Chapter 2 finds a SIS solely through black-box model evaluations, one limitation of this approach is its computational efficiency. Given p input features, the algorithm scales as $\mathcal{O}(p^2)$. While our method performed well on the datasets we evaluated in Chapter 2 (up to approximately 1000 input features), this procedure may be prohibitively expensive for high-dimensional inputs. To overcome this limitation, we introduced the Batched Gradient SIS algorithm (Chapter 3) to find SIS on high-dimensional inputs, including ImageNet images.

We applied SIS to popular image classifiers and discovered *overinterpretation*, a novel failure mode of ML models that make confident predictions based upon nonsensical patterns present in benchmark datasets. A model that overinterprets spurious feature subsets may fail to generalize outside of the benchmark setting. Importantly, despite their lack of semantically meaningful features, we found these sparse pixel-subsets are indeed underlying statistical signals that suffice to accurately generalize from the benchmark training data to the benchmark test data. Thus, our results suggest that overinterpretation is caused by spurious statistical signals present in benchmark datasets, including CIFAR-10 and ImageNet. We found that different models rationalize their predictions based on different sufficient input subsets, sug-

gesting optimal image classification rules remain underdetermined by the training data. Our results suggest that ensembles of networks or regularization through input dropout can each mitigate overinterpretation and increase a model's generalizability to unseen data.

Our overinterpretation results call into question model interpretability methods whose outputs are encouraged to align with prior human beliefs of proper classifier operating behavior (Adebayo et al., 2018). Given the existence of non-salient pixel-subsets that alone suffice for correct classification, a model may solely rely on such patterns. In this case, an interpretability method that faithfully describes the model should output these nonsensical rationales, whereas interpretability methods that bias rationales toward human priors may produce results that mislead users to think their models behave as intended. The SIS method permitted the discovery of overinterpretation by identifying rationales that were faithful to the underlying model yet nonsensical to humans, demonstrating one way model interpretation can be used to evaluate ML models prior to deployment. Separately, we applied the SIS methodology to compare the features informative to pruned neural networks to those informative to their unpruned counterparts, and interpretation revealed weight-pruned networks preserve parent features better than filter-pruned networks (Liebenwein et al., 2021).

In Chapter 4, we introduced the EvalVax and OptiVax framework for the computational evaluation and design of peptide vaccines. Our n-times coverage objective computes the fraction of the population predicted to be covered by at least n peptide-HLA hits. The presentation of multiple peptides by HLA molecules in a given individual (1) engages additional T cell clonotypes for a stronger cellular immune response, (2) permits differences in peptide immunogenicity across individuals to increase the probability that at least one vaccine peptide will be immunogenic, and (3) reduces the chance of vaccine escape by pathogenic drift. Thus, we expect vaccines designed using this framework will elicit a more robust cellular immune response and can confer pan-variant immunity against mutable pathogens. Our OptiVax framework permits both *de novo* vaccine design and augmentation of subunit vaccines to fill population coverage gaps, as we further explored in Liu et al. (2021). Our framework is focused

on the design of optimal compact vaccine payloads, and the resulting vaccine sets can be delivered by any suitable delivery platform, such as mRNA encapsulated in lipid nanoparticles (LNPs) as used in our COVID-19 challenge study (Chapter 5).

One challenge for the computational design of vaccines is the imperfect prediction of immunogenic epitopes. Ideally, the ML model used by our framework would predict peptide-HLA immunogenicity, but immunogenicity also depends upon each individual's randomized T cell receptor (TCR) repertoire, and thus peptide immunogenicity may differ across individuals with the same HLA diplotype. In absence of these models, we adopt ML models that predict peptide-HLA binding affinity, as peptide-HLA binding is a prerequisite for peptide-HLA presentation and T cell activation, and higher peptide-HLA affinity is correlated with immunogenicity (Sette et al., 1994). We calibrated these ML models using available SARS-CoV-2 experimental data and chose strict thresholds of predicted affinity to maximize precision, and we use ensembles to mitigate idiosyncratic errors of individual networks. Our framework permits the integration of experimental or clinical immunogenicity data where available in place of ML predictions, as we demonstrate in Liu et al. (2021).

Another challenge addressed by our framework is the combinatorial selection of a compact set of vaccine peptides from a large set of candidates to maximize population coverage. We used HLA haplotype frequencies to compute n-times population coverage across three different ancestries self-reporting as Black, White, or Asian, and we found that our greedy optimization procedure with beam search effectively designed compact vaccines for SARS-CoV-2. Our framework can be used for the design of vaccines targeting a specific haplotype distribution. Our framework design also permits the flexible choice of optimizer, and the n-times coverage objective can alternatively be cast as a probabilistic model (Dai and Gifford, 2023) or an integer linear program (Liu et al., 2022; Dimitrakakis, 2021).

In Chapter 5, we validated the n-times coverage framework through an animal challenge study. We immunized HLA transgenic mice with a T cell vaccine ("MIT-T-COVID") containing short peptide epitopes that are conserved over 22 SARS-CoV-2 variants of concern designed using the OptiVax framework. We found our vaccine

elicited robust T cell responses and consequently protected mice from mortality after challenge with the SARS-CoV-2 Beta variant compared to a PBS control. Our results highlight the importance of the n-times coverage vaccine design objective. The MIT-T-COVID vaccine contains eight MHC class I epitopes that were all previously found to activate T cell responses in peripheral blood mononuclear cells (PBMCs) from convalescent COVID-19 patients with HLA-A*02:01. However, we found that only two of the eight MHC class I epitopes were immunogenic in HLA-A*02:01 transgenic mice. The lack of observed immunogenicity of all epitopes is consistent with data from Snyder et al. (2020) that show these specific vaccine epitopes were each only immunogenic in 44–86% of convalescent patient PBMC samples (Table 5.1), reflecting the differences in peptide immunogenicity across individuals, as well as prior studies of HLA transgenic mouse models (Kotturi et al., 2009). For MHC class II, or vaccine included three SARS-CoV-2 epitopes predicted to be displayed by the endogenous mouse allele H-2-IAb, and we found one of three epitopes to be immunogenic in our study. However, despite only a limited subset of immunogenic vaccine peptides observed in mice from the MIT-T-COVID vaccine, our vaccine induced both SARS-CoV-2 specific $CD8^+$ and $CD4^+$ T cell responses that mediated an effective antiviral response in mice challenged with SARS-CoV-2. Together, these data illuminate the need for n-times coverage of vaccine designs to maximize the likelihood that at least one peptide is immunogenic in each individual. Our results suggest that optimal values of n may be chosen based upon the observed immunogenicity of vaccine peptides in experimental screens, the specific HLA alleles covered by a vaccine, and peptide delivery platform constraints.

6.1 Future Work

Model interpretation and its applications in biology provide ripe opportunities for future research. Our results reveal the importance of dataset quality, and noisy or sparse datasets are often a limitation of applying ML in biology. In addition to improving ML methods and datasets, there are many unknowns about the immune

system and biological systems in general. Since we design ML methods to reflect underlying biological mechanisms, learning more about those mechanisms can help develop more accurate computational methods.

Our SIS framework requires the selection of a feature mask that is uninformative to a trained model. However, a limitation of this approach is that a masked input may be out-of-distribution or does not fully destroy all information about each masked feature, and thus the resulting SIS may be sensitive to the choice of mask. Consistent with the approach of Ribeiro et al. (2018), future work can explore a stronger SIS definition that requires the SIS be sufficient for classification in the presence of *any* background, thus eliminating dependence on the choice of mask. Other work could include seeking theoretical guarantees about the subsets found by the SIS backward selection algorithm, which are currently only known for backward selection in certain linear settings (Das and Kempe, 2008).

The discovery of overinterpretation by our sufficient input subsets framework revealed a new failure mode of deep neural networks and benchmark datasets. Mitigating overinterpretation and the broader task of ensuring models are accurate for the right reasons remain significant challenges for ML. While we identified strategies for partially mitigating overinterpretation, additional work is needed to develop ML methods that guide models to rely exclusively on causal and interpretable input features. One approach is to regularize neural networks by constraining saliency attributions (Ross et al., 2017; Simpson et al., 2019; Viviano et al., 2021). However, these methods would require a human annotator to highlight the correct pixels as an auxiliary supervision signal. Our results suggested the presence of many different spurious patterns in a given dataset that a model could rely on, which poses an additional challenge for methods that guide a model to rely on causal features. Further, our work suggests a need for new methods to curate training datasets that do not contain spurious signals in order to train more robust models. Finally, our methods can be extended for the evaluation of benchmark datasets to ensure the benchmarks can accurately measure generalizability.

Our computational vaccine design framework introduced the idea of multiple im-

munogenic peptide-HLA "hits" in each individual as a vaccine design objective. At present, we rely upon imperfect ML models that predict peptide-HLA binding and calibrated the models to maximize precision. While we found these models to work sufficiently well in practice, more accurate models to predict peptide-HLA immunogenicity would permit the design of vaccines with additional hits in each individual, which we expect would further increase vaccine-induced responses. Present training data for ML models predicting peptide-HLA binding are generally restricted to *in vitro* binding affinity measurements or eluted ligands identified by mass spectrometry (Reynisson et al., 2020a), and these data do not consider T cell activation by the resulting peptide-HLA complex. While T cell activation by a peptide-HLA complex may ultimately differ across individuals, it is possible that structural features (Riley et al., 2019) or data from high-throughout library-on-library TCR-pMHC screens (Dobson et al., 2022) could be used to train more accurate ML models and merits further study. It may be challenging to train ML models that generalize as well to rare HLA alleles as present data are skewed toward high frequency HLA alleles, and thus, curation of training data over a diverse range of human samples will be essential.

The COVID-19 pandemic has reinforced the importance of pan-variance in vaccine designs, as SARS-CoV-2 variants have escaped neutralization by vaccine-induced humoral immunity (Garcia-Beltran et al., 2021). To design pan-variant vaccines, we filtered epitopes by mutation rate and other criteria, and filters in our framework can be flexibly chosen to address the specific requirements of a vaccine design. We used a straightforward heuristic that computes mutation rate over presently known variants and only permits highly conserved peptides, and we found our vaccine payload remained conserved in more recent variants of concern. Further work is needed to rigorously evaluate this approach and further develop methods that can accurately predict conserved epitopes across future variants.

Our MIT-T-COVID vaccine activated both $CD8^+$ and $CD4^+$ T cells in immunized mice, and we found a significant increase in both $CD8^+$ and $CD4^+$ T cells infiltrating the lungs of MIT-T-COVID immunized mice compared to unimmunized mice at

7 days post infection. Future work to activate $CD8^+$ and $CD4^+$ T cell responses independently would elucidate the individual roles of $CD8^+$ and $CD4^+$ T cells in mediating viral clearance. The study by Pardieck et al. (2022) solely activated a $CD8^+$ T cell response by vaccination with a single epitope but found that a third dose was necessary to confer vaccine protection, and thus it would be interesting to understand the specific role of $CD4^+$ T cells activated by our vaccine as $CD4^+$ T cells help activate $CD8^+$ T cell responses (Zhang et al., 2009). In addition, our study was limited to viral challenge at 37 days post immunization. Cohen et al. (2021) found that virus-specific T cells had a half-life of roughly 200 days in COVID-19 patients. Thus, further studies can investigate the durability of antiviral T cell responses at longer time intervals between immunization and challenge.

The polypeptide construct encoded by the MIT-T-COVID vaccine links vaccine peptides with flexible linkers to facilitate proteasomal cleavage, which may introduce an additional failure point if peptides are not cleaved at the appropriate positions. The lack of any immunogenic SARS-CoV-2 epitopes in mice immunized with synthetic peptides mixed with poly IC adjuvant supports our hypothesis that the lack of immunogenicity of all MIT-T-COVID epitopes is the result of an insufficient TCR repertoire in the transgenic mice, rather than lack of peptide delivery and processing. However, future studies are needed to validate that all epitopes were indeed displayed on antigen-presenting cells (APCs) by HLA-A*02:01 to confirm this hypothesis and fully validate the construct design. Finally, the total delivery capacity of this platform and capacity of the immune system to respond to a diverse range of vaccine-delivered epitopes is presently unknown. Future experiments can adopt known control epitopes to determine the total number of immunogenic epitopes that can be incorporated into this delivery platform to inform future vaccine design efforts.

Appendix A

Additional Sufficient Input Subsets Experiments

This appendix contains additional details and results for the material in Chapter 2. Further experiments and results can be found in the Supplementary Information of Carter et al. (2019).

A.1 Additional Details of Sentiment Analysis Experiments

Here, we provide additional details of our experiments applying SIS to interpret LSTM models predicting sentiment in beer reviews (Section 2.4).

A.1.1 Imputation Strategies: Mean vs. Hot-deck

In Section 2.2, we discuss the problem of masking input features. Here, we show that the mean-imputation approach (in which missing inputs are masked with a mean embedding, taken over the entire vocabulary) produces a nearly identical change in prediction to a nondeterministic hot-deck approach (in which missing inputs are replaced by randomly sampling feature-values from the data). Figure A-1 shows the change in prediction $f(\mathbf{x} \setminus \{i\}) - f(\mathbf{x})$ by both imputation techniques after drawing

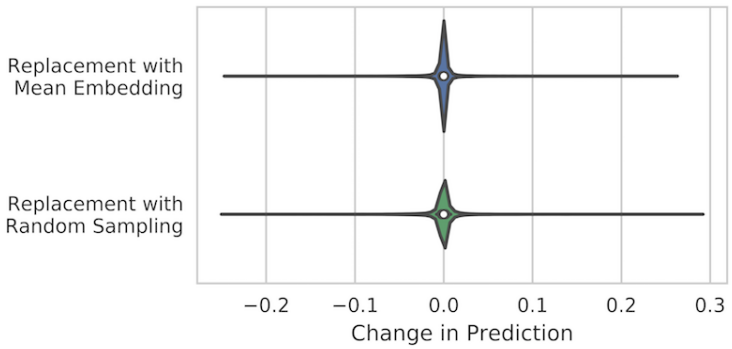


Figure A-1: Change in prediction ($f(\mathbf{x} \setminus \{i\}) - f(\mathbf{x})$) after masking a randomly chosen word with mean imputation or hot-deck imputation. 10,000 replacements were sampled from the aroma beer reviews training set.

a training example $\mathbf{x}$ and word $x_i \in \mathbf{x}$ (both uniformly at random) and replacing x_i with either the mean embedding or a randomly selected word (drawn from the vocabulary, based on counts in the training corpus). This procedure is repeated 10,000 times. Both resulting distributions have mean near zero ($\mu_{\text{mean-embedding}} = -7.0\text{e}{-4}$, $\mu_{\text{hot-deck}} = -7.4\text{e}{-4}$), and the distribution for mean embedding is slightly narrower ($\sigma_{\text{mean-embedding}} = 0.013$, $\sigma_{\text{hot-deck}} = 0.018$). Because we find that these two imputation approaches perform equally well on average, we adopt mean-imputation as our preferred method for masking information about features' values when applying SIS.

We also explored other options for masking word information, e.g., replacement with a zero embedding, replacement with the learned <PAD> embedding, and simply removing the word entirely from the input sequence, but each of these alternative options led to undesirably larger changes in predicted values as a result of masking, indicating they appear more informative to f than replacement via the feature-mean.

A.1.2 Additional Results for Aroma Aspect

This section includes additional results applying our SIS clustering methodology (Section 2.7) to interpret LSTM sentiment predictors. Here, we present the full SIS clustering for both reviews with strong positive and strong negative predicted sentiment (Section 2.7.1).

Table A.1: All clusters of sufficient input subsets extracted from reviews from the test set predicted to have positive aroma by the LSTM. Frequency indicates the number of occurrences of the SIS in the cluster.

Cluster	SIS #1	Freq.	SIS #2	Freq.	SIS #3	Freq.	SIS #4	Freq.
C_1	smell amazing wonderful	2	nice wonderful nose	2	wonderful amazing	2	amazing amazing	2
C_2	grapefruit mango pineapple	2	pineapple grapefruit pineapple grapefruit	1	hops grapefruit pineapple floyds	1	mango pineapple incredible	1
C_3	nice smell citrus nice grapefruit taste	1	smell great complex ripe taste	1	nice smell nice hop smell pine taste	1	love nice nice smell bliss taste	1
C_4	fresh great fantastic taste	1	rich great fantastic hoped	1	fantastic cherries fantastic	1	everyone great snifters fantastic	1
C_5	awesome bounds	1	awesome grapefruit awesome	1	awesome awesome pleasing	1	awesome nailed nailed	1
C_6	creme brulee brulee	3	creme brulee decadent	1	incredible creme brulee	1	creme brulee exceptional	1
C_7	oak vanilla chocolate cinnamon vanilla oak love	1	dose oak chocolate vanilla acidic	1	vanilla figs oak thinner great	1	chocolate aroma oak vanilla dessert	1

Table A.2: All clusters of sufficient input subsets extracted from reviews from the test set predicted to have negative aroma by the LSTM. Frequency indicates the number of occurrences of the SIS in the cluster. Dashes are used in clusters with under 4 unique SIS.

Cluster	SIS #1	Freq.	SIS #2	Freq.	SIS #3	Freq.	SIS #4	Freq.
C_1	awful	15	skunky skunky	9	skunky t	7	skunky taste	6
C_2	garbage	3	taste garbage	1	garbage avoid	1	garbage rice	1
C_3	vomit	16	-	-	-	-	-	-
C_4	gross rotten	1	rotten forte	1	awkward rotten	1	rotten offputting	1
C_5	rancid horrid	1	rancid t	1	rancid	1	rancid avoid	1
C_6	rice t rice	2	rice rice	1	rice tasteless	1	budweiser rice	1

A.1.3 Understanding Differences Between Sentiment Predictors

We also include the full joint SIS clustering (clustering SIS from LSTM and text CNN models together) for reviews with strong positive and strong negative predicted sentiment (Section 2.8.1).

Table A.3: Joint clustering of the SIS extracted from beer reviews predicted to have positive aroma by LSTM or CNN model. Frequency indicates the number of occurrences of the SIS in the cluster. Percentages quantify SIS per cluster from the LSTM. Dashes are used in clusters with under 4 unique SIS.

Cluster	SIS #1	Freq.	SIS #2	Freq.	SIS #3	Freq.	SIS #4	Freq.
C_1 (LSTM: 20%)	rich chocolate	13	very rich	9	chocolate complex	5	smells rich	4
C_2 (LSTM: 21%)	great	248	amazing	119	wonderful	112	fantastic	75
C_3 (LSTM: 47%)	best smelling	23	pineapple mango	6	mango pineapple	6	pineapple grapefruit	5
C_4 (LSTM: 5%)	excellent	42	excellent flemish flemish	1	excellent excellent phenomenal	1	-	-
C_5 (LSTM: 33%)	oak chocolate	2	chocolate raisins raisins oak bourbon	1	chocolate oak	1	raisins chocolate	1
C_6 (LSTM: 5%)	goodness	19	watering goodness	1	-	-	-	-
C_7 (LSTM: 24%)	pumpkin pie	25	huge pumpkin aroma pumpkin pie	1	aroma perfect pumpkin pie taste	1	smell pumpkin nutmeg cinnamon pie	1
C_8 (LSTM: 5%)	jd	13	tremendous	8	tremendous jd	1	-	-
C_9 (LSTM: 40%)	brulee	14	creme brulee brulee	3	creme creme	1	creme brulee amazing	1
C_{10} (LSTM: 0%)	s wow	20	-	-	-	-	-	-
C_{11} (LSTM: 0%)	delicious	56	-	-	-	-	-	-
C_{12} (LSTM: 0%)	very nice	23	-	-	-	-	-	-
C_{13} (LSTM: 70%)	complex aroma	5	aroma complex peaches complex	1	aroma complex interesting cherries	1	aroma complex	1

Table A.4: Joint clustering of the SIS extracted from beer reviews predicted to have negative aroma by LSTM or CNN model. Frequency indicates the number of occurrences of the SIS in the cluster. Percentages quantify SIS per cluster from the LSTM. Dashes are used in clusters with under 4 unique SIS.

Cluster	SIS #1	Freq.	SIS #2	Freq.	SIS #3	Freq.	SIS #4	Freq.
C_1 (LSTM: 29%)	not	247	no	105	bad	104	macro	94
C_2 (LSTM: 100%)	gross rotten	1	-	-	-	-	-	-
C_3 (LSTM: 100%)	rotten garbage	1	-	-	-	-	-	-
C_4 (LSTM: 62%)	vomit	26	-	-	-	-	-	-
C_5 (LSTM: 21%)	budweiser	22	sewage budweiser	1	metal budweiser	1	budweiser budweiser budweiser	1
C_6 (LSTM: 100%)	garbage rice	1	-	-	-	-	-	-
C_7 (LSTM: 3%)	n't	19	adjuncts	14	n't adjuncts	1	-	-
C_8 (LSTM: 0%)	faint	82	-	-	-	-	-	-
C_9 (LSTM: 0%)	adjunct	42	-	-	-	-	-	-

Appendix B

Additional Overinterpretation Experiments

This appendix contains additional details and results for the material in Chapter 3.

B.1 Details of Batched Gradient SIS Algorithm

It is computationally infeasible to scale the original backward selection procedure of SIS (Section 2.2) to ImageNet. As each ImageNet image contains $299 \times 299 = 89401$ pixels, running backward selection to find one SIS for an image would require ~ 4 billion forward passes through the network. Here we introduce a more efficient gradient-based approximation to the original SIS procedure (via **Batched Gradient SIScollection**, **Batched Gradient BackSelect**, and **Batched Gradient FindSIS**) that allows us to find SIS on larger ImageNet images in a reasonable time. The **Batched Gradient SIScollection** procedure described below identifies a complete collection of disjoint masks for an input $\mathbf{x}$, where each mask M specifies a pixel-subset of the input $\mathbf{x}_S = \mathbf{x} \odot (1 - M)$ such that $f(\mathbf{x}_S \geq \tau)$. Here f outputs the probability assigned by the network to its predicted class (i.e., its confidence).

The idea behind our approximation algorithm is two-fold: (1) Instead of separately masking every remaining pixel to find the least critical pixel (whose masking least reduces the confidence in the network's prediction), we use the *gradient* with respect

to the mask as a means of ordering. (2) Instead of masking just 1 pixel per iteration, we mask larger subsets of $k \geq 1$ pixels per iteration. More formally, let $\mathbf{x}$ be an image of dimensions $H \times W \times C$ where H is the height, W the width, and C the channel. Let $f(\mathbf{x})$ be the network's confidence on image $\mathbf{x}$ and τ the target SIS confidence threshold. Recall that we only compute SIS for images where $f(\mathbf{x}) \geq \tau$. Let M be the mask with dimensions $H \times W$ with 0 indicating an unmasked feature (pixel) and 1 indicating a masked feature. We initialize M as all 0s (all features unmasked). At iteration i, we compute the gradient of f with respect to the input pixels and mask $\nabla M = \nabla_M f(\mathbf{x} \odot (1 - M))$. Here M is the current mask updated after each iteration. In each iteration, we find the block of k features to mask, G^*, chosen in descending order by value of entries in ∇M. The mask is updated after each iteration by masking this block of k features until all features have been masked. Given p input features, our **Batched Gradient SIScollection** procedure returns j sufficient input subsets in $\mathcal{O}(\frac{p}{k} \cdot j)$ evaluations of ∇f (as opposed to $\mathcal{O}(p^2 j)$ evaluations of f in the original SIS procedure (Section 2.2).

Here, we use $k = 100$, which allows us to find one SIS for each of 32 ImageNet images (i.e., a mini-batch) in ∼1-2 minutes using **Batched Gradient FindSIS**. Note that while our algorithm is an approximate procedure, the pixel-subsets produced are real sufficient input subsets, i.e., they always satisfy $f(\mathbf{x}_S \geq \tau)$. For CIFAR-10 images (which are smaller in size), we use the original SIS procedure (Section 2.2). For both datasets, we treat all channels of each pixel as a single feature.

Algorithm 4: Batched Gradient SIScollection

Input: function f, input $\mathbf{x}$, threshold τ, batch size k (number of pixels)

$M = \mathbf{0}$

for $j = 1, 2, \ldots$ **do**

  $R =$ **Batched Gradient BackSelect**$(f, \mathbf{x}, M, k)$

  $M_j =$ **Batched Gradient FindSIS**$(f, \mathbf{x}, \tau, R)$

  $M \leftarrow M + M_j$

  if $f(\mathbf{x} \odot (1 - M)) < \tau$ **then**

    return $M_1, \ldots, M_{j-1}$

  end if

end for

Algorithm 5: Batched Gradient BackSelect

Input: function f, input $\mathbf{x}$, mask M, batch size k (number of pixels)

$R =$ empty stack

while $M \neq \mathbf{1}$ **do**

  $G^* = \mathrm{Top}_k\left(\nabla_M f(\mathbf{x} \odot (1 - M)\right)$

  Update $M \leftarrow M + G^*$

  Push G^* onto top of R

end while

return R

Algorithm 6: Batched Gradient FindSIS

Input: function f, input $\mathbf{x}$, threshold τ, stack R

$M = \mathbf{1}$

while $f(\mathbf{x} \odot (1 - M)) < \tau$ **do**

  Pop G from top of R

  Update $M \leftarrow M - G$

end while

if $f(\mathbf{x} \odot (1 - M)) \geq \tau$ **then**

  return M

else

  return *None*

end if

B.2 Model Implementation and Training Details

CIFAR-10 Models

We first describe the implementation and training details for the CIFAR-10 models used in this work (Section 3.2.1). The ResNet20 architecture (He et al., 2016a) has 16 initial filters and a total of 0.27M parameters. ResNet18 (He et al., 2016b) has 64 initial filters and contains 11.2M parameters. The VGG16 architecture (Simonyan and Zisserman, 2015) uses batch normalization and contains 14.7M parameters.

All models are trained for 200 epochs with a batch size of 128. We minimize cross-entropy via SGD with Nesterov momentum (Sutskever et al., 2013) using momentum of 0.9 and weight decay of 5e-4. The learning rate is initialized as 0.1 and is reduced by a factor of 5 after epochs 60, 120, and 160. Datasets are normalized using per-channel mean and standard deviation, and we use standard data augmentation strategies consisting of random crops and horizontal flips (He et al., 2016b).

The adversarially robust model we evaluated is the `adv_trained` model of Madry et al. (2018), available on GitHub[1].

To apply the SIS procedure to CIFAR-10 images, we use an implementation available on GitHub[2]. For confidently classified images on which we run SIS, we find one sufficient input subset per image using the FindSIS procedure. When masking pixels, we mask all channels of each pixel as a single feature.

ImageNet Models

For finding SIS, we use pre-trained models (Inception v3 (Szegedy et al., 2016) and ResNet50 (He et al., 2016a)) provided by PyTorch (Paszke et al., 2019) in the torchvision package (PyTorch version 1.4.0, torchvision version 0.5.0).

When training new ImageNet classifiers, we adopt model implementations and training scripts from PyTorch (Paszke et al., 2019), obtained from GitHub[3]. Models

[1] `https://github.com/MadryLab/cifar10_challenge`

[2] `https://github.com/google-research/google-research/blob/master/sufficient_input_subsets/sis.py`

[3] `https://github.com/pytorch/examples/blob/master/imagenet/main.py`

are trained for 90 epochs using batch size 256 (Inception-v3) or 512 (ResNet50). We minimize cross-entropy via SGD using momentum of 0.9 and weight decay of 1e-4. The learning rate is initialized as 0.1 and reduced by a factor of 10 every 30 epochs. Datasets are normalized using per-channel mean and standard deviation. For Inception v3, images are cropped to 299 x 299 pixels. For ResNet50, images are cropped to 224 x 224. When training Inception v3, we define the model using the `aux_logits=False` argument. We do not use data augmentation when training models on pixel-subsets of images.

Hardware Details

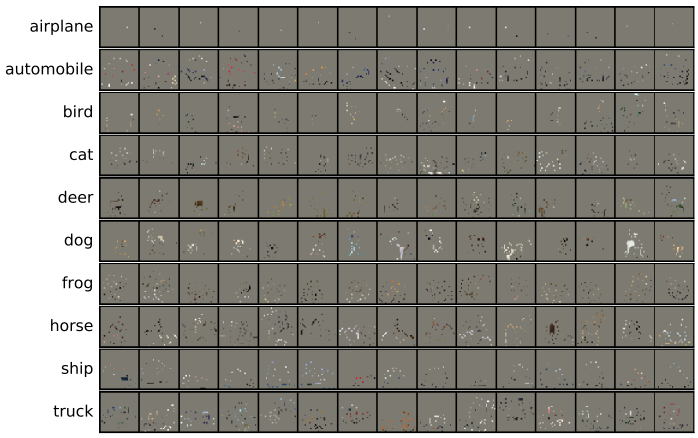
Each CIFAR-10 model is trained on 1 NVIDIA GeForce RTX 2080 Ti GPU. Once models are trained, SIS are computed across multiple GPUs (by parallelizing over individual images). Each SIS (for 1 CIFAR-10 image) takes roughly 30-60 seconds to compute (depending on the model architecture).

ImageNet models are trained on 2–3 NVIDIA Titan RTX GPUs. For finding SIS from pre-trained ImageNet models, we run Batched Gradient BackSelect for batches of 32 images across 10 NVIDIA GeForce RTX 2080 Ti GPUs, which takes roughly 1-2 minutes per batch (details in Appendix B.1).

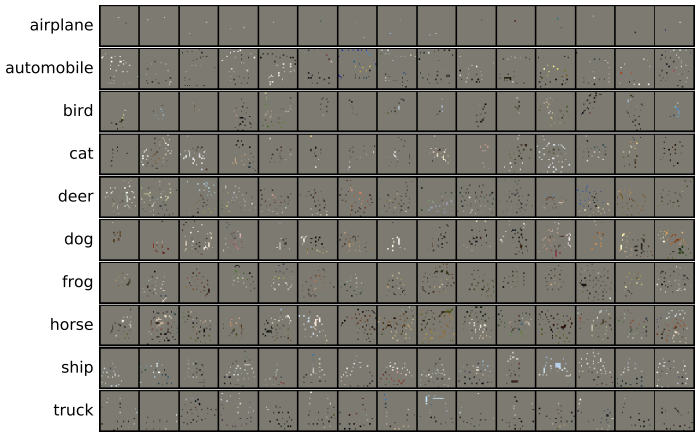
B.3 Additional Examples of CIFAR-10 Sufficient Input Subsets

B.3.1 SIS of Individual Networks

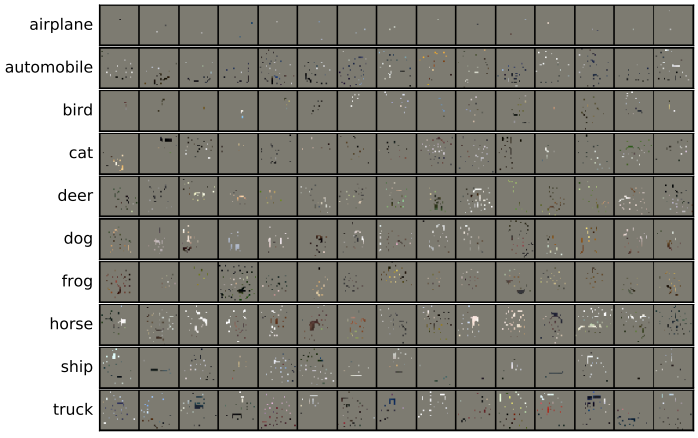
Figure B-1 shows a sample of SIS for each of our three architectures. These images were randomly sampled among all CIFAR-10 test images confidently (confidence $\geq$ 0.99) predicted to belong to the class written on the left. Out of 10000 CIFAR-10 test images, 8596 were predicted with $\geq 99\%$ confidence by ResNet18 (7829 by ResNet20, 9048 by VGG16). SIS are computed under a threshold of 0.99, so all images shown in this figure are classified with probability $\geq 99\%$ confidence as belonging to the listed class.



(a) ResNet20



(b) ResNet18



(c) VGG16

Figure B-1: Examples of SIS (threshold 0.99) on random sample of CIFAR-10 test images (15 per class, different random sample for each architecture). All images shown here are predicted to belong to the listed class with $\geq 99\%$ confidence.

B.3.2 Ensemble Sufficient Input Subsets

Figure B-2 shows examples of SIS from one of our model ensembles (a homogeneous ensemble of ResNet18 networks, see Section 3.2.1), along with corresponding SIS for the same image from each of the five member networks in the ensemble. We use a SIS threshold of 0.99, so all images are classified with $\geq 99\%$ confidence. These examples highlight how the ensemble SIS are larger and draw class-evidence from the individual members' SIS.

B.4 Additional Results on CIFAR-10

B.4.1 Training on Pixel-Subsets With Data Augmentation

Table B.1 presents results similar to those in Section 3.3.2 and Table 3.1, but where models are trained on 5% pixel-subsets with data augmentation (as described in Appendix B.2). We find training without data augmentation slightly improves accuracy when training classifiers on 5% pixel-subsets of CIFAR-10.

Table B.1: Accuracy of CIFAR-10 classifiers trained and evaluated on full images, 5% backward selection (BS) pixel-subsets, and 5% random pixel-subsets *with* data augmentation (+). Accuracy is reported as mean ± standard deviation (%) over five runs.

Model	Train On	Evaluate On	CIFAR-10 Test Acc.	CIFAR-10-C Acc.
ResNet20	5% BS Subsets (+)	5% BS Subsets	92.26 ± 0.01	70.21 ± 0.14
	5% Random (+)	5% Random	48.87 ± 0.41	42.66 ± 0.15
ResNet18	5% BS Subsets (+)	5% BS Subsets	94.51 ± 0.38	74.91 ± 0.41
	5% Random (+)	5% Random	49.03 ± 0.92	42.97 ± 0.82
VGG16	5% BS Subsets (+)	5% BS Subsets	91.17 ± 0.04	71.82 ± 0.13
	5% Random (+)	5% Random	51.32 ± 1.35	44.56 ± 0.96

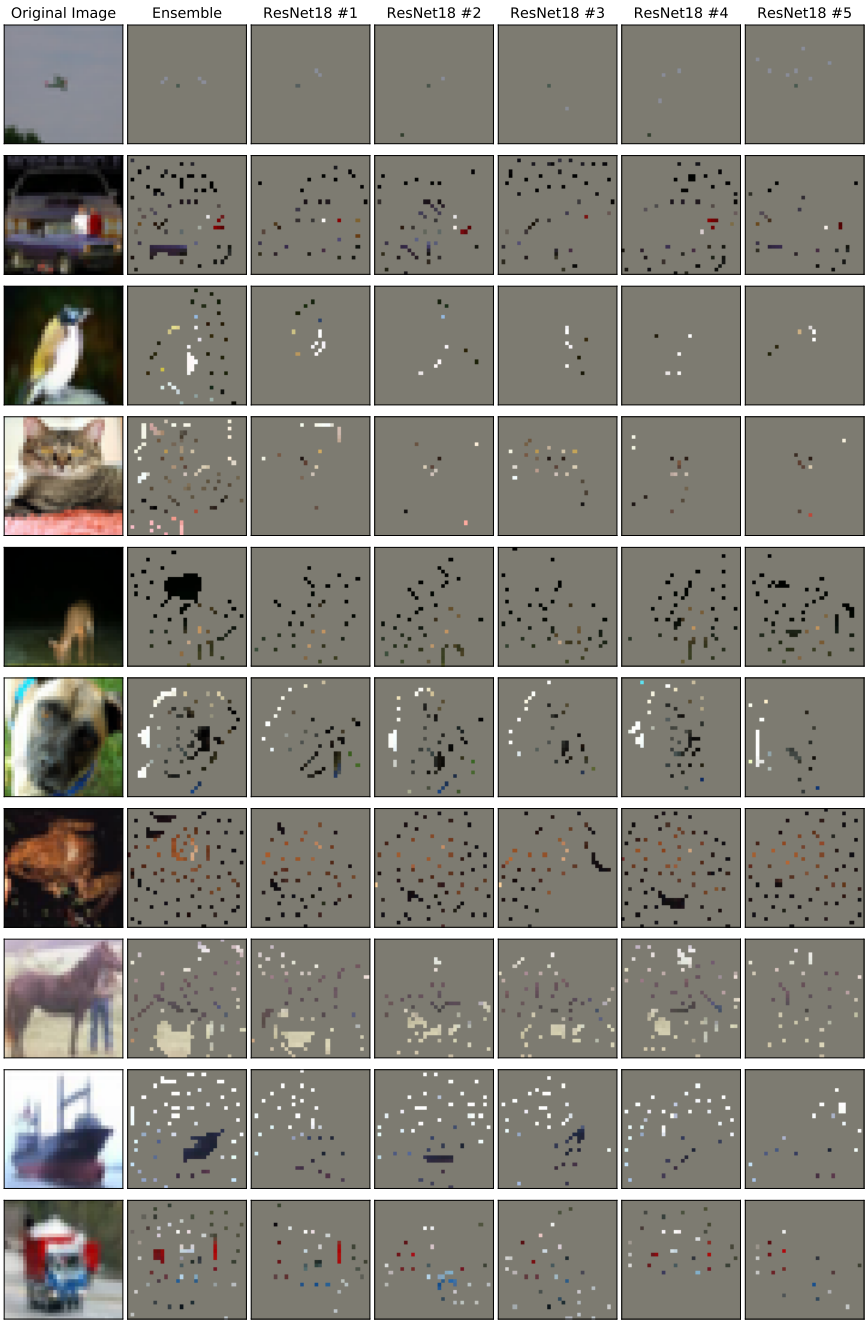


Figure B-2: Examples of SIS (threshold 0.99) from the ResNet18 homogeneous ensemble (Section 3.2.1) and its member models. Each row shows original CIFAR-10 image (left), followed by SIS from the ensemble (second column) and the SIS from each of its 5 member networks (remaining columns). Each image shown is classified with $\geq 99\%$ confidence by its respective network.

B.4.2 Training on Pixel-Subsets With Different Architectures

Table B.2 presents results of training and evaluating models on 5% pixel-subsets drawn from different architectures. Models were trained without data augmentation on subsets from one replicate of each base architecture. We find accuracy from training and evaluating a model on 5% pixel-subsets of images derived from a different architecture is commensurate with accuracy of training and evaluating a new model of the same type on those subsets (Table 3.1).

Table B.2: Accuracy of CIFAR-10 classifiers trained and evaluated on 5% backward selection (BS) pixel-subsets from different architectures. Accuracy is reported as mean ± standard deviation (%) over five runs.

5% Subsets from Model	Model Trained	CIFAR-10 Test Acc.	CIFAR-10-C Acc.
ResNet20	ResNet18	92.53 ± 0.02	70.56 ± 0.04
	VGG16	92.47 ± 0.02	70.42 ± 0.14
ResNet18	ResNet20	94.88 ± 0.03	75.14 ± 0.10
	VGG16	94.88 ± 0.05	75.13 ± 0.09
VGG16	ResNet20	92.05 ± 0.14	73.01 ± 0.08
	ResNet18	92.57 ± 0.10	73.33 ± 0.21

B.4.3 Additional Results for Models Trained on Pixel-Subsets

Table B.3 presents results of models trained on 5% backward selection (BS) or random pixel-subsets of CIFAR-10 training images, evaluated on full (original) CIFAR-10 test images. While accuracies are generally significantly higher than random guessing, we note that full images are highly out-of-distribution for a model trained on images with only 5% pixel-subsets and hence such a model cannot properly generalize to full images. Further, the model trained on 5% images may not rely on the same features as the model trained on full images as it is trained on a substantially different training set.

Table B.3: Accuracy of CIFAR-10 classifiers trained on 5% backward selection (BS) or random pixel-subsets with (+) and without (−) data augmentation. Accuracy is reported as mean ± standard deviation (%) over five runs.

Model	Train On	Evaluate On	CIFAR-10 Test Acc.	CIFAR-10-C Acc.
ResNet20	5% BS Subsets (−)	Full Images	21.02 ± 1.57	17.50 ± 1.15
	5% Random (−)	Full Images	38.66 ± 3.31	36.40 ± 2.73
	5% BS Subsets (+)	Full Images	10.87 ± 1.50	10.75 ± 1.32
	5% Random (+)	Full Images	37.08 ± 3.51	33.78 ± 2.81
ResNet18	5% BS Subsets (−)	Full Images	20.86 ± 2.74	18.20 ± 1.43
	5% Random (−)	Full Images	26.05 ± 7.59	25.03 ± 6.41
	5% BS Subsets (+)	Full Images	11.83 ± 1.74	11.48 ± 1.15
	5% Random (+)	Full Images	20.98 ± 4.61	20.35 ± 3.56
VGG16	5% BS Subsets (−)	Full Images	41.63 ± 3.55	30.34 ± 1.97
	5% Random (−)	Full Images	25.73 ± 6.08	23.56 ± 4.39
	5% BS Subsets (+)	Full Images	14.32 ± 3.40	13.22 ± 2.01
	5% Random (+)	Full Images	27.58 ± 3.96	24.92 ± 3.10

B.4.4 Additional Results for SIS Size and Model Accuracy

Figure B-3 shows percentage increase in mean SIS size for correctly classified images compared to misclassified images from the CIFAR-10-C dataset.

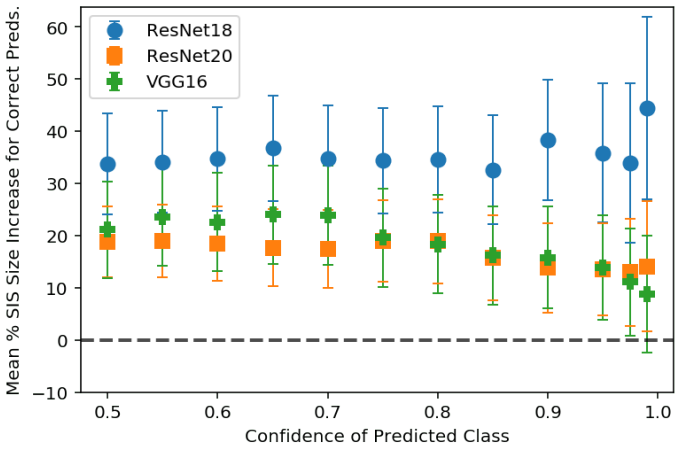
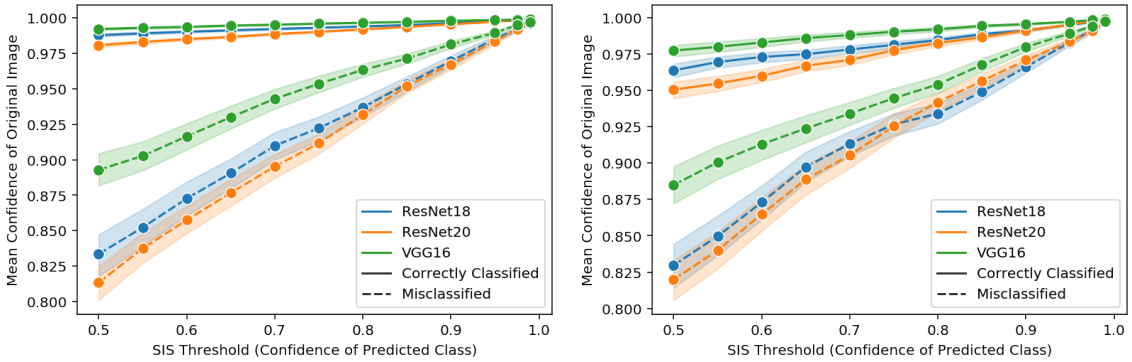


Figure B-3: Percentage increase in mean SIS size of correctly classified images compared to misclassified images from a random sample of CIFAR-10-C test set. Positive values indicate larger mean SIS size for correctly classified images. Error bars indicate 95% confidence interval for the difference in means.

Figure B-4 shows the mean confidence of each group of correctly and incorrectly classified images that we consider at each confidence threshold (at each confidence threshold along the x-axis, we evaluate SIS size in Figure 3-5 on the set of images that originally were classified with at least that level of confidence). We find model confidence is uniformly lower on the misclassified inputs.



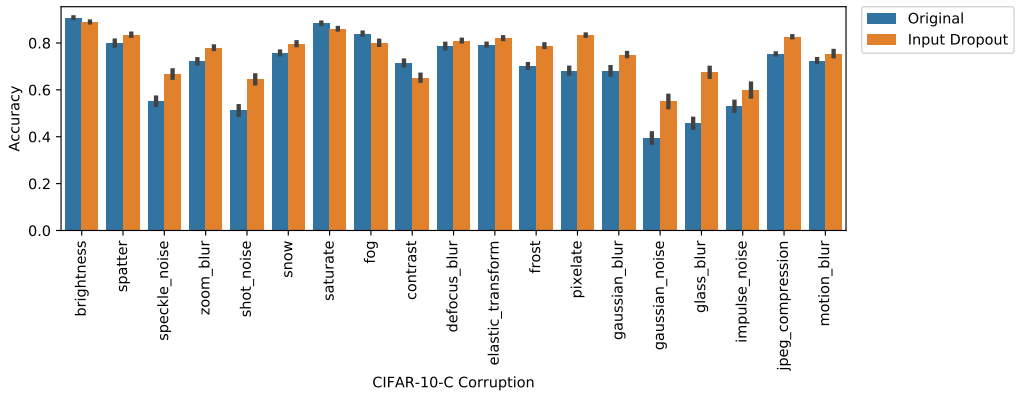
(a) CIFAR-10 test set

(b) CIFAR-10-C test set

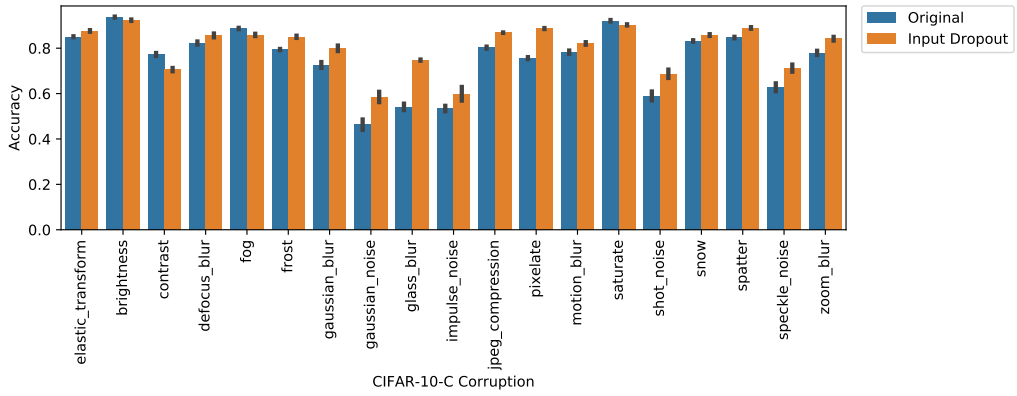
Figure B-4: Mean confidence of correctly vs. incorrectly classified images for each corresponding SIS threshold we evaluate in Figure 3-5 across the (a) CIFAR-10 test set and (b) our random sample of the CIFAR-10-C test set. Shaded region indicates 95% confidence interval.

B.4.5 Additional Results for Input Dropout

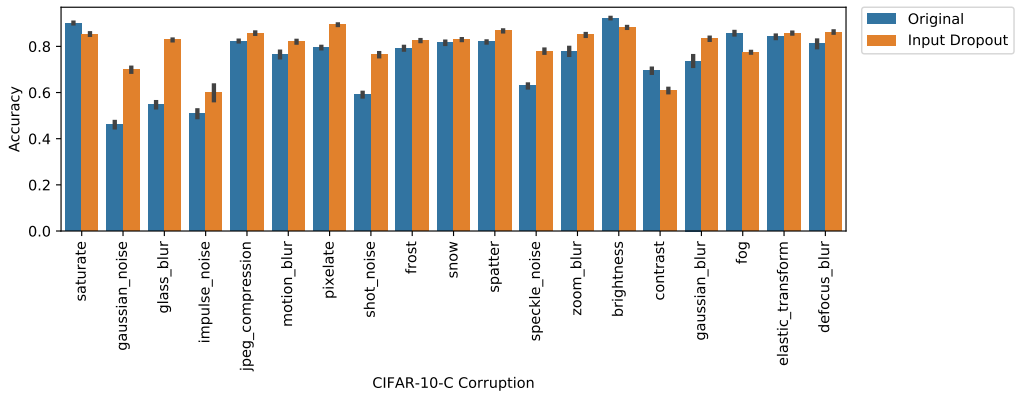
Figure B-5 shows the accuracy improvement on each individual corruption of the CIFAR-10-C out-of-distribution test set for models trained with input dropout (Section 3.3.5) compared to original models.



(a) ResNet20



(b) ResNet18



(c) VGG16

Figure B-5: Accuracy on individual corruptions of CIFAR-10-C out-of-distribution images for original models and models trained with input dropout (Section 3.3.5). Accuracy is given as mean ± standard deviation over five replicate models.

B.4.6 Results on CIFAR-10.1

Table B.4 reports accuracy of the models from Section 3.3.2 computed on the CIFAR-10.1 v6 dataset (Recht et al., 2018), which contains 2000 class-balanced images drawn from the Tiny Images repository (Torralba et al., 2008) in a similar fashion to that of CIFAR-10, though Recht et al. (2018) found a large drop in classification accuracy on these images.

B.4.7 SIS and Calibrated Models

We calibrated one model of each architecture class after training using Temperature Scaling (Guo et al., 2017) based on an implementation available on GitHub[4]. The CIFAR-10 test set was randomly split into a 5k validation set (for optimization of the temperature parameter) and a 5k held-out test set (for final evaluation of ECE). Table B.5 shows the Expected Calibration Error (ECE) of each model on held-out test images before and after calibration, as well as mean SIS size using confidence threshold 0.99 computed on the entire CIFAR-10 test set. We find that while the mean SIS size (for test images that the re-calibrated model can classify with $\geq 99\%$ confidence) does increase slightly, the resulting SIS subsets are still semantically meaningless and far below the threshold of SIS size where humans can meaningfully start to classify CIFAR images with any degree of accuracy (Figure B-6). We note that one of the key findings of our work is that even when we compute SIS subsets from uncalibrated models, those subsets still contain enough signal for training entirely new classifiers that can generalize as well to the corresponding test subsets (Section 3.3.2).

B.4.8 SIS with Random Tie-breaking

We suspect the concentration of pixels on the bottom border for ResNet20 (Figure 3-3a) is a result of tie-breaking during backward selection of the SIS procedure. To explore this hypothesis, we modified the tie-breaking procedure to randomly (rather than deterministically) break ties during SIS backward selection by adding random

[4] `https://github.com/gpleiss/temperature_scaling`

Table B.4: Accuracy of CIFAR-10 classifiers trained and evaluated on full images, 5% backward selection (BS) pixel-subsets, and 5% random pixel-subsets reported on CIFAR-10.1 v6 dataset (evaluating models from Section 3.3.2 that were trained on full images or 5% subsets of the CIFAR-10 train set). Where possible, accuracy is reported as mean ± standard deviation (%) over five runs. For training on BS subsets, we run BS on all images for a single model of each type and average over five models trained on these subsets.

Model	Train On	Evaluate On	CIFAR-10.1 Acc.
ResNet20	Full Images	Full Images	83.98 ± 0.68
		5% BS Subsets	82.80
		5% Random	10.00 ± 0.00
	5% BS Subsets	5% BS Subsets	82.56 ± 0.07
	5% Random	5% Random	39.78 ± 1.27
	Input Dropout (Full)	Input Dropout (Full)	81.88 ± 0.44
ResNet18	Full Images	Full Images	88.89 ± 0.45
		5% BS Subsets	89.35
		5% Random	10.06 ± 0.11
	5% BS Subsets	5% BS Subsets	89.49 ± 0.04
	5% Random	5% Random	39.45 ± 1.02
	Input Dropout (Full)	Input Dropout (Full)	86.28 ± 0.33
VGG16	Full Images	Full Images	86.23 ± 0.79
		5% BS Subsets	86.45
		5% Random	9.78 ± 0.26
	5% BS Subsets	5% BS Subsets	85.61 ± 0.19
	5% Random	5% Random	40.98 ± 1.27
	Input Dropout (Full)	Input Dropout (Full)	81.00 ± 0.65
Ensemble (ResNet18)	Full Images	Full Images	90.30
		5% Random	10.05

Table B.5: Results of model calibration by temperature scaling. Expected Calibration Error (ECE) is computed on a held-out set of 5k CIFAR-10 test images. SIS are computed using a threshold of 0.99 on all CIFAR-10 test images classified with $\geq$ 99% confidence (and corresponding number of such images listed). SIS size is given as mean $\pm$ standard deviation.

Model	ECE (%)	SIS Size (% of Image)	Num. Images Pred. ≥ 0.99
ResNet20 Uncalibrated	3.91	2.36 ± 1.21	7829
ResNet20 Calibrated	0.91	2.94 ± 1.39	5805
ResNet18 Uncalibrated	2.49	2.53 ± 1.53	8596
ResNet18 Calibrated	1.00	3.54 ± 1.94	5934
VGG16 Uncalibrated	4.95	2.18 ± 1.37	9048
VGG16 Calibrated	1.56	8.26 ± 2.86	23

Gaussian noise ($\mu = 0$, $\sigma^2 = 1e{-}12$) to the model's outputs for each remaining masked pixel at each iteration of backward selection. For each image in a sample of 1000 CIFAR-10 test images, we repeated this randomization procedure three times and found the resulting heatmap of 5% backward selection pixel-subsets for ResNet20 more concentrated in the image centers rather than bottom border (Figure B-7).

B.4.9 Confidence Curves for SIS Backward Selection on CIFAR-10

Figure B-8 shows the predicted confidence on the remaining pixels at each step of SIS backward selection for the entire CIFAR-10 test set for each architecture trained on CIFAR-10.

(a) ResNet20 Calibrated

(b) ResNet18 Calibrated

(c) VGG16 Calibrated

Figure B-6: Examples of SIS (threshold 0.99) on sample of CIFAR-10 test images from calibrated models. All images shown are predicted to belong to the listed class with $\geq 99\%$ confidence.

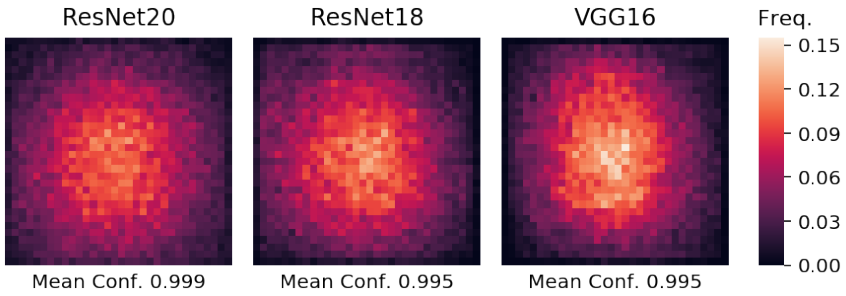
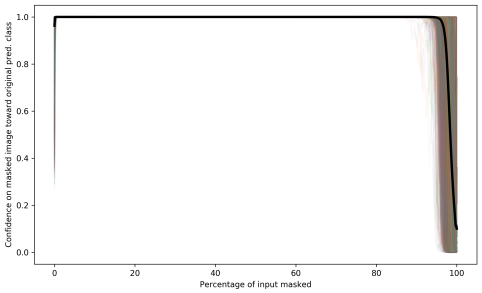
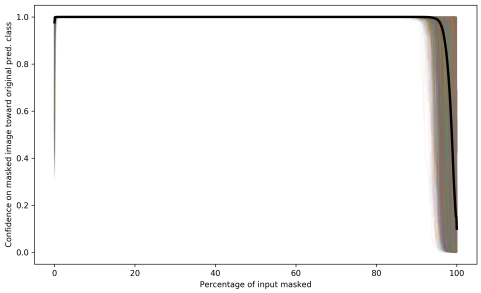


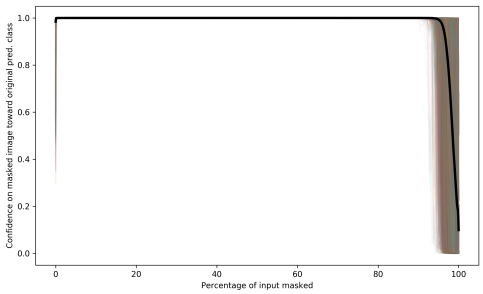
Figure B-7: Heatmap of pixel locations comprising 5% backward selection pixel-subsets computed on a set of 1000 CIFAR-10 test set images with random tie-breaking during backward selection.



(a) ResNet20



(b) ResNet18

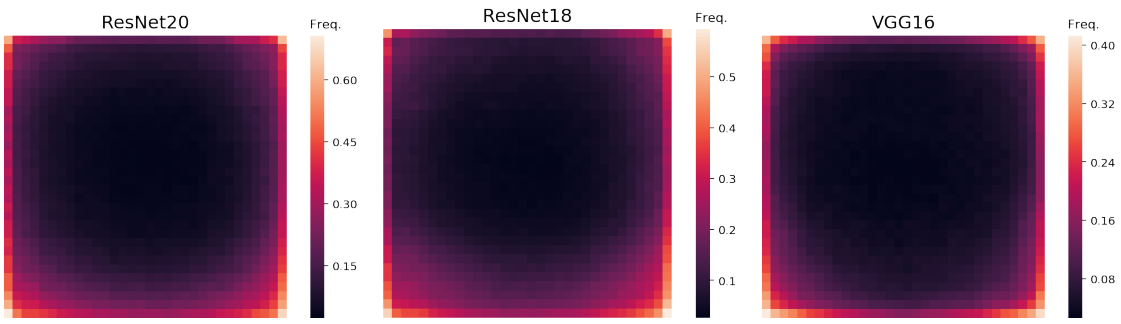


(c) VGG16

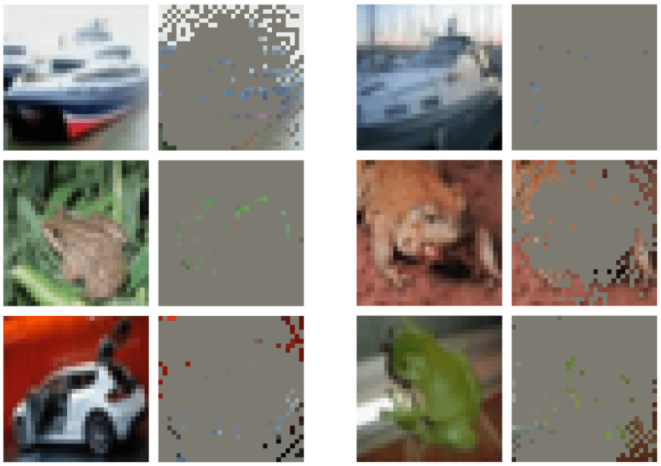
Figure B-8: Prediction history on remaining (unmasked) pixels at each step of the SIS backward selection procedure for all CIFAR-10 test set images. Black line depicts mean confidence at each step.

B.4.10 Batched Gradient SIS on CIFAR-10

We also ran Batched Gradient SIS on the entire CIFAR-10 test set for ResNet18 and found Batched Gradient SIS produced edge-heavy heatmaps for CIFAR-10 (Figure B-9a). For CIFAR-10, we set $k = 1$ to remove a single pixel per iteration of Batched Gradient SIS. These heatmap differences (compared to Figure 3-3) are a result of the different valid equivalent SIS subsets found by the two SIS discovery algorithms. However, since all SIS subsets are validated with a model and guaranteed to be sufficient for classification at the specified threshold, the heatmaps are accurate depictions of what is sufficient for the model to classify images at the threshold. Overinterpretation is independent of the SIS algorithm used because both algorithms produce human-uninterpretable sufficient subsets (Figure B-9b).



(a)



(b)

Figure B-9: Results of running Batched Gradient SIS (threshold 0.99) on CIFAR-10. (a) Heatmaps of SIS pixel locations computed on entire CIFAR-10 test set for each architecture. (b) Example Batched Gradient SIS for ResNet18 (all images and SIS subsets shown are classified with $\geq 99\%$ confidence).

B.5 Details of Human Classification Benchmark

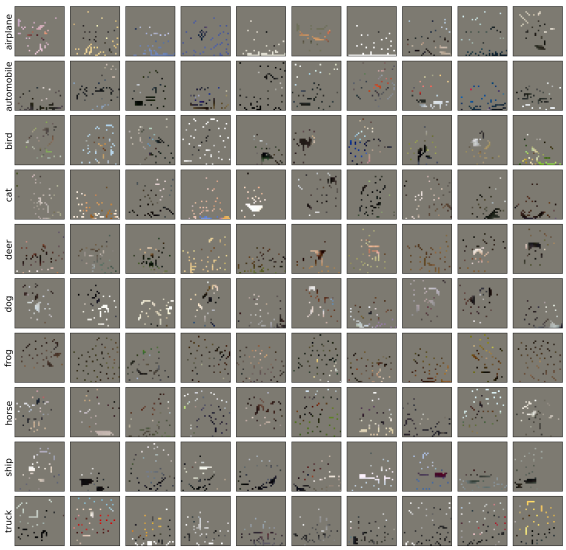
Here we include additional details on our benchmark of human classification accuracy of sparse pixel-subsets (Section 3.2.4). Figure B-10 shows all images shown to users (100 images each for 5%, 30% and 50% pixel-subsets of CIFAR-10 test images). Each set of 100 images has pixel-subsets stemming from each of the three architectures roughly equally (35 ResNet20, 35 ResNet18, 30 VGG16).[5] Figure B-11 shows the correlation between human classification accuracy and pixel-subset size (accuracies shown in Table B.6).

Table B.6: Human classification accuracy on a sample of CIFAR-10 test image pixel-subsets of varying sparsity (see Section 3.2.4). Accuracies given as mean ± standard deviation.

Fraction of Images	Human Classification Accuracy (%)
5%	19.2 ± 4.8
30%	40.0 ± 2.5
50%	68.2 ± 3.6

[5]The human classification benchmark was performed using pixel-subsets computed from earlier implementations of the three CNN architectures (in Keras rather than PyTorch). Figure B-10 shows all pixel-subsets derived from these models that were shown to users in the human classification benchmark. ResNet20 was based on a Keras example using 16 initial filters and optimized with Adam for 200 epochs (batch size 32, initial learning rate 0.001, reduced after epochs 80, 120, 160, and 180 to 1e-4, 1e-5, 1e-6, and 5e-7, respectively). ResNet18 was based on a GitHub implementation using 64 initial filters, initial strides (1, 1), initial kernel size (3, 3), no initial pooling layer, weight decay 0.0005 and trained using SGD with Nesterov momentum 0.9 for 200 epochs (batch size 128, initial learning rate 0.1, reduced by a factor of 5 after epochs 60, 120, and 160). VGG16 was based on a GitHub implementation trained with weight decay 0.0005 and SGD with Nesterov momentum 0.9 for 250 epochs (batch size 128, initial learning rate 0.1, decayed after each epoch as $0.1 \cdot 0.5^{\lfloor \text{epoch}/20 \rfloor}$). We selected the final model checkpoint that maximized test accuracy. We found these models exhibited similar overinterpretation behavior to the final models.

- https://keras.io/examples/cifar10_resnet/
- https://github.com/keras-team/keras-contrib/blob/master/keras_contrib/applications/resnet.py
- https://github.com/geifmany/cifar-vgg/blob/e7d4bd4807d15631177a2fafabb5497d0e4be3ba/cifar10vgg.py



(a) 5% Pixel-Subsets

(b) 30% Pixel-Subsets

(c) 50% Pixel-Subsets

Figure B-10: Pixel-subsets of CIFAR-10 test images shown to participants in our human classification benchmark (Section 3.2.4).

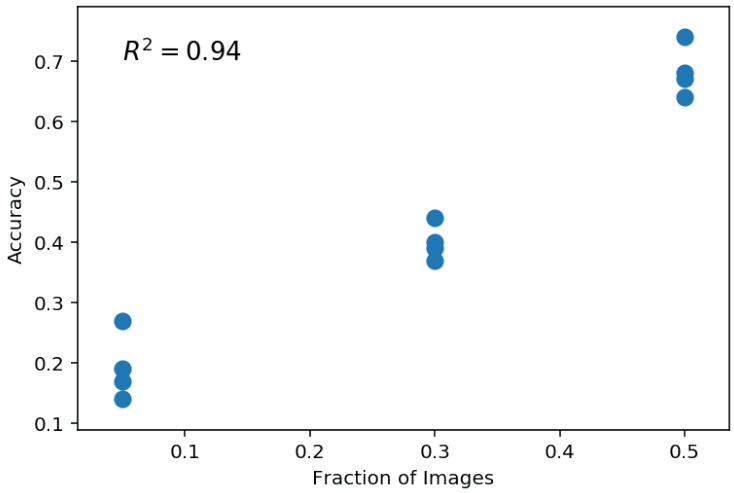


Figure B-11: Human classification accuracy on a sample of CIFAR-10 test image pixel-subsets (see Section 3.2.4).

B.6 Additional Results of ImageNet Overinterpretation

B.6.1 Training CNNs on ImageNet Pixel-Subsets

We extracted 10% backward selection (BS) pixel-subsets by applying Batched Gradient BackSelect to all ImageNet train and validation images using pre-trained Inception v3 and ResNet50 models from PyTorch (Paszke et al., 2019). We kept the top 10% of pixels and masked the remaining 90% with zeros. We trained new models of the same type on these 10% BS pixel-subsets of ImageNet training set images (training details in Appendix B.2) and evaluated the resulting models on the corresponding 10% pixel-subsets of ImageNet validation images. Table B.7 shows a small loss in validation accuracy, suggesting these 10% pixel-subsets that are indiscernible by humans contain statistically valid signals that generalize to validation images. Models trained on 10% pixel-subsets were trained without data augmentation. As with CIFAR-10 (Appendix B.4), we found training models on pixel-subsets with standard data augmentation techniques (random crops and horizontal flips) resulted in worse validation accuracy.

We also trained and evaluated ImageNet models on random pixel-subsets, and results are shown in Table B.7. For training on random pixel-subsets, each of the five training runs was trained on different random pixel-subsets. For evaluation of pre-trained models on random subsets, each pre-trained model was evaluated on five different random random pixel-subsets. All pixels in random pixel-subsets were drawn uniformly at random, and the remaining pixels masked with zeros. We found random 10% pixel-subsets significantly less informative to pre-trained classifiers than 10% backward selection pixel-subsets from Batched Gradient SIS.

We repeated the experiment of Table B.2 and found for ImageNet that 10% pixel-subsets from one architecture can also be used to train a new model of a different architecture. We trained a new DenseNet-121 model (Huang et al., 2017) on 10%

BS pixel-subsets of ImageNet training images drawn from the ResNet50[6], and the DenseNet-121 was able to classify the corresponding 10% BS pixel-subsets of ImageNet validation images as accurately as the ResNet50 trained on the 10% BS pixel-subsets (Table B.7).

B.6.2 Additional Examples of SIS on ImageNet

Figure B-12 shows additional examples of SIS (threshold 0.9) on ImageNet validation images for the pre-trained Inception v3 found via Batched Gradient FindSIS. Figure B-13 shows examples of SIS for the pre-trained ResNet50.

[6] we used subsets drawn from ResNet50 as the default input image size for Inception v3 is 299×299 while the default input image size for ResNet50 and DenseNet-121 is 224×224

Table B.7: Accuracy of models on ImageNet validation images trained and evaluated on full images, backward selection (BS) pixel-subsets, and random pixel-subsets. Accuracy for training on 10% BS Subsets is reported as mean ± standard deviation (%) over five training runs with different random initialization. For training/evaluation on BS pixel-subsets, we run backward selection on all ImageNet images using a single pre-trained model of each type, but average over five models trained on these subsets. For training on random pixel-subsets, each of the five training runs was trained on different random pixel-subsets. For evaluation of pre-trained models on random subsets, each pre-trained model was evaluated on five different random random pixel-subsets. All pixels in random pixel-subsets were drawn uniformly at random.

Model	Train On	Evaluate On	Top 1 Acc.	Top 5 Acc.
Inception v3	Full Images (pre-trained)	Full Images	77.21	93.53
		10% BS Subsets	73.87	83.43
		15% BS Subsets	76.15	84.93
		20% BS Subsets	76.75	85.40
		10% Random	0.75 ± 0.02	2.55 ± 0.03
		15% Random	1.51 ± 0.03	4.61 ± 0.03
		20% Random	2.83 ± 0.03	7.75 ± 0.03
	10% BS Subsets	10% BS Subsets	71.37 ± 0.15	83.73 ± 0.10
	10% Random	10% Random	64.53 ± 0.16	85.36 ± 0.10
ResNet50	Full Images (pre-trained)	Full Images	76.13	92.86
		10% BS Subsets	45.14	64.12
		15% BS Subsets	61.06	75.26
		20% BS Subsets	68.35	79.46
		10% Random	0.28 ± 0.02	1.03 ± 0.01
		15% Random	0.43 ± 0.00	1.54 ± 0.03
		20% Random	0.67 ± 0.02	2.37 ± 0.02
	10% BS Subsets	10% BS Subsets	65.71 ± 0.08	80.45 ± 0.08
	10% Random	10% Random	55.70 ± 0.24	79.06 ± 0.17
DenseNet-121	10% BS Subsets (from ResNet50)	10% BS Subsets (from ResNet50)	65.67 ± 0.19	81.30 ± 0.10

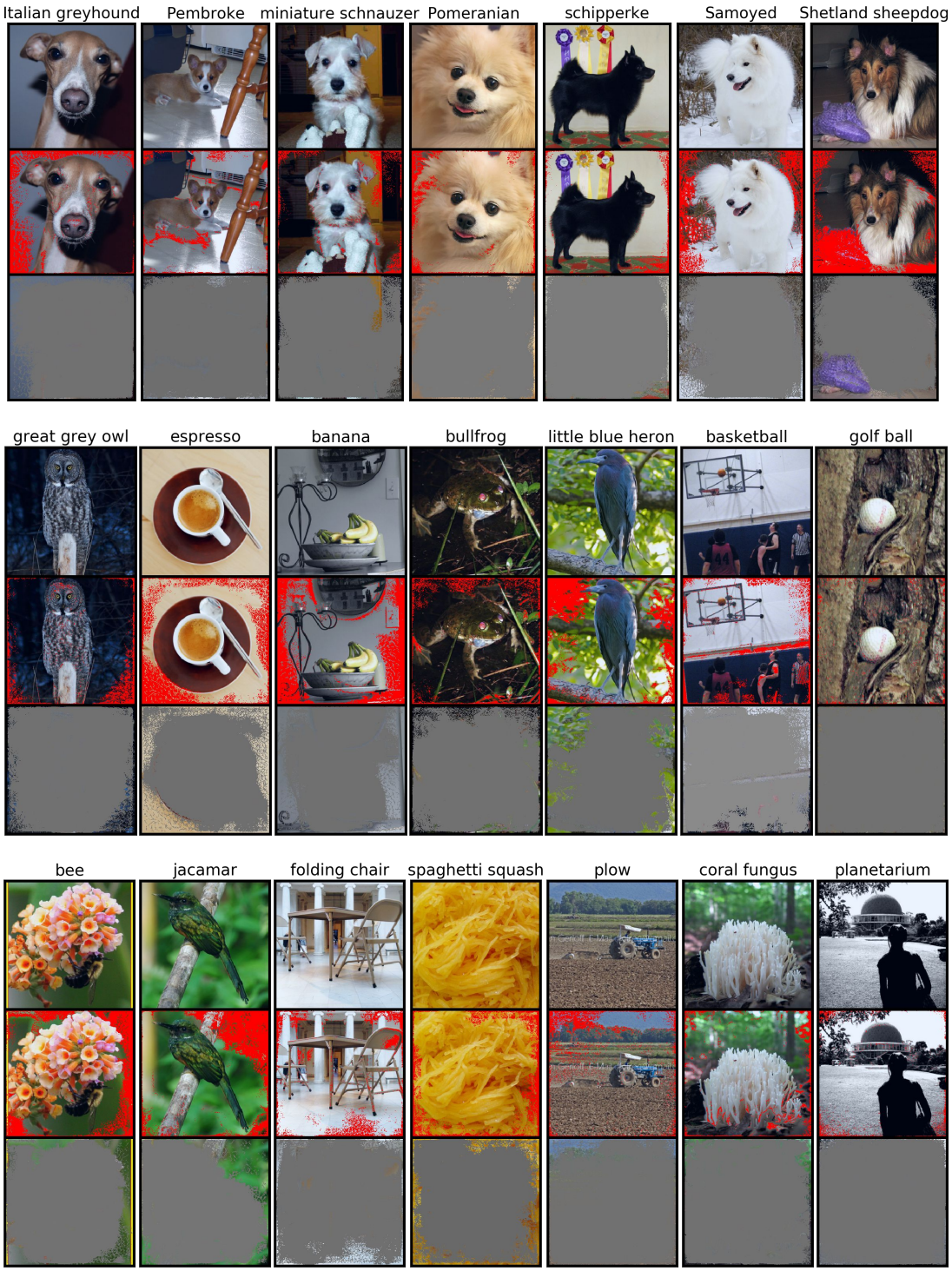


Figure B-12: Example SIS (threshold 0.9) from ImageNet validation images (top row of each block) for Inception v3. The middle rows show the location of SIS pixels (red) and the bottom rows show images with all non-SIS pixels masked but are still classified by the Inception v3 model with $\geq 90\%$ confidence.

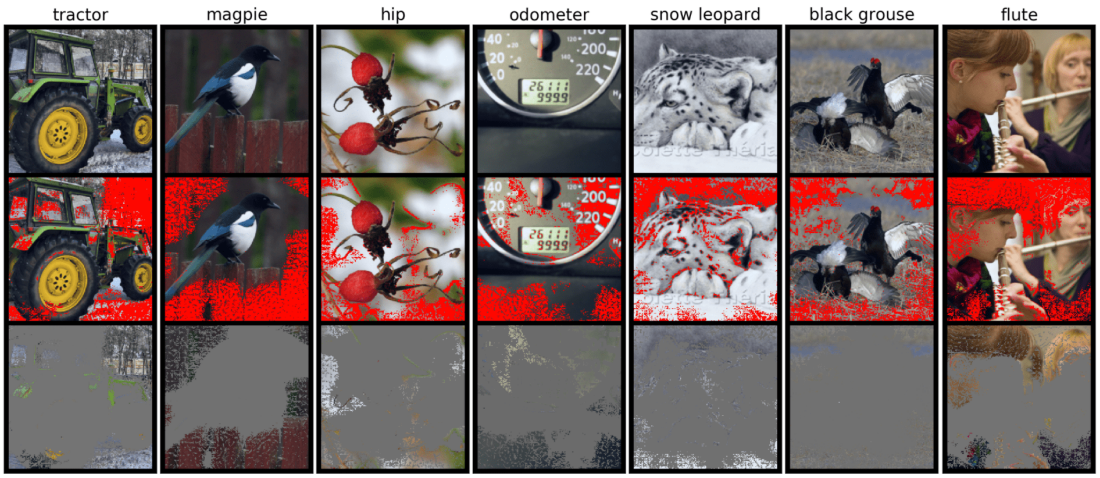


Figure B-13: Example SIS (threshold 0.9) from ImageNet validation images (top row of each block) for ResNet50. The middle rows show the location of SIS pixels (red) and the bottom rows show images with all non-SIS pixels masked but are still classified by the ResNet50 model with $\geq 90\%$ confidence.

We also explored the relationship between pixel saliency and the order pixels were removed by Batched Gradient BackSelect. Surprisingly, as shown in Figure B-14 for Inception v3, we found that the most salient pixels were often *eliminated first* and thus unnecessary for maintaining high predicted confidence on the remaining pixel-subsets and subsequently for training on pixel-subsets. Figure B-15 shows the predicted confidence on remaining pixels at each step of the Batched Gradient BackSelect procedure for a random sample of 32 ImageNet validation images by the Inception v3 model.

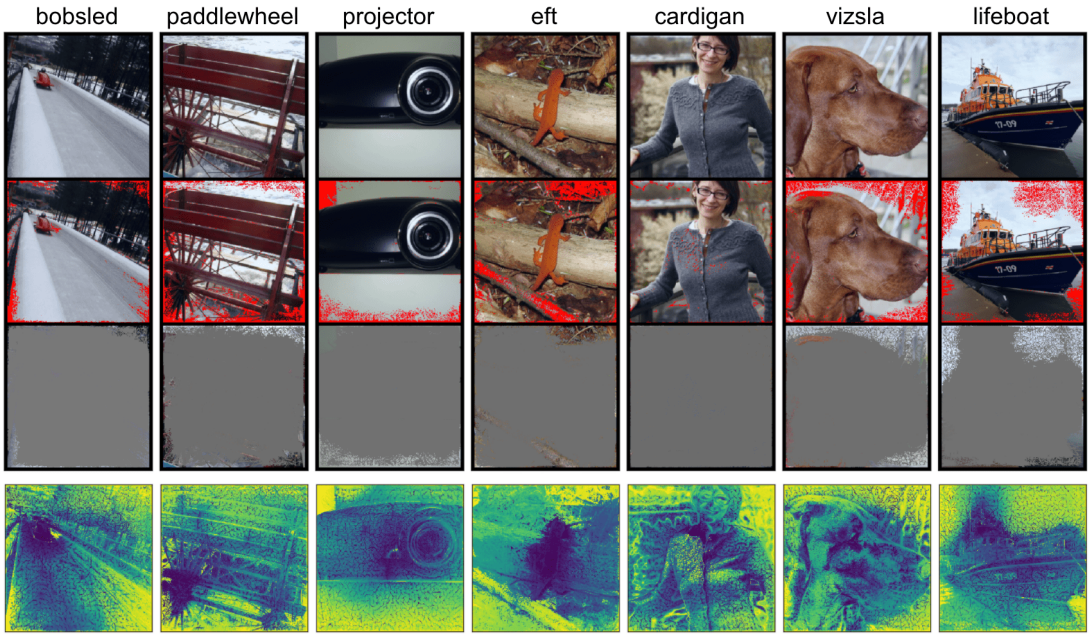


Figure B-14: SIS subsets and ordering of pixels removed by Batched Gradient FindSIS in a sample of ImageNet validation images that are confidently ($\geq 90\%$) and correctly classified by the Inception v3 model. The top row shows original images, second row shows the location of SIS pixels (red), and third row shows images with all non-SIS pixels masked (and are still classified correctly with $\geq 90\%$ confidence). The heatmaps in the bottom row depict the ordering of batches of pixels removed during backward selection (blue = earliest, yellow = latest).

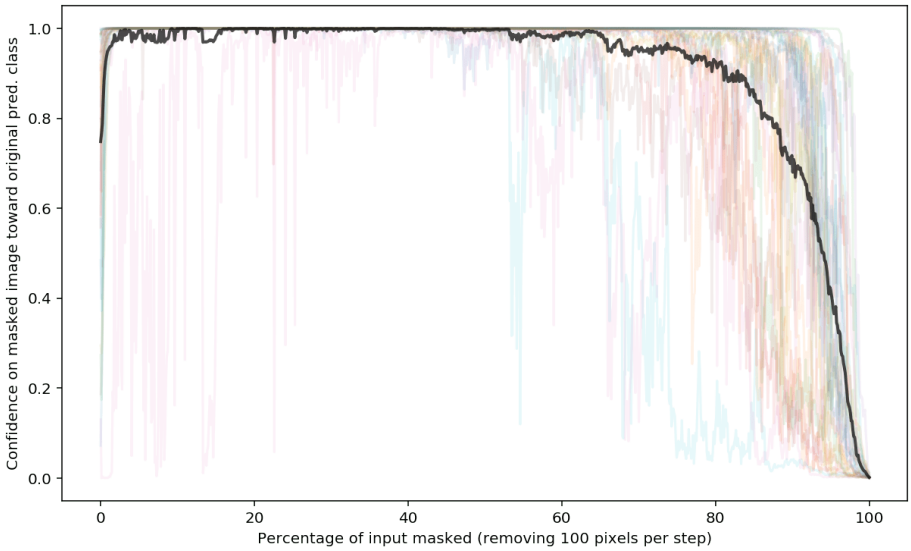


Figure B-15: Prediction history on remaining (unmasked) pixels at each step of the Batched Gradient BackSelect procedure for a random sample of 32 ImageNet validation images by the Inception v3 model. Black line depicts mean confidence at each step.

B.6.3 SIS Size by Class

Figure B-16 shows the distribution of SIS sizes by predicted class (SIS threshold 0.9) for all ImageNet validation images classified with $\geq 90\%$ confidence (23080 images) by the pre-trained Inception v3.

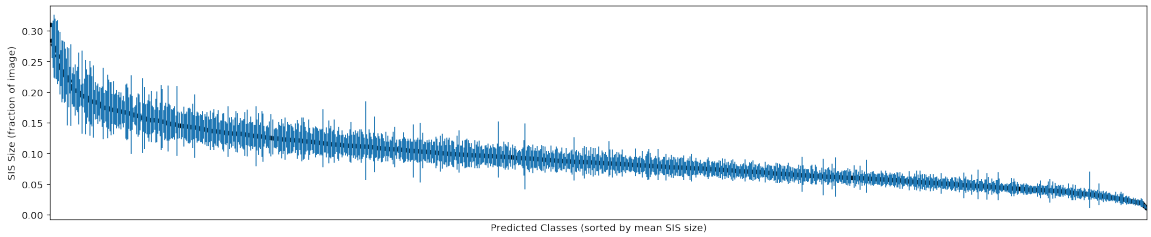


Figure B-16: Mean SIS size per predicted ImageNet class by a pre-trained Inception v3 on ImageNet computed on ImageNet validation images (SIS threshold 0.9). Classes are sorted by mean SIS size. 95% confidence intervals are indicated around each mean. The top 5 classes with largest mean SIS size (mean % of image) are: English foxhound (40.0%), bee eater (28.4%), trolleybus (27.7%), Japanese spaniel (27.3%), whippet (27.0%). The 5 classes with the smallest mean SIS size are: bearskin (1.1%), bath towel (1.3%), wallet (1.4%), fire screen (1.7%), coffeepot (1.9%).

B.6.4 SIS for Vision Transformers

We applied Batched Gradient SIS to a vision transformer (ViT) (Dosovitskiy et al., 2021) as ViTs have been shown to be more robust to perturbations and shifts than CNNs (Naseer et al., 2021). We used a pre-trained `B_16_imagenet1k` ViT model available from GitHub[7], which we found achieves 83.9% top-1 ImageNet validation accuracy. Figure B-17 shows an example of the resulting SIS, suggesting this ViT likewise suffers from overinterpretation on ImageNet data.

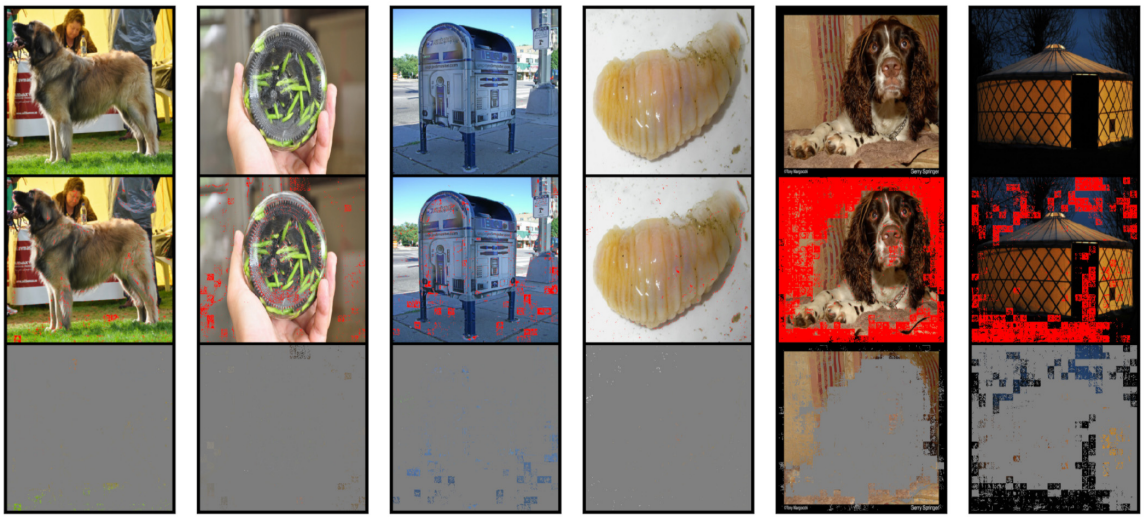


Figure B-17: Example SIS (threshold 0.9) from ImageNet validation images (top row of each block) for a vision transformer (ViT). The middle rows show the location of SIS pixels (red) and the bottom rows show images with all non-SIS pixels masked but are still classified by the ViT model with $\geq 90\%$ confidence.

[7] `https://github.com/lukemelas/PyTorch-Pretrained-ViT`

B.6.5 SIS for SimCLR ResNet50

We applied Batched Gradient SIS to a ResNet50 model trained using the SimCLR contrastive learning framework (Chen et al., 2020). We used a pre-trained `ResNet50 (1x)` SimCLRv1 model available from GitHub[8] converted for PyTorch[9], which we found achieves 68.9% top-1 ImageNet validation accuracy. Note that since this model does not normalize the input data, here our zeros mask corresponds to a black image. We found that on an all-zeros input, the mean predicted confidence over all 1000 ImageNet classes was 0.001, and the maximum confidence toward any class was 0.0065. Figure B-18 shows an example of the resulting SIS.

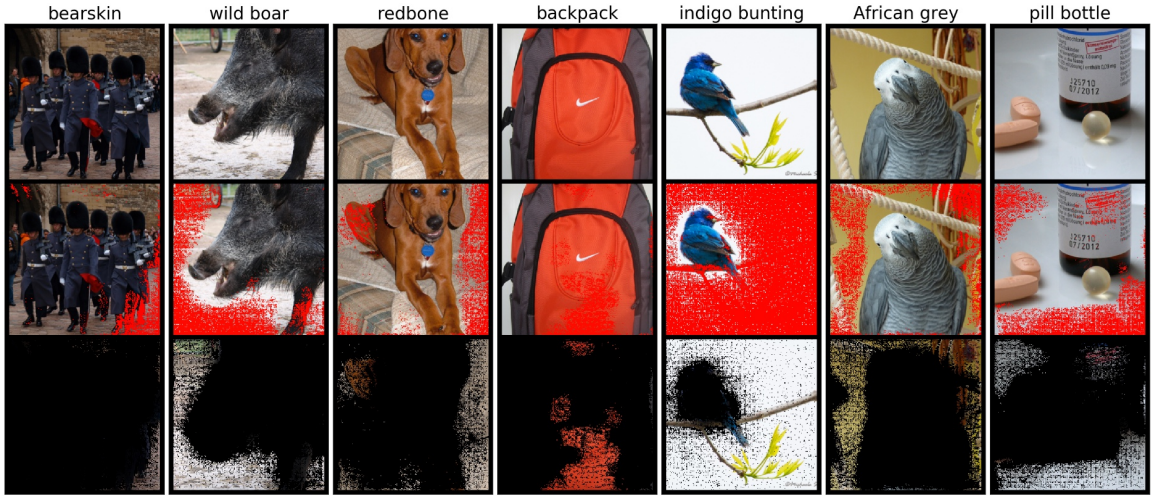


Figure B-18: Example SIS (threshold 0.9) from ImageNet validation images (top row of each block) for a pre-trained SimCLRv1 ResNet50 (1x) model. The middle rows show the location of SIS pixels (red) and the bottom rows show images with all non-SIS pixels masked but are still classified by the SimCLR model with $\geq 90\%$ confidence.

[8] `https://github.com/google-research/simclr`
[9] `https://github.com/tonylins/simclr-converter`

Appendix C

Additional Details for COVID-19 Vaccine Challenge Study

This appendix contains additional details of methods and results for the material in Chapter 5. Further results can be found in the Supplementary Material of Carter et al. (2023), available at: https://www.frontiersin.org/articles/10.3389/fimmu.2023.1135815/full#supplementary-material.

C.1 Supplementary Methods

C.1.1 Mice

A total of 33 male HLA-A*02:01 human transgenic mice (Taconic 9659) were used that were between 12 and 16 weeks old when they received their initial immunization. These CB6F1 background mice carry the MHC class I alleles HLA-A*02:01, H-2-Kb, H-2-Db, H-2-Kd, and H-2-Dd, and H-2-Ld. The mice carry the MHC class II alleles H-2-IAd, H-2-IEd, and H-2-IAb. We also immunized a female cohort of the same transgenic mouse strain (Taconic 9659, 3 mice per vaccine) for immunogenicity measurement, and results are shown in Figures C-8–C-10.

C.1.2 Tissue Culture and Virus

Vero E6 cells (ATCC, CRL:1586) were grown in minimum essential medium (EMEM, Gibco) supplemented with penicillin (100 units/mL), streptomycin (100 µg/mL) and 10% fetal bovine serum (FBS). Two strains of SARS-CoV-2 were used in this study, SARS-CoV-2 (US_WA-1/2020 isolate) and Beta (B.1.351/SA, Strain: hCoV-19/USA/MD-HP01542/2021). Both viruses were propagated and quantified in Vero E6 cells and stored at -80 °C until needed.

C.1.3 MIT-T-COVID Vaccine Design

MHC Class I Epitopes. Eight peptides from SARS-CoV-2 were selected as a subset of the MHC class I *de novo* MIRA only vaccine design of Liu et al. (2021). We filtered this set of 36 peptides to the 8 peptides predicted to be displayed by HLA-A*02:01 by a combined MIRA and machine learning model of peptide-HLA immunogenicity (Liu et al., 2021). The combined model predicts which HLA molecule displayed a peptide that was observed to be immunogenic in a MIRA experiment, and uses machine learning predictions of peptide display for HLA alleles not observed or peptides not tested in MIRA data. Thus, all eight MHC class I peptides in our vaccine were previously observed to be immunogenic in data from convalescent COVID-19 patients (Snyder et al., 2020). We further validated that all peptides are predicted to bind HLA-A*02:01 with high (less than or equal to 50 nM) affinity using the NetMHCpan-4.1 (Reynisson et al., 2020a) and MHCflurry 2.0 (O'Donnell et al., 2020b) machine learning models. For inclusion in the assembled construct, the eight vaccine peptides were randomly shuffled, and alternate peptides were flanked with five additional amino acids at each terminus as originally flanked in the SARS-CoV-2 proteome. The CD4-2 vaccine epitope produced a $CD8^+$ response, and this may be a consequence of a contained MHC class I epitope, with candidates including SLINTLNDL (HLA-A*02:01 predicted binding 55 nM), ASLINTLNDL (H-2-Db predicted binding 289 nM), FYTSKTTVASL (H-2-Kd predicted binding 293 nM), FYFYTSKTTV (H-2-Kd predicted binding 40 nM), and YFYTSKTTV (H-2-Kd predicted

binding 56 nM).

MHC Class II Epitopes. Three peptides from SARS-CoV-2 were optimized for predicted binding to H-2-IAb. We scored all SARS-CoV-2 peptides of length 13–25 using the sliding window approach described in Section 4.1.2 and a machine learning ensemble that outputs the mean predicted binding affinity (IC50) of NetMHCIIpan-4.0 (Reynisson et al., 2020b) and PUFFIN (Zeng and Gifford, 2019). We selected the top three peptides by predicted binding affinity using a greedy selection strategy with a minimum edit distance constraint of 5 between peptides to avoid selecting overlapping windows. All three peptides were flanked with an additional five amino acids per terminus from the SARS-CoV-2 proteome.

The start and end amino acid positions of each vaccine peptide in its origin gene is shown in Table 5.1. For SARS-CoV-2, peptides are aligned to reference proteins in UniProt (Consortium, 2019) (UniProt IDs: P0DTC2 (S), P0DTC3 (ORF3a), P0DTC5 (M), P0DTC9 (N), P0DTD1 (ORF1ab)). All of the epitopes are conserved over these variants of concern: Alpha (B.1.1.7), Beta (B.1.351), Gamma (P.1), Delta (21A, 21J, B.1.617.2), Kappa (B.1.617.1), Epsilon (B.1.427, B.1.429), Iota (B.1.526), Lambda (C.37), Mu (B.1.621) , Omicron (BA.1, BA.2, BA.4, BA.5, BA.2.12.1, BA.2.75, BQ.1, XBB, XBB.1.5), EU1 (B.1.177).

C.1.4 MIT-T-COVID Vaccine Formulation

Codon optimization for mouse expression of the MIT-T-COVID vaccine construct was performed using the IDT Codon Optimization Tool (Integrated DNA Technologies). The resulting nucleic acid sequence is provided in Figure C-1. RNA was synthesized by TriLink BioTechnologies as a modified mRNA transcript with full substitution of 5-Methoxy-U, capped (Cap 1) using CleanCap® AG and polyadenylated (120A). RNA containing lipid nanoparticles were prepared as previously described (Pardi et al., 2017). Briefly, an ethanolic solution of ALC-0315 (Patent WO2017075531), cholesterol, distearoylphosphatidylcholine (DSPC), and 2-[(polyethylene glycol)-2000] N,N ditetradecylacetamide (ALC-0159, Patent Application US14732218) was rapidly

mixed with an solution of RNA in citrate buffer at pH 4.0 (composition described in Patent WO2018081480). Physical properties of the LNP such as size and polydispersity were assessed by Malvern Zetasizer and encapsulation efficiency by Ribogreen assay (Life Technologies).

```
ATG AGG GTC ACA GCT CCT CGG ACC TTG ATC CTC CTT TTG TCT GGT GCT CTT GCA CTG ACT GAG ACT TGG GCC GGG
TCA GGA GGC AGT GGA GGA GGA GGA TCC GGG GGT TAT TTG TAT GCT CTG GTT TAT TTT CTG GGC GGG TCC GGA GGC
GGT GGC TCT GGC GGG AGG TCC AAG AAT CCA CTT CTC TAC GAC GCA AAC TAT TTC TTG TGT TGG CAC ACC AAT GGG
GGG AGC GGT GGC GGA GGA AGC GGT GGG TTC GTG GAC GGA GTT CCC TTT GTT GTT GGT GGG TCA GGC GGA GGA GGC
TCT GGC GGG GCT TAC TAT GTA GGG TAC CTG CAG CCC CGA ACA TTC CTT TTG AAA TAC AAC GAG AAC GGT GGA TCC
GGT GGG GGA GGA AGT GGA GGG TTT CTG AAT AGA TTC ACC ACC ACT CTG GGA GGT TCT GGC GGC GGG GGT TCT GGT
GGA CGG CTG ACT AAA TAC ACA ATG GCC GAT CTT GTT TAC GCA TTG CGG CAT TTT GAT GAG GGA GGC AGT GGC GGA
GGG GGA TCC GGC GGC AGC ATA ATA GCT TAC ACC ATG TCA CTG GGA GGG AGC GGA GGG GGC GGG AGC GGC GGT TTG
CTC CTT TTC GTG ACA GTG TAT AGC CAT CTC CTT CTG GTG GCA GCT GGC CTT GAA GGG GGG AGC GGT GGA GGA GGT
AGC GGT GGC GCC ACT TCT AGG ACA TTG AGT TAC TAT GGG GGC AGC GGA GGA GGA GGT TCT GGA GGC AAG ACC TTC
CCC CCT ACA GAG CCC AAG GGA GGT TCC GGC GGC GGG GGC AGT GGT GGG GAA GAG ATC GCC ATT ATC TTG GCT TCC
TTT AGT GCT TCA ACA AGC GCT TTT GTA GAG ACC GTA AAG GGC CTC GAT TAT GGA GGT TCA GGG GGA GGA GGC TCA
GGT GGG AAG TCA ATA CTG TCT CCT CTT TAT GCA TTT GCA TCA GAG GCT GCA AGA GTT GTC CGA TCC ATT TTT TCT
CGC ACT CTT GGG GGG TCC GGC GGA GGC GGG TCT GGC GGT GTG GAT TAT GGT GCT AGG TTT TAT TTT TAC ACT TCC
AAA ACC ACT GTT GCC TCT CTC ATA AAT ACC CTC AAT GAC CTG GGA GGG TCT GGT GGC GGG GGG AGT GGC GGG AAT
CTT GTT CCT ATG GTT GCA ACA GTA GGA GGT TCC GGA GGG GGT GGC AGC GGA GGA AAG CCA GTG TCC AAG ATG AGA
ATG GCA ACC CCT TTG CTG ATG CAG GCC CTG GGT GGC AGT CTG GGA GGT GGT GGC TCC GGC ATC GTA GGT ATA GTC
GCC GGA CTT GCA GTT TTG GCC GTG GTA GTG ATA GGC GCA GTT GTT GCC ACC GTT ATG TGC CGA CGA AAG AGT TCA
GGC GGC AAG GGT GGT TCT TAC TCC CAA GCT GCA AGC TCC GAC TCC GCT CAG GGG AGT GAT GTT AGC TTG ACT GCA
TGA
```

Figure C-1: Nucleic acid sequence for assembled vaccine construct in Figure 5-1A. Codon optimization for mouse expression was performed using the IDT Codon Optimization Tool (Integrated DNA Technologies).

C.1.5 Animal Immunization

Thirty-three HLA-A*02:01 human transgenic mice were randomly divided into three groups and immunized twice at three-week intervals with vehicle (PBS/300 mM sucrose), 10 µg of Comirnaty® vaccine or 10 µg of MIT-T-COVID vaccine. All vaccines were administered as a 50 µL intramuscular injection. The Comirnaty® vaccine was wastage vaccine that was diluted for human administration (0.9% NaCl diluent), and remaining unusable wastage vaccine in vials was flash frozen at -80 °C. This wastage vaccine was later thawed and immediately administered without dilution (50 µL is 10 µg of mRNA). No Comirnaty® was used that could have been administered to humans. Since Comirnaty® was thawed twice, our results may not be representative of its best performance. The MIT-T-COVID vaccine was diluted to 10 µg in 50 µL with PBS with 300 mM sucrose and then administered. In the female unchallenged cohort, direct peptide immunization was performed to test the importance of mRNA-LNP delivery. Three mice were immunized with an injection of 14 short synthetic peptides in the MIT-T-COVID vaccine (15 µg per peptide, 210 µg in total) adjuvanted with 50 µg of high molecular weight polyinosine-polycytidylic acid (Poly(I:C), InvivoGen, tlrl-pic) in a volume of 150 µL via intramuscular injection. These peptides include 11 MHC class I epitopes and 3 MHC class II epitopes present in the MIT mRNA vaccine (all Table 5.1 epitopes except CD4-3). CD4-3 was not included during peptide/poly IC immunization and was used as a negative control for immunogenicity measurement.

C.1.6 Viral Challenge

At two weeks post booster immunization, eight mice of each group were challenged with 5×10^4 $TCID_{50}$/60 mL of SARS-CoV-2 (B.1.351/SA, Strain: hCoV-19/USA/MD-HP01542/2021) via intranasal (IN) route. Mice were weighted daily and clinically observed at least once daily and scored based on a 1–4 grading system that describes the clinical well-being. Three mice in each group were euthanized at 2 dpi for assessing viral loads and histopathology of the lung. The remaining five mice were continued

monitored for weight changes, other signs of clinical illness, and mortality (if any) for up to 7 dpi before euthanasia for assessing antibody responses within the blood and viral loads and histopathology of the lung. Animal studies were conducted at Galveston National Laboratory at University of Texas Medical Branch at Galveston, Texas, based on a protocol approved by the Institutional Animal Care and Use Committee at UTMB at Galveston.

C.1.7 Assessment of Mortality and Morbidity

Differentially immunized and challenged mice were monitored at least once each day for the morbidity and mortality and assigned the clinical scores based on the following: 1: Healthy, 2: ruffled fur, lethargic, 3: hunched posture, orbital tightening, increased respiratory rate, and/or $> 15\%$ weight loss, and 4: dyspnea and/or cyanosis, reluctance to move when stimulated or $> 20\%$ weight loss.

C.1.8 Immunogenicity Measurements

At 14 days post booster immunization, three mice of each group were sacrificed for harvesting splenocytes in 2 mL R10 medium (RPMI, 10% FBS, 1%P/S, 10 mM HEPES). Briefly, spleens were homogenized and subjected to filtration onto 40 µm cell strainers, followed by a wash of strainers with 10 mL PBS and centrifuged at 500 g for 5 min at 4 °C. Cell pellet was resuspended by using 2 mL of 1x red blood cell lysis buffer (eBioscience) for 2–3 min, followed by a supplement of 20 mL PBS. Resulting cell suspensions were centrifuged at 500 g for 5 min, resuspended in 2 mL R10 medium before counting the numbers under a microscope. For a brief in vitro stimulation, aliquots of 10^6 cells were incubated with indicated peptide at a final concentration of 1 µg/mL in each well of a 96-well plate. GolgiPlug (5 µg/mL, BD Bioscience) was added into the culture at 1 hr post stimulation and followed by an additional 4 hrs incubation. For cell surface staining, cultured splenocytes were resuspended in 40 µL FACS buffer containing fluorochrome-conjugated antibodies and incubated 1 hr at 4 °C followed by cell fixation using Cytofix/Cytoperm buffer (BD Bioscience) for

20 min at 4 °C. Cells were further incubated with fluorochrome-conjugated cytokine antibodies overnight. The next day, cells were washed twice using 1x Perm/Wash buffer and resuspended in 300 µL FACS buffer for analysis. For Foxp3 staining, cells were fixed and permeabilized by Foxp3/Transcription Factor Staining Buffer Set (Thermo Fisher Scientific) according to the instruction. Fixable Viability Dye eFluor506 (Thermo Fisher Scientific) was also used in all sample staining to exclude dead cells from our data analysis. The fluorochrome-conjugated anti-mouse antibodies included: FITC-conjugated CD3 (17A2, Biolegend), efluor450-conjugated CD4 (GK1.5, eBioscience), PE-Cy7-conjugated CD8 (53-6.7, eBioscience), PE-conjugated IFN-γ (XMG1.2, eBioscience), PerCP-eFlour710- conjugated TNF-α (MP6-XV22, eBioscience), APC-conjugated IL-2 (JES6-5H4, eBioscience). For the Foxp3 staining: Pacific Blue-conjugated CD4 (GK1.5), FITC-conjugated CD25 (PC61), Percp-Cy5.5-conjugated CTLA4 (UC-10-4B9), PE-conjugated Foxp3 (FJK-16s). For the $CD44^+$ T cell analysis (female cohort only): FITC-conjugated CD3 (17A2, Biolegend), efluor450-conjugated CD4 (GK1.5, eBioscience), PE-Cy7-conjugated CD8 (53-6.7, eBioscience), APC-conjugated mouse/human CD44 (IM7, BioLegend). Cell acquisition was performed a BD LSR Fortessa and data were analyzed using BD FACSDiva 9.0 and FlowJo 10 (FlowJo, LLC). Lymphocytes were defined by SSC-A vs. FSC-A plots. Singlet cells were defined by FSC-H vs. FSC-A plots. Dead cells were excluded by positive staining with viability dye. $CD4^+$ and $CD8^+$ T cells were gated from $CD3^+$ cells. The cytokines secreting $CD4^+$ and $CD8^+$ T cells were then identified with IFN-γ, TNF-α, and IL-2 expression. Boundaries between positive and negative cells for the given marker were defined by the fluorescence minus one (FMO) control and adjusted according to the unstimulated splenocyte group. For the Treg cell gating strategy, $CD4^+$ T cells were gated from $CD3^+$ cells and identified by the positivity of Foxp3 and CD25 staining.

C.1.9 Viral Titer Assay

For virus quantitation, the frozen lung specimens were weighed before homogenization in PBS/2% FBS solution using the TissueLyser (Qiagen), as previously de-

scribed (Tseng et al., 2007). The homogenates were centrifuged to remove cellular debris. Cell debris-free homogenates were used to quantifying infectious viruses in the standard Vero E6 cell-based infectivity assays in 96-well microtiter plates, as we routinely used in the lab (Tseng et al., 2012). Titers of virus were expressed as 50% tissue culture infectious dose per gram of tissue ($TCID_{50}/g$).

C.1.10 Antibody Neutralization Assay

Sera of mice collected at 7 dpi were used for measuring specific antibody responses. Briefly, sera were heat-inactivated (56 °C) for 30 min, were stored at -80 °C until needed. For determining the SARS-CoV-2 neutralizing antibody titers, serially two-fold (starting from 1:40) and duplicate dilutions of heat-inactivated sera were incubated with 100 TCID of SARS-CoV-2 (US_WA-1/2020 isolate) or Beta (B.1.351/SA, Strain: hCoV-19/USA/MD-HP01542/2021) at 37 °C for 1 h before transferring into designated wells of confluent Vero E6 cells grown in 96-well microtiter plates. Vero E6 cells cultured with medium with or without virus were included as positive and negative controls, respectively. After incubation at 37 °C for 3 days, individual wells were observed under the microscopy for the status of virus-induced formation of cytopathic effect. The 100% neutralizing titers (NT_{100}) of sera were expressed as the lowest dilution folds capable of completely preventing the formation of viral infection-induced cytopathic effect in 100% of the wells.

C.1.11 Serum IgG/IgM Response by ELISA

ELISA was applied to verified serum IgG and IgM responses with SARS-CoV-2 (US_WA-1/2020 isolate) infected Vero-E6 cell lysate. In brief, SARS-CoV-2 infected Vero-E6 cell lysate was coated on 96-well plate (Corning) at 1 µg/well in PBS for overnight at 4 °C. The plates were blocked using 1% BSA/PBST for 1 h at room temperature. The 5-fold serial-dilution serum from each mouse (starting at 1:100) was then added into antigen-coated plates and incubated for 1 h at 37 °C. The plates were then washed three times with PBST (PBS/0.1% Tween-20) followed by incuba-

tion with 100 µL of anti-mouse IgG and IgM HRP conjugated secondary antibody (Jackson immunoresearch) (1:2000) for 1 h at 37 °C. After three times wash using PBST, 100 µL of ABTS substrate (Seracare) was added to the plates and incubated for 30 min in the dark. After stopped by adding 1% SDS. Then absorbance at 405 nm (OD 405 nm) was measured with the plate reader (Molecular Devices) and analyzed with GraphPad Prism Version 9.1.2.

C.1.12 RNA Extraction and Quantitative RT-PCR

Lung tissues were weighted and homogenized in 1 mL of Trizol reagent (Invitrogen) using TissueLyser (Qiagen). The RNA was then extracted using Direct-zol RNA miniprep kits (Zymo research) according to the manufacturer's instructions. 500 ng total RNA was then applied to cDNA synthesis using iScript cDNA Synthesis kit (Biorad) according to the manufacturer's instructions. The viral genomic RNA, subgenomic RNA and mice 18s rRNA has then been amplified using iQ SYBR green supermix (Biorad) and performed using CFX96 real time system (Biorad). The samples were run in duplicate using the following conditions: 95 °C for 3 min then 45 cycles of 95 °C for 15 s and 58 °C for 30 s. The level of expression was then normalized with 18s rRNA and calculated using the $2^{-\Delta\Delta C_T}$ method, as we have previously described (Tseng et al., 2007; Agrawal et al., 2015).

The primer set for SARS-CoV-2 RNA amplification is nCoV-F (ACAGGTACGTTAATAGTTAATAGCGT) and nCoV-R (ATATTGCAGCAGTACGCACACA). SgLeadSARS2-F (CGATCTCTTGTAGATCTGTTCTC) and nCoV-R (ATATTGCAGCAGTACGCACACA) were used for subgenomic SARS-CoV-2 RNA amplification. Mouse 18s rRNA was served as the internal control and amplified using 18s-F (GGACCAGAGCGAAAGCATTTGCC) and 18s-R (TCAATCTCGGGTGGCTGAACGC).

C.1.13 Immunohistochemistry

All slides were prepared by the Histopathology Core (UTMB) into 5 µm paraffin-embedded sections for immunohistochemistry (IHC). IHC staining and analysis were

performed by UTMB according to previously published protocols (Tseng et al., 2007; Yoshikawa et al., 2009). In brief, a standard IHC sequential incubation staining protocol was followed to detect the SARS-CoV-2 spike (S) protein, $CD4^+$ cells, or $CD8^+$ cells using a rabbit-raised anti-SARS-CoV-2 S protein antibody (1:5000 dilution, ab272504, Abcam plc, Cambridge UK), a rabbit monoclonal anti-CD4 antibody (1:250 dilution, ab183685, ab183685, Abcam plc, Cambridge, UK), and a rabbit monoclonal anti-CD8 antibody (1:500 dilution, ab217344, Abcam plc, Cambridge, UK) followed by peroxidase-conjugated secondary antibody and 3,3'-Diaminobenzidine (DAB) substrate kit (MP-7802, Vector Laboratories, Burlingame, CA). Slides were counterstained with hematoxylin (MHS16-500ML, Sigma-Aldrich Inc., St. Louis, MO) and antigen expression was examined under 10X and 40X magnifications using an Olympus IX71 microscope.

$CD4^+$ and $CD8^+$ cell counts were quantified using CellProfiler 4.2.4 (Stirling et al., 2021). The image analysis pipeline included the following modules: (1) UnmixColors (for each of DAB and hematoxylin stains), (2) ImageMath (applied to DAB stain images only; subtract 0.5 from all intensity values and set values less than 0 equal to 0), and (3) IdentifyPrimaryObjects (require object diameter 20–60 pixels [DAB stain] or 15–60 pixels [hematoxylin stain]; Global threshold strategy; Otsu thresholding method with two-class thresholding, threshold smoothing scale 1.3488, threshold correction factor 1.0, lower threshold bound 0.0, upper threshold bound 1.0, no log transform before thresholding; Shape method to distinguish clumped objects and draw dividing lines between clumped objects; automatically calculate size of smoothing filter for declumping; automatically calculate minimum allowed distance between local maxima; speed up by using lower-resolution image to find local maxima; fill holes in identified objects after declumping only; continue handling objects if excessive number of objects identified).

C.1.14 Statistical Analysis

One-way ANOVA tests were performed in Python using the SciPy package (Virtanen et al., 2020). Two-way ANOVA and Tukey's tests were performed in Python using

the statsmodels package (Seabold and Perktold, 2010). Logrank tests were performed using GraphPad Prism.

C.2 Supplementary Figures

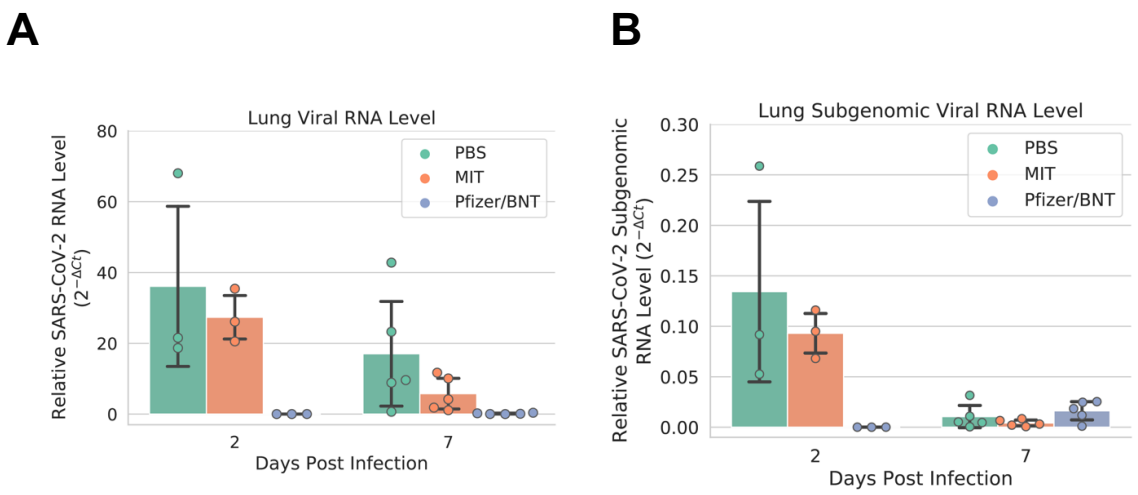


Figure C-2: (A) Lung viral RNA level, and (B) lung subgenomic viral RNA level. Error bars indicate the standard deviation around each mean.

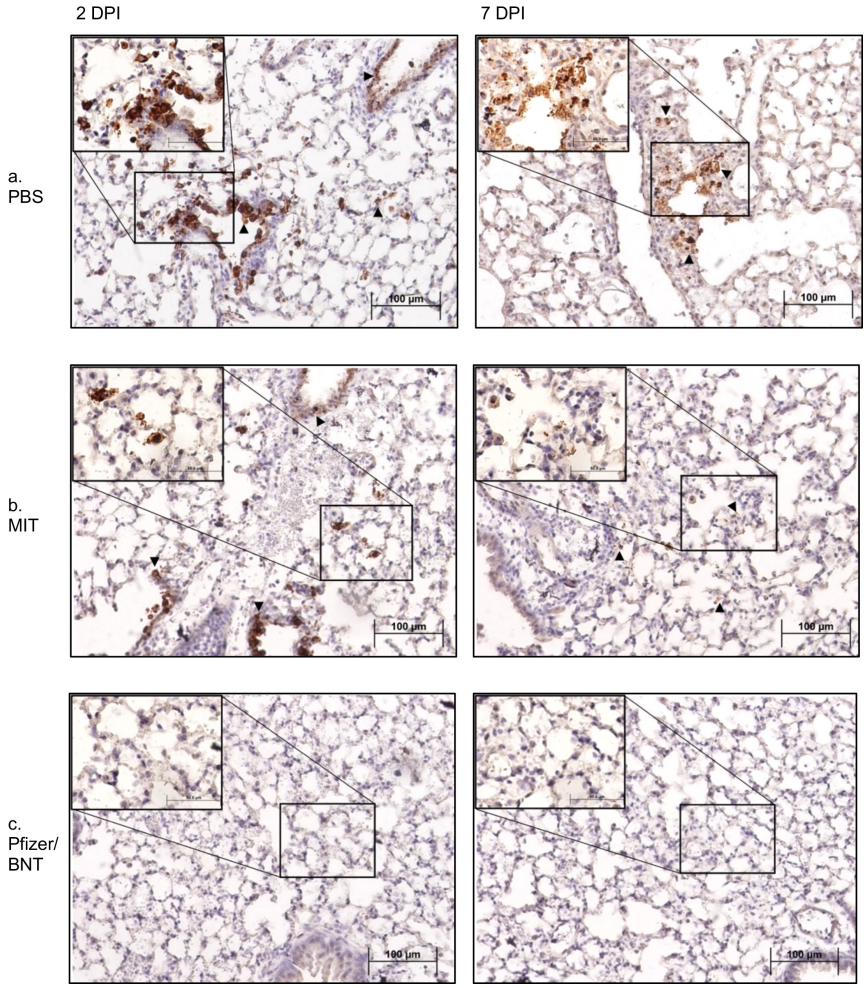


Figure C-3: All lung samples were subjected to IHC staining for SARS-CoV-2 spike protein (brown) with hematoxylin counterstain (blue), representative images of which are shown here. Inset images taken at 40X magnification. Black arrowheads indicate selected areas of viral infection. (a) Specimens immunized with PBS exhibited extensive staining indicative of viral infection throughout the epithelium of both the bronchioles and the alveolar sacs, with the viral infection appearing more intense at 2 days post infection (dpi, left) than at 7 dpi (right). Although viral infection is significantly reduced by 7 dpi, viral antigen was still readily detectable throughout alveoli. (b) In comparison, specimens immunized with the MIT-T-COVID vaccine exhibited similarly extensive viral infection at 2 dpi (left) throughout the bronchiolar and alveolar epithelia, albeit somewhat reduced in intensity. However, by 7 dpi (right), viral infection was significantly reduced in both extent and intensity, with brown puncta being detected only in a few alveoli scattered throughout the tissue. (c) Contrasted with both PBS and MIT-T-COVID-immunized specimens, the Pfizer/BNT-immunized specimens exhibited significantly reduced viral infections at both 2 (left) and 7 dpi (right). With the exception of a single area at 7 dpi (see Figure C-4), viral antigen was undetected at both timepoints.

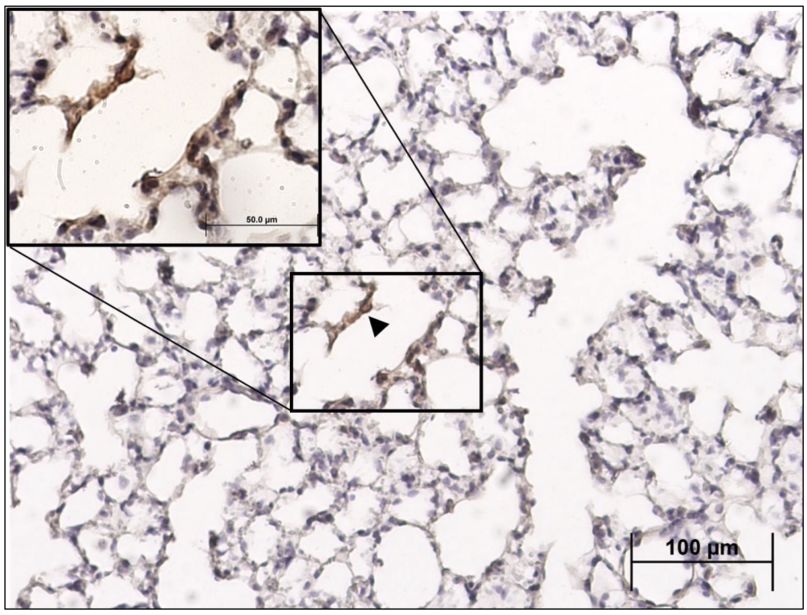


Figure C-4: Sole region of detectable SARS-CoV-2 spike antigen in alveoli of Comirnaty®-immunized lung specimens at 7 days post infection. Black arrowhead indicates infected cell. See also Figure C-3.

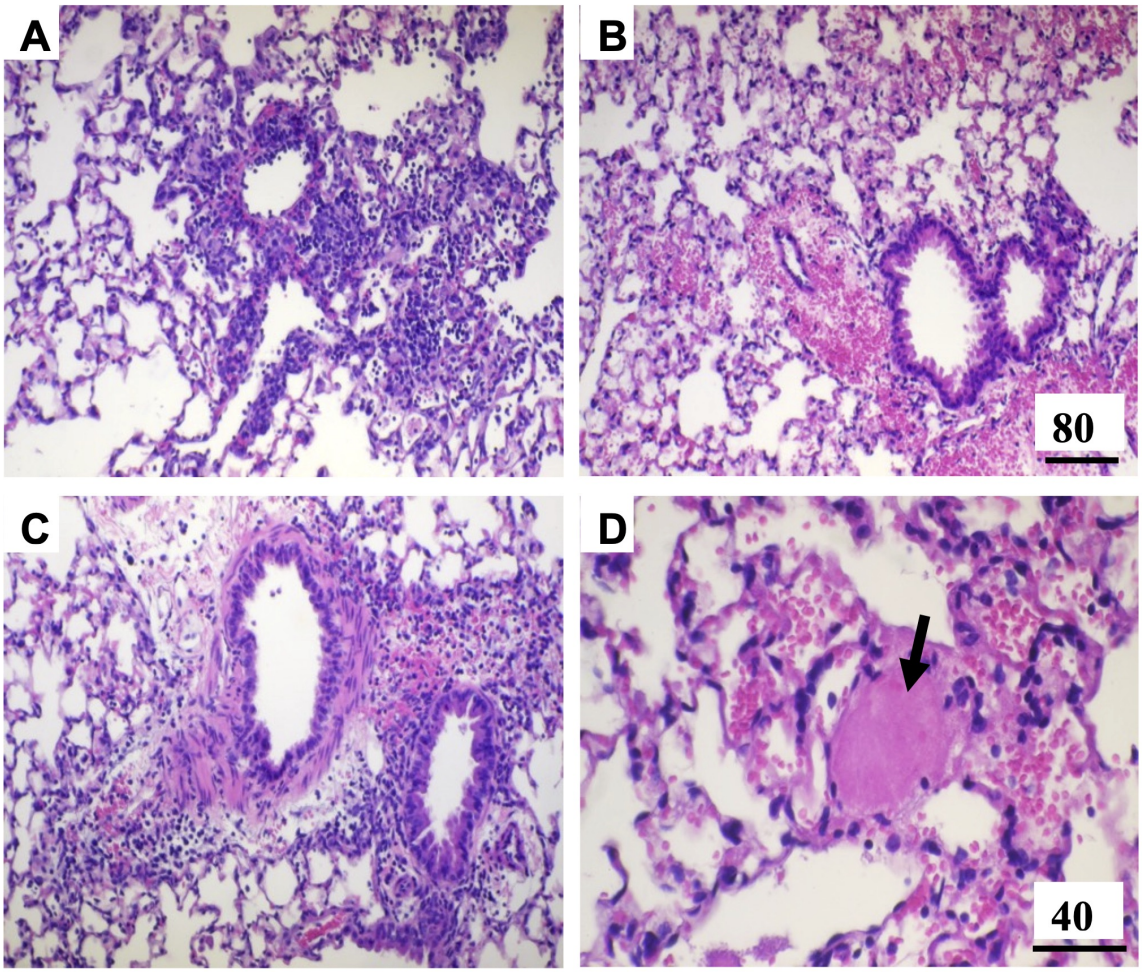


Figure C-5: Lung histopathology. Lungs of mice immunized with MIT-T-COVID (A) or Pfizer/BNT (B) are compared with those with PBS (C). At 7 dpi, the MIT-T-COVID-immunized group showed extensive lymphocytic infiltrations in perivascular regions and spaces around bronchi, bronchioles, and alveoli. There are fewer infiltrations found in the Pfizer/BNT or PBS groups and they are only localized at perivascular regions around bronchi and large bronchioles. There is widespread congestion along with hemorrhage and few foci of thromboembolism (arrow in D) seen in the Pfizer/BNT group but not others. Bar = 80 µm in A, B and C; Bar = 40 µm in D.

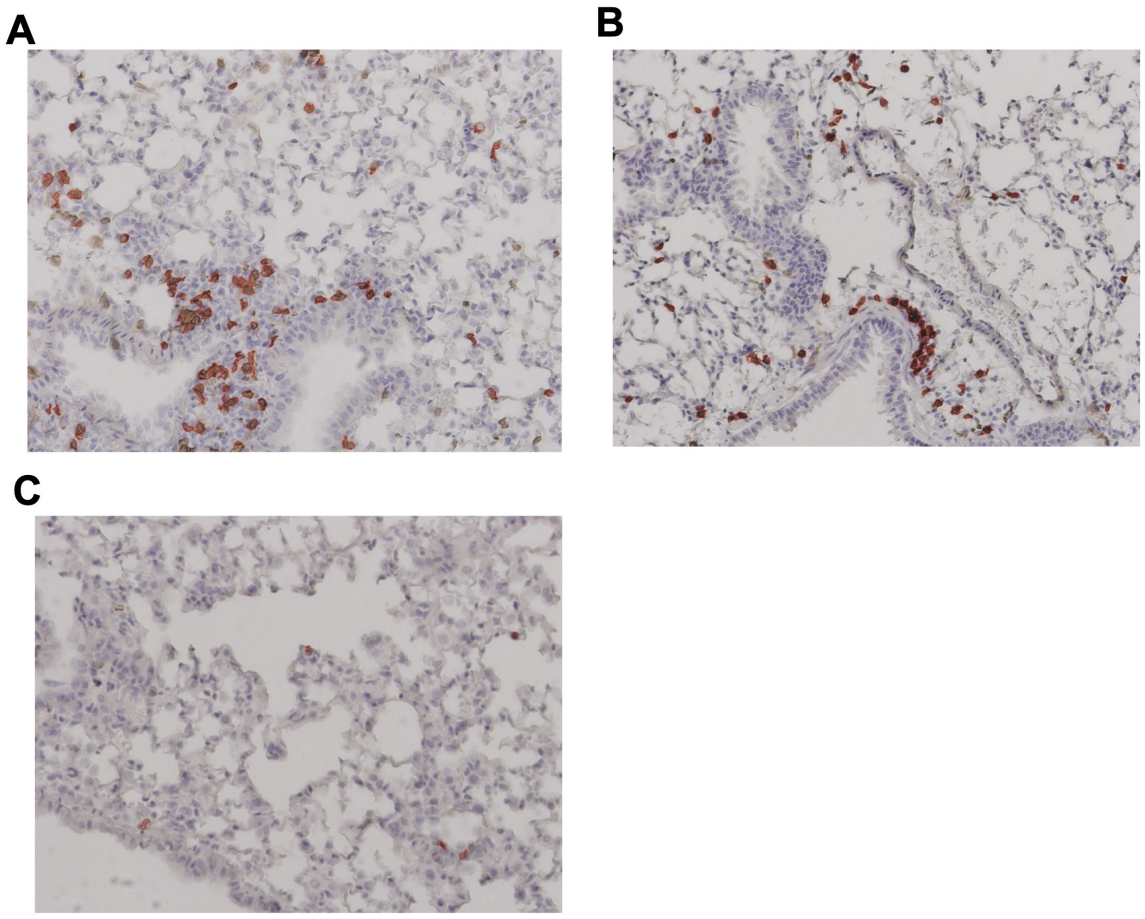


Figure C-6: Lung immunohistochemistry for $CD4^+$ cells at 7 dpi. Example $CD4^+$ stain images for (A) MIT-T-COVID, (B) Pfizer/BNT, and (C) PBS-immunized animals. Lung samples were subjected to IHC staining for CD4 (brown) with hematoxylin counterstain (blue). Images were taken at 10X magnification. Red outlines indicate $CD4^+$ cells identified and counted by CellProfiler software (Appendix C.1). See also Figure 5-4 and Figure C-7.

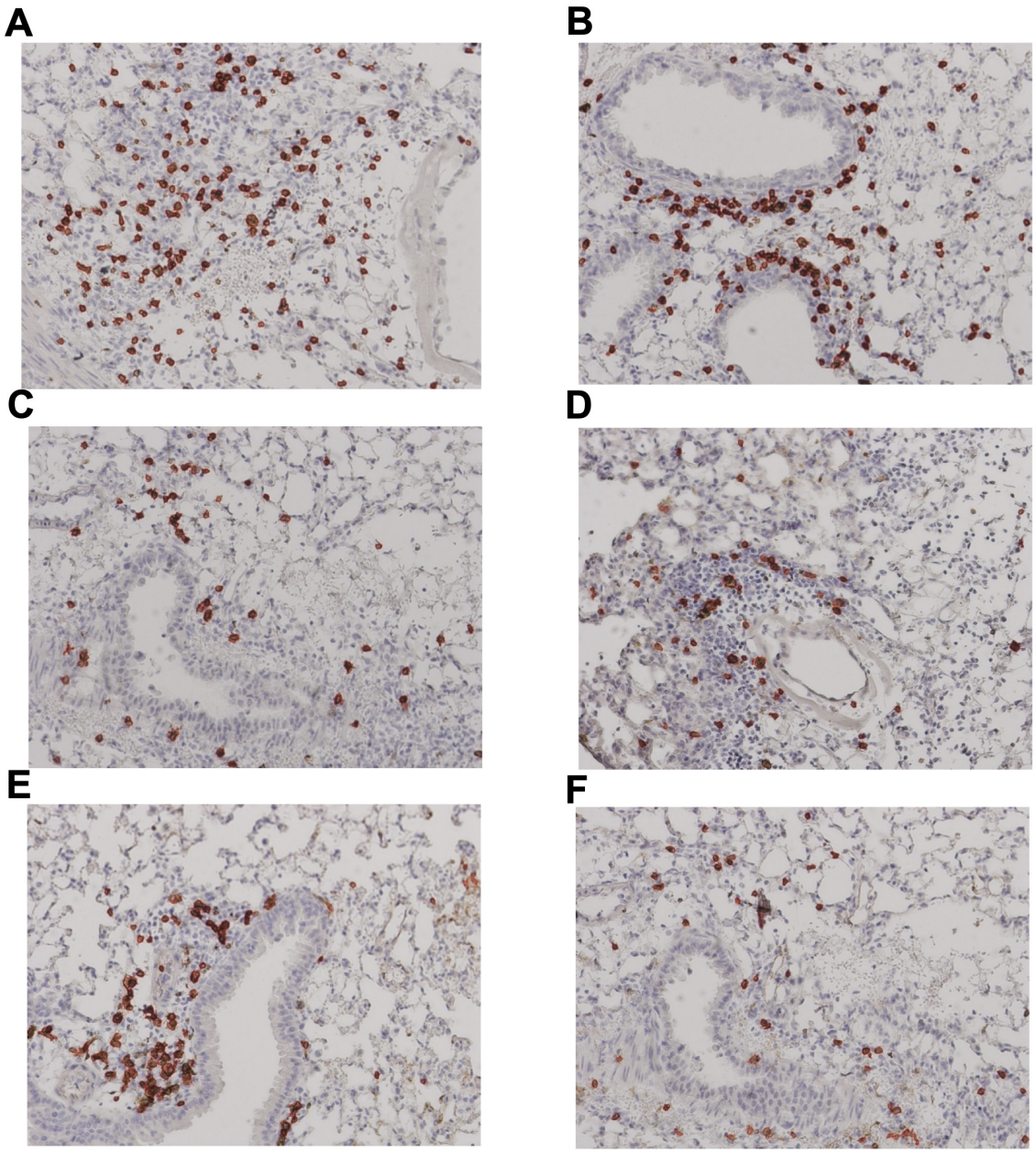


Figure C-7: Lung immunohistochemistry for $CD8^+$ and $CD4^+$ cells at 2 dpi. Example $CD8^+$ stain images for (A) MIT-T-COVID, (B) Pfizer/BNT, and (C) PBS-immunized animals. Example $CD4^+$ stain images for (D) MIT-T-COVID, (E) Pfizer/BNT, and (F) PBS-immunized animals. Lung samples were subjected to IHC staining for CD4 (brown) with hematoxylin counterstain (blue). Images were taken at 10X magnification. Red outlines indicate cells identified and counted by CellProfiler software (Appendix C.1). See also Figure 5-4 and Figure C-6.

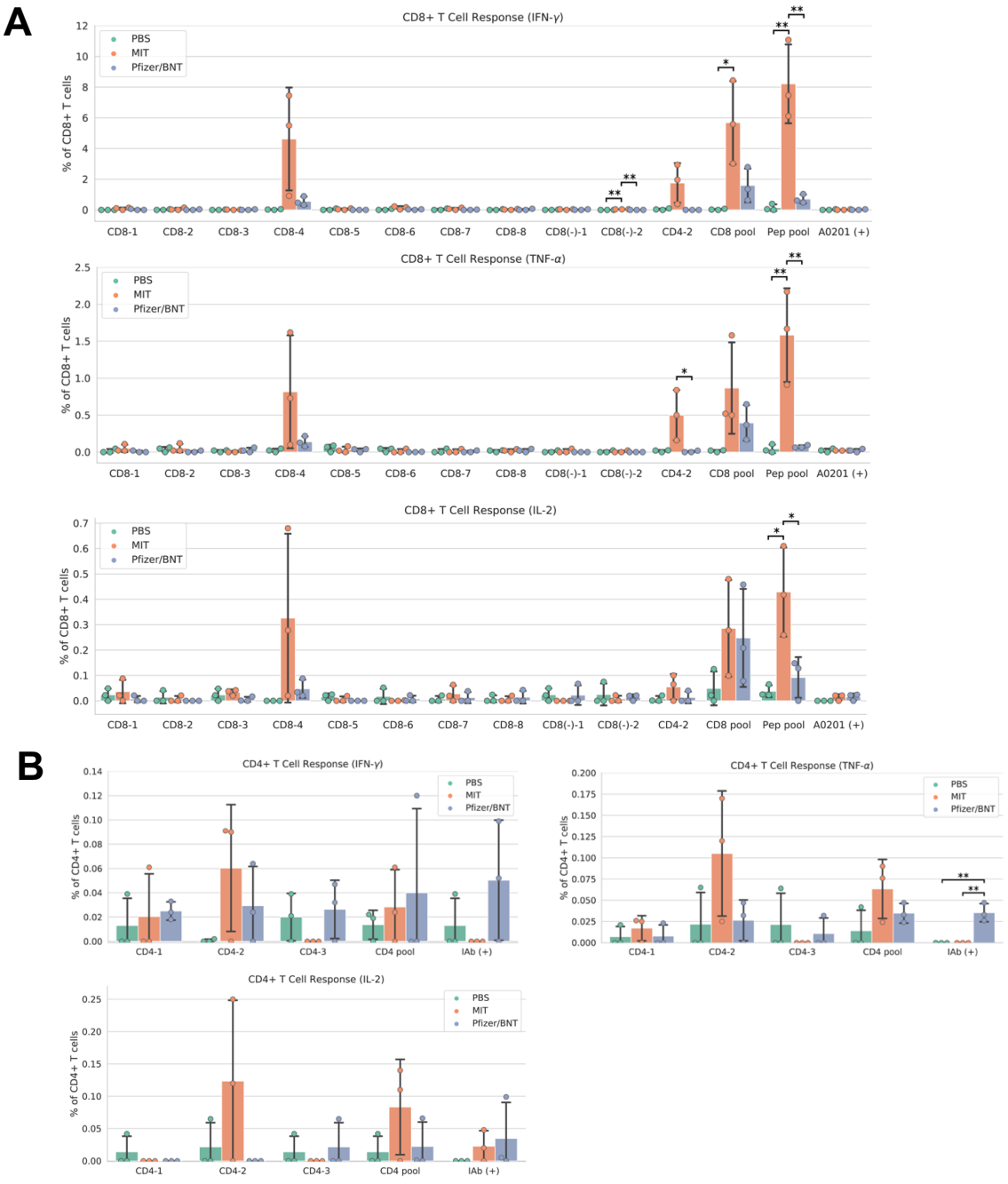


Figure C-8: Vaccine immunogenicity in female mouse cohort (Appendix C.1). (A) $CD8^+$ T cell responses, (B) $CD4^+$ T cell responses. The CD8 pool includes MHC class I peptides CD8-1–CD8-8 (Table 5.1). The CD4 pool includes MHC class II peptides CD4-1, CD4-2, and CD4-3. The Pep pool includes all query peptides in Table 5.1. Error bars indicate the standard deviation around each mean. P values were computed by one-way ANOVA with Tukey's test. $^*P < 0.05$, $^{**}P < 0.01$. See also Figure C-9.

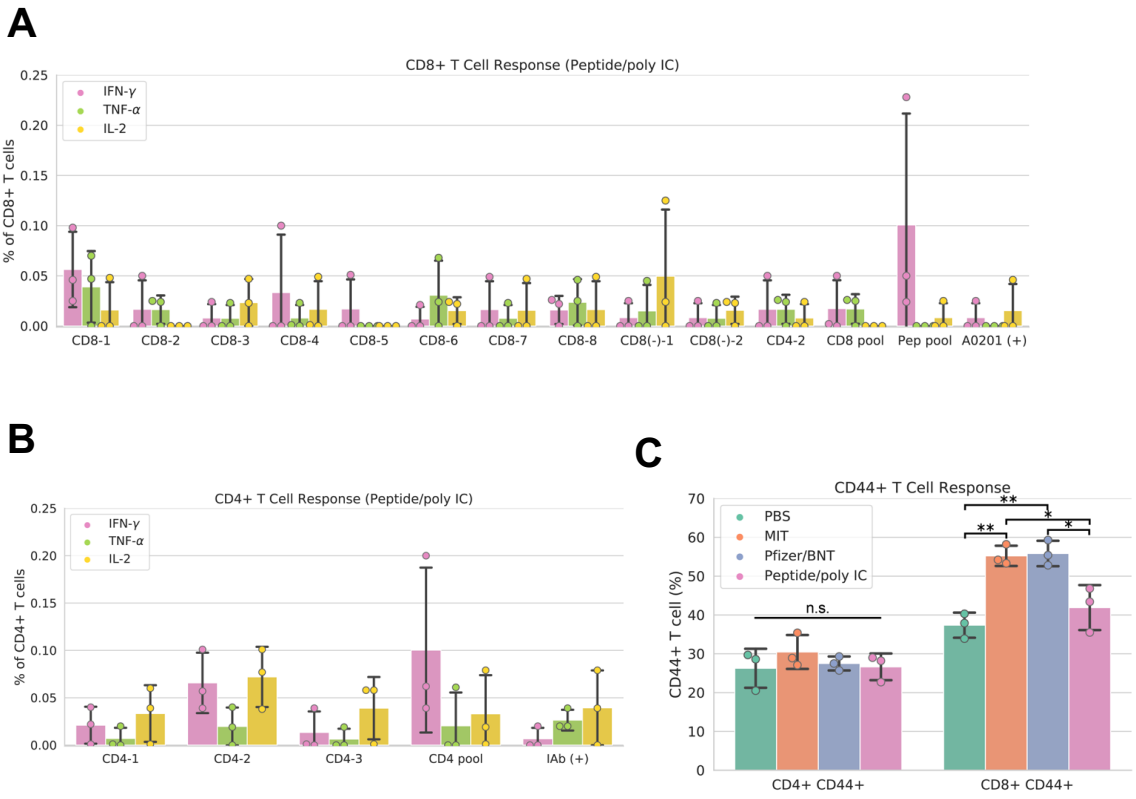


Figure C-9: Immunogenicity of Peptide/poly IC immunization in female mouse cohort (Appendix C.1). (A) $CD8^+$ T cell responses, (B) $CD4^+$ T cell responses, and (C) $CD44^+$ T cell responses. The CD8 pool includes MHC class I peptides CD8-1–CD8-8 (Table 5.1). Mice were immunized with all Table 5.1 epitopes except CD4-3 (negative control). The CD4 pool includes MHC class II peptides CD4-1, CD4-2, and CD4-3. The Pep pool includes all query peptides in Table 5.1. Error bars indicate the standard deviation around each mean. P values in (C) were computed by one-way ANOVA with Tukey's test. $^*P < 0.05$, $^{**}P < 0.01$, n.s. = not significant. See also Figure C-8.

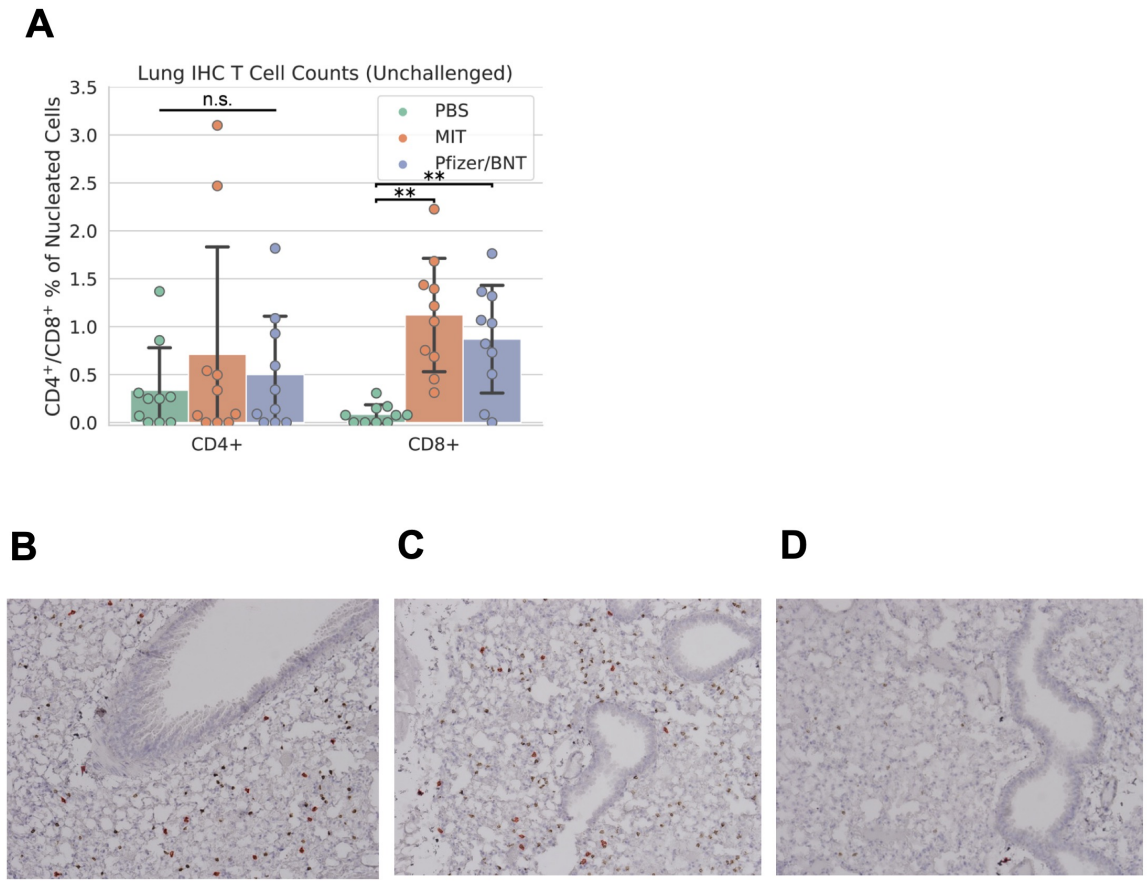


Figure C-10: Lung immunohistochemistry for $CD8^+$ and $CD4^+$ cells in unchallenged female mouse cohort. (A) Counts of $CD8^+$ and $CD4^+$ T cells expressed as a percentage of all nucleated cells visible in each field from lung tissue. Example $CD8^+$ stain images for (B) MIT-T-COVID, (C) Pfizer/BNT, and (D) PBS-immunized animals. Lung samples were subjected to IHC staining for CD8 (brown) with hematoxylin counterstain (blue). Images were taken at 10x magnification. Red outlines in (B)–(D) indicate $CD8^+$ cells identified and counted by CellProfiler software (Appendix C.1). Error bars indicate the standard deviation around each mean. P values were computed by one-way ANOVA with Tukey's test. $^{**}P < 0.01$, n.s. = not significant.

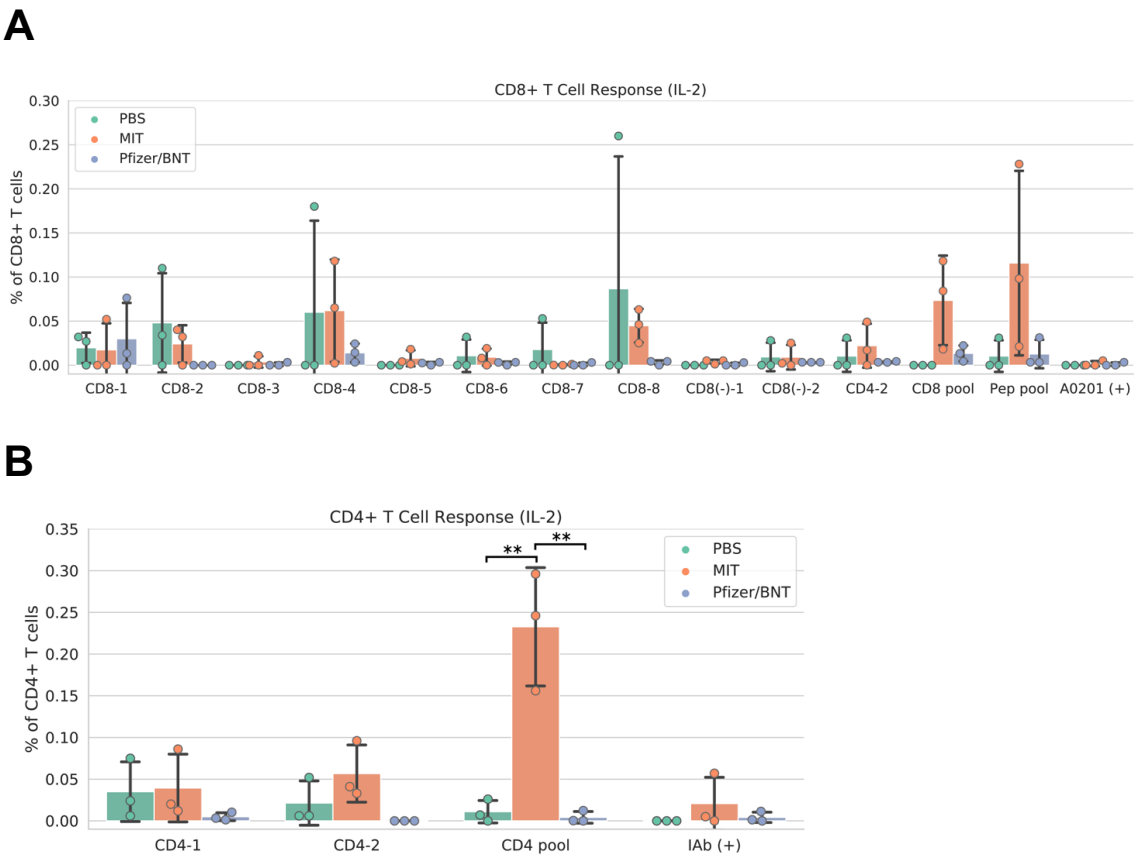


Figure C-11: Vaccine immunogenicity interleukin-2 (IL-2) measurements (male cohort). (A) $CD8^+$ T cell responses, (B) $CD4^+$ T cell responses. The CD8 pool includes MHC class I peptides CD8-1–CD8-8 (Table 5.1). The CD4 pool includes MHC class II peptides CD4-1 and CD4-2. The Pep pool includes all query peptides in Table 5.1 except CD4-3. Error bars indicate the standard deviation around each mean. P values were computed by one-way ANOVA with Tukey's test. $^{**}P < 0.01$. See also Figure 5-2.

Bibliography

Abdelmageed, M. I., Abdelmoneim, A. H., Mustafa, M. I., Elfadol, N. M., Murshed, N. S., Shantier, S. W., and Makhawi, A. M. (2020). Design of multi epitope-based peptide vaccine against E protein of human 2019-nCoV: An immunoinformatics approach. *bioRxiv*.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*.

Agarwal, C. and Nguyen, A. (2020). Explaining image classifiers by removing input features using generative models. In *Proceedings of the Asian Conference on Computer Vision*.

Agrawal, A. S., Garron, T., Tao, X., Peng, B.-H., Wakamiya, M., Chan, T.-S., Couch, R. B., and Tseng, C.-T. K. (2015). Generation of a transgenic mouse model of Middle East respiratory syndrome coronavirus infection and disease. *Journal of Virology*, 89(7):3659–3670.

Ahmed, S. F., Quadeer, A. A., and McKay, M. R. (2020). Preliminary identification of potential vaccine targets for the COVID-19 coronavirus (SARS-CoV-2) based on SARS-CoV immunological studies. *Viruses*, 12(3):254.

Akhand, M. R. N., Azim, K. F., Hoque, S. F., Moli, M. A., Joy, B. D., Akter, H., Afif, I. K., Ahmed, N., and Hasan, M. (2020). Genome based evolutionary study of SARS-CoV-2 towards the prediction of epitope based chimeric vaccine. *bioRxiv*.

Arunachalam, P. S., Charles, T. P., Joag, V., Bollimpelli, V. S., Scott, M. K. D., Wimmers, F., Burton, S. L., Labranche, C. C., Petitdemange, C., Gangadhara, S., Styles, T. M., Quarnstrom, C. F., Walter, K. A., Ketas, T. J., Legere, T., Jagadeesh Reddy, P. B., Kasturi, S. P., Tsai, A., Yeung, B. Z., Gupta, S., Tomai, M., Vasilakos, J., Shaw, G. M., Kang, C.-Y., Moore, J. P., Subramaniam, S., Khatri, P., Montefiori, D., Kozlowski, P. A., Derdeyn, C. A., Hunter, E., Masopust, D., Amara, R. R., and Pulendran, B. (2020). T cell-inducing vaccine durably prevents mucosal SHIV infection even with lower neutralizing antibody titers. *Nature Medicine*, 26(6):932–940.

Azodi, C. B., Tang, J., and Shiu, S.-H. (2020). Opening the black box: interpretable machine learning for geneticists. *Trends in Genetics*, 36(6):442–455.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, 10(7):e0130140.

Baden, L. R., El Sahly, H. M., Essink, B., Kotloff, K., Frey, S., Novak, R., Diemert, D., Spector, S. A., Rouphael, N., Creech, C. B., et al. (2021). Efficacy and safety of the mRNA-1273 SARS-CoV-2 vaccine. *New England Journal of Medicine*, 384(5):403–416.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831.

Banerjee, A., Santra, D., and Maiti, S. (2020). Energetics based epitope screening in SARS CoV-2 (COVID 19) spike glycoprotein by immuno-informatic analysis aiming to a suitable vaccine development. *bioRxiv*.

Baruah, V. and Bose, S. (2020). Immunoinformatics-aided identification of T cell and B cell epitopes in the surface glycoprotein of 2019-nCoV. *Journal of Medical Virology*.

Benjamini, O., Rokach, L., Itchaki, G., Braester, A., Shvidel, L., Goldschmidt, N., Shapira, S., Dally, N., Avigdor, A., Rahav, G., et al. (2022). Safety and efficacy of the BNT162b mRNA COVID-19 vaccine in patients with chronic lymphocytic leukemia. *Haematologica*, 107(3):625–634.

Bhattacharya, M., Sharma, A. R., Patra, P., Ghosh, P., Sharma, G., Patra, B. C., Lee, S.-S., and Chakraborty, C. (2020). Development of epitope-based peptide vaccine against novel coronavirus 2019 (SARS-COV-2): Immunoinformatics approach. *Journal of medical virology*, 92(6):618–631.

Bileschi, M. L., Belanger, D., Bryant, D. H., Sanderson, T., Carter, B., Sculley, D., Bateman, A., DePristo, M. A., and Colwell, L. J. (2022). Using deep learning to annotate the protein universe. *Nature Biotechnology*, pages 1–6.

Brendel, W. and Bethge, M. (2019). Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. In *International Conference on Learning Representations*.

Bui, H.-H., Sidney, J., Dinh, K., Southwood, S., Newman, M. J., and Sette, A. (2006). Predicting population coverage of T-cell epitope-based diagnostics and vaccines. *BMC bioinformatics*, 7(1):153.

Buus, S., Lauemøller, S., Worning, P., Kesmir, C., Frimurer, T., Corbet, S., Fomsgaard, A., Hilden, J., Holm, A., and Brunak, S. (2003). Sensitive quantitative predictions of peptide-MHC binding by a 'Query by Committee' artificial neural network approach. *Tissue Antigens*, 62(5):378–384.

Carlini, N. and Wagner, D. (2017a). Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*.

Carlini, N. and Wagner, D. (2017b). Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*.

Carter, B., Bileschi, M., Smith, J., Sanderson, T., Bryant, D., Belanger, D., and Colwell, L. J. (2020). Critiquing protein family classification models using sufficient input subsets. *Journal of Computational Biology*, 27(8):1219–1231.

Carter, B., Huang, P., Liu, G., Liang, Y., Lin, P. J. C., Peng, B.-H., McKay, L. G. A., Dimitrakakis, A., Hsu, J., Tat, V., Saenkham-Huntsinger, P., Chen, J., Kaseke, C., Gaiha, G. D., Xu, Q., Griffiths, A., Tam, Y. K., Tseng, C.-T. K., and Gifford, D. K. (2023). A pan-variant mRNA-LNP T cell vaccine protects HLA transgenic mice from mortality after infection with SARS-CoV-2 Beta. *Frontiers in Immunology*, 14:1135815.

Carter, B., Jain, S., Mueller, J. W., and Gifford, D. (2021). Overinterpretation reveals image classification model pathologies. *Advances in Neural Information Processing Systems*, 34:15395–15407.

Carter, B., Mueller, J., Jain, S., and Gifford, D. (2019). What made you do this? Understanding black-box decisions with sufficient input subsets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 567–576.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2018). Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

Chicz, R. M., Urban, R. G., Lane, W. S., Gorga, J. C., Stern, L. J., Vignali, D. A., and Strominger, J. L. (1992). Predominant naturally processed peptides bound to HLA-DR1 are derived from MHC-related molecules and are heterogeneous in size. *Nature*, 358(6389):764–768.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Cohen, A. A., van Doremalen, N., Greaney, A. J., Andersen, H., Sharma, A., Starr, T. N., Keeffe, J. R., Fan, C., Schulz, J. E., Gnanapragasam, P. N., et al. (2022).

Mosaic RBD nanoparticles protect against challenge by diverse sarbecoviruses in animal models. *Science*, 377(6606):eabq0839.

Cohen, K. W., Linderman, S. L., Moodie, Z., Czartoski, J., Lai, L., Mantus, G., Norwood, C., Nyhoff, L. E., Edara, V. V., Floyd, K., et al. (2021). Longitudinal analysis shows durable and broad immune memory after SARS-CoV-2 infection with persisting antibody responses and memory B and T cells. *Cell Reports Medicine*, 2(7):100354.

Consortium, E. P. et al. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57.

Consortium, U. (2019). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515.

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18.

Coutard, B., Valle, C., de Lamballerie, X., Canard, B., Seidah, N., and Decroly, E. (2020). The spike glycoprotein of the new coronavirus 2019-nCoV contains a furin-like cleavage site absent in CoV of the same clade. *Antiviral Research*, 176:104742.

Croft, N. P., Smith, S. A., Pickering, J., Sidney, J., Peters, B., Faridi, P., Witney, M. J., Sebastian, P., Flesch, I. E., Heading, S. L., et al. (2019). Most viral peptides displayed by class I MHC on infected cells are immunogenic. *Proceedings of the National Academy of Sciences*, 116(8):3112–3117.

Dabkowski, P. and Gal, Y. (2017). Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*.

Dai, Z. and Gifford, D. K. (2023). Constrained submodular optimization for vaccine design. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Das, A. and Kempe, D. (2008). Algorithms for subset selection in linear regression. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 45–54.

de Oliveira, P. M. N., Mendes-de Almeida, D. P., Porto, V. B. G., Cordeiro, C. C., Teixeira, G. V., Pedro, R. S., Takey, P. R. G., Lignani, L. K., Xavier, J. R., da Gama, V. C. D., et al. (2022). Vaccine-induced immune thrombotic thrombocytopenia after COVID-19 vaccination: Description of a series of 39 cases in Brazil. *Vaccine*, 40(33):4788–4795.

de Silva, T. I., Liu, G., Lindsey, B. B., Dong, D., Moore, S. C., Hsu, N. S., Shah, D., Wellington, D., Mentzer, A. J., Angyal, A., et al. (2021). The impact of viral mutations on recognition by SARS-CoV-2 specific T cells. *iScience*, 24(11):103353.

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., and Das, P. (2018). Explanations based on the missing: towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*.

Dimitrakakis, A. (2021). Refinement of the computational vaccine optimization framework (OptiVax) through the development and analysis of a better algorithm for vaccine design choice. Master's thesis, Massachusetts Institute of Technology.

Dobson, C. S., Reich, A. N., Gaglione, S., Smith, B. E., Kim, E. J., Dong, J., Ronsard, L., Okonkwo, V., Lingwood, D., Dougan, M., et al. (2022). Antigen identification and high-throughput interaction mapping by reprogramming viral entry. *Nature Methods*, 19(4):449–460.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Elbe, S. and Buckland-Merrett, G. (2017). Data, disease and diplomacy: GISAID's innovative contribution to global health. *Global Challenges*, 1(1):33–46.

Ellis, J. M., Henson, V., Slack, R., Ng, J., Hartzman, R. J., and Hurley, C. K. (2000). Frequencies of HLA-A2 alleles in five US population groups: Predominance of A*02011 and identification of HLA-A*0231. *Human Immunology*, 61(3):334–340.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Excoffier, L. and Slatkin, M. (1995). Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–927.

Fast, E., Altman, R. B., and Chen, B. (2020). Potential T-cell and B-cell epitopes of 2019-nCoV. *bioRxiv*.

Feng, S., Wallace, E., Grissom II, A., Iyyer, M., Rodriguez, P., and Boyd-Graber, J. (2018). Pathologies of neural models make interpretations difficult. In *Empirical Methods in Natural Language Processing*.

Finkel, Y., Mizrahi, O., Nachshon, A., Weingarten-Gabbay, S., Yahalom-Ronen, Y., Tamir, H., Achdout, H., Melamed, S., Weiss, S., Isrealy, T., et al. (2020). The coding capacity of SARS-CoV-2. *bioRxiv*.

Fong, R., Patrick, M., and Vedaldi, A. (2019). Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2950–2958.

Fong, R. C. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Frosst, N. and Hinton, G. (2017). Distilling a neural network into a soft decision tree. In *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML*.

Garcia-Beltran, W. F., Lam, E. C., Denis, K. S., Nitido, A. D., Garcia, Z. H., Hauser, B. M., Feldman, J., Pavlovic, M. N., Gregory, D. J., Poznansky, M. C., et al. (2021). Multiple SARS-CoV-2 variants escape neutralization by vaccine-induced humoral immunity. *Cell*, 184(9):2372–2383.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2017). Texture and art with deep neural networks. *Current Opinion in Neurobiology*, 46:178–186.

Geers, D., Shamier, M. C., Bogers, S., den Hartog, G., Gommers, L., Nieuwkoop, N. N., Schmitz, K. S., Rijsbergen, L. C., van Osch, J. A., Dijkhuizen, E., et al. (2021). SARS-CoV-2 variants of concern partially escape humoral but not T cell responses in COVID-19 convalescent donors and vaccine recipients. *Science Immunology*, 6(59):eabj1750.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*.

Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. (2019). Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*.

Goh, K.-S., Chang, E., and Cheng, K.-T. (2001). SVM binary classifier ensembles for image classification. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 395–402. ACM.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Greener, J. G., Kandathil, S. M., Moffat, L., and Jones, D. T. (2022). A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, 23(1):40–55.

Grifoni, A., Sidney, J., Zhang, Y., Scheuermann, R. H., Peters, B., and Sette, A. (2020a). A sequence homology and bioinformatic approach can predict candidate targets for immune responses to SARS-CoV-2. *Cell Host & Microbe*, 27(4):671–680.

Grifoni, A., Weiskopf, D., Ramirez, S. I., Mateus, J., Dan, J. M., Moderbacher, C. R., Rawlings, S. A., Sutherland, A., Premkumar, L., Jadi, R. S., et al. (2020b). Targets of T cell responses to SARS-CoV-2 coronavirus in humans with COVID-19 disease and unexposed individuals. *Cell*, 181(7):1489–1501.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*.

Gupta, E., Mishra, R. K., and Niraj, R. R. K. (2020). Identification of potential vaccine candidates against SARS-CoV-2, a step forward to fight novel coronavirus 2019-nCoV: A reverse vaccinology approach. *bioRxiv*.

Gupta, R., Jung, E., and Brunak, S. (2004). Prediction of N-glycosylation sites in human proteins. *In preparation*.

Hadfield, J., Megill, C., Bell, S. M., Huddleston, J., Potter, B., Callender, C., Sagulenko, P., Bedford, T., and Neher, R. A. (2018). Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, 34(23):4121–4123.

Hajnik, R. L., Plante, J. A., Liang, Y., Alameh, M.-G., Tang, J., Bonam, S. R., Zhong, C., Adam, A., Scharton, D., Rafael, G. H., et al. (2022). Dual spike and nucleocapsid mRNA vaccination confer protection against SARS-CoV-2 Omicron and Delta variants in preclinical models. *Science Translational Medicine*, 14(662):eabq1945.

Halfmann, P. J., Iida, S., Iwatsuki-Horimoto, K., Maemura, T., Kiso, M., Scheaffer, S. M., Darling, T. L., Joshi, A., Loeber, S., Singh, G., et al. (2022). SARS-CoV-2 Omicron virus causes attenuated disease in mice and hamsters. *Nature*, 603(7902):687–692.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European Conference on Computer Vision*. Springer.

Heitmann, J. S., Bilich, T., Tandler, C., Nelde, A., Maringer, Y., Marconato, M., Reusch, J., Jäger, S., Denk, M., Richter, M., et al. (2022). A COVID-19 peptide vaccine for the induction of SARS-CoV-2 T cell immunity. *Nature*, 601(7894):617–622.

Helmberg, W., Dunivin, R., and Feolo, M. (2004). The sequencing-based typing tool of dbMHC: typing highly polymorphic gene sequences. *Nucleic Acids Research*, 32(suppl_2):W173–W175.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Herst, C. V., Burkholz, S., Sidney, J., Sette, A., Harris, P. E., Massey, S., Brasel, T., Cunha-Neto, E., Rosa, D. S., Chao, W. C. H., et al. (2020). An effective CTL peptide vaccine for ebola zaire based on survivors' CD8+ targeting of a particular nucleocapsid protein epitope with potential implications for COVID-19 vaccine design. *Vaccine*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hong, S. R., Hullman, J., and Bertini, E. (2020). Human factors in model interpretability: Industry practices, challenges, and needs. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–26.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. (2019). A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*.

Hu, Z., Ott, P. A., and Wu, C. J. (2018). Towards personalized, tumour-specific, therapeutic vaccines for cancer. *Nature Reviews Immunology*, 18(3):168.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*.

Ismail, S., Ahmad, S., and Azam, S. S. (2020). Immuno-informatics characterization SARS-CoV-2 spike glycoprotein for prioritization of epitope based multivalent peptide vaccine. *bioRxiv*.

Israelow, B., Mao, T., Klein, J., Song, E., Menasche, B., Omer, S. B., and Iwasaki, A. (2021). Adaptive immune determinants of viral clearance and protection in mouse models of SARS-CoV-2. *Science Immunology*, 6(64):eabl4509.

Jensen, K. K., Andreatta, M., Marcatili, P., Buus, S., Greenbaum, J. A., Yan, Z., Sette, A., Peters, B., and Nielsen, M. (2018). Improved methods for predicting peptide binding affinity to MHC class II molecules. *Immunology*, 154(3):394–406.

Ju, C., Bibaut, A., and van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818.

Jurtz, V., Paul, S., Andreatta, M., Marcatili, P., Peters, B., and Nielsen, M. (2017). NetMHCpan-4.0: improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. *Journal of Immunology*, 199(9):3360–3368.

Kared, H., Redd, A. D., Bloch, E. M., Bonny, T. S., Sumatoh, H., Kairi, F., Carbajo, D., Abel, B., Newell, E. W., Bettinotti, M. P., et al. (2021). SARS-CoV-2–specific CD8+ t cell responses in convalescent COVID-19 individuals. *Journal of Clinical Investigation*, 131(5):e145476.

Karpathy, A. (2011). Lessons learned from manually classifying CIFAR-10. `http://karpathy.github.io/2011/04/27/manually-classifying-cifar10`.

Kenter, G. G., Welters, M. J., Valentijn, A. R. P., Lowik, M. J., Berends-van der Meer, D. M., Vloon, A. P., Essahsah, F., Fathers, L. M., Offringa, R., Drijfhout, J. W., et al. (2009). Vaccination against HPV-16 oncoproteins for vulvar intraepithelial neoplasia. *New England Journal of Medicine*, 361(19):1838–1847.

Khan, A., Alam, A., Imam, N., Siddiqui, M. F., and Ishrat, R. (2020). Design of an epitope-based peptide vaccine against the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2): A vaccine informatics approach. *bioRxiv*.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*.

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. (2019). The (un)reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer.

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. (2018). Learning how to explain neural networks: PatternNet and PatternAttribution. In *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kingstad-Bakke, B., Lee, W., Chandrasekar, S. S., Gasper, D. J., Salas-Quinchucua, C., Cleven, T., Sullivan, J. A., Talaat, A., Osorio, J. E., and Suresh, M. (2022).

Vaccine-induced systemic and mucosal T cell immunity to SARS-CoV-2 viral variants. *Proceedings of the National Academy of Sciences*, 119(20):e2118312119.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. (2018). Human decisions and machine predictions. *The Quarterly Journal of Economics*, 133(1):237–293.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

Kotturi, M. F., Assarsson, E., Peters, B., Grey, H., Oseroff, C., Pasquetto, V., and Sette, A. (2009). Of mice and humans: how good are HLA transgenic mice as a model of human immune responses? *Immunome Research*, 5:3.

Kreiter, S., Selmi, A., Diken, M., Sebastian, M., Osterloh, P., Schild, H., Huber, C., Türeci, Ö., and Sahin, U. (2008). Increased antigen presentation efficiency by coupling antigens to MHC class I trafficking signals. *Journal of Immunology*, 180(1):309–318.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kumari, P., Rothan, H. A., Natekar, J. P., Stone, S., Pathak, H., Strate, P. G., Arora, K., Brinton, M. A., and Kumar, M. (2021). Neuroinvasion and encephalitis following intranasal inoculation of SARS-CoV-2 in K18-hACE2 mice. *Viruses*, 13(1):132.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1–8.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, C. H.-J. and Koohy, H. (2020). In silico identification of vaccine targets for 2019-nCoV. *F1000Research*, 9.

Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In *Empirical Methods in Natural Language Processing*.

Li, J., Monroe, W., and Jurafsky, D. (2017). Understanding neural networks through representation erasure. *arXiv:1612.08220*.

Li, W., Joshi, M. D., Singhania, S., Ramsey, K. H., and Murthy, A. K. (2014). Peptide vaccine: progress and challenges. *Vaccines*, 2(3):515–536.

Liebenwein, L., Baykal, C., Carter, B., Gifford, D., and Rus, D. (2021). Lost in pruning: The effects of pruning neural networks beyond test accuracy. *Proceedings of Machine Learning and Systems*, 3:93–138.

Lipton, Z. C. (2016). The mythos of model interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*.

Liu, G., Carter, B., Bricken, T., Jain, S., Viard, M., Carrington, M., and Gifford, D. K. (2020a). Computationally optimized SARS-CoV-2 MHC class I and II vaccine formulations predicted to target human haplotype distributions. *Cell Systems*, 11(2):131–144.

Liu, G., Carter, B., and Gifford, D. K. (2021). Predicted cellular immunity population coverage gaps for SARS-CoV-2 subunit vaccines and their augmentation by compact peptide sets. *Cell Systems*, 12(1):102–107.

Liu, G., Dimitrakakis, A., Carter, B., and Gifford, D. (2022). Maximum n-times coverage for vaccine design. In *International Conference on Learning Representations*.

Liu, G., Zeng, H., Mueller, J., Carter, B., Wang, Z., Schilz, J., Horny, G., Birnbaum, M. E., Ewert, S., and Gifford, D. K. (2020b). Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics*, 36(7):2126–2133.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Martinez, D. R., Schäfer, A., Leist, S. R., De la Cruz, G., West, A., Atochina-Vasserman, E. N., Lindesmith, L. C., Pardi, N., Parks, R., Barr, M., et al. (2021). Chimeric spike mRNA vaccines protect against Sarbecovirus challenge in mice. *Science*, 373(6558):991–998.

Matchett, W. E., Joag, V., Stolley, J. M., Shepherd, F. K., Quarnstrom, C. F., Mickelson, C. K., Wijeyesinghe, S., Soerens, A. G., Becker, S., Thiede, J. M., et al. (2021). Cutting edge: nucleocapsid vaccine elicits spike-independent SARS-CoV-2 protective immunity. *Journal of Immunology*, 207(2):376–379.

Mathelier, A., Fornes, O., Arenillas, D. J., Chen, C.-y., Denay, G., Lee, J., Shi, W., Shyr, C., Tan, G., Worsley-Hunt, R., et al. (2016). JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 44(D1):D110–D115.

McAuley, J., Leskovec, J., and Jurafsky, D. (2012). Learning attitudes and attributes from multi-aspect reviews. In *IEEE International Conference on Data Mining*, pages 1020–1025.

Mitra, D., Shekhar, N., Pandey, J., Jain, A., and Swaroop, S. (2020). Multi-epitope based peptide vaccine design against SARS-CoV-2 using its spike protein. *bioRxiv*.

Moise, L., Gutierrez, A., Kibria, F., Martin, R., Tassone, R., Liu, R., Terry, F., Martin, B., and De Groot, A. S. (2015). iVAX: An integrated toolkit for the selection and optimization of antigens and the design of epitope-driven vaccines. *Human vaccines & immunotherapeutics*, 11(9):2312–2321.

Moss, P. (2022). The T cell immune response against SARS-CoV-2. *Nature Immunology*, 23(2):186–193.

Murdoch, W. J., Liu, P. J., and Yu, B. (2018). Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations*.

Nakiboneka, R., Mugaba, S., Auma, B. O., Kintu, C., Lindan, C., Nanteza, M. B., Kaleebu, P., and Serwanga, J. (2019). Interferon gamma (IFN-γ) negative CD4+ and CD8+ T-cells can produce immune mediators in response to viral antigens. *Vaccine*, 37(1):113–122.

Nardin, E. H., Oliveira, G. A., Calvo-Calle, J. M., Castro, Z. R., Nussenzweig, R. S., Schmeckpeper, B., Hall, B. F., Diggs, C., Bodison, S., and Edelman, R. (2000). Synthetic malaria peptide vaccine elicits high levels of antibodies in vaccinees of defined HLA genotypes. *Journal of Infectious Diseases*, 182(5):1486–1496.

Naseer, M. M., Ranasinghe, K., Khan, S. H., Hayat, M., Shahbaz Khan, F., and Yang, M.-H. (2021). Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308.

Nathan, A., Rossin, E. J., Kaseke, C., Park, R. J., Khatri, A., Koundakjian, D., Urbach, J. M., Singh, N. K., Bashirova, A., Tano-Menka, R., et al. (2021). Structure-guided T cell vaccine design for SARS-CoV-2 variants and sarbecoviruses. *Cell*, 184(17):4401–4413.

Nerli, S. and Sgourakis, N. G. (2020). Structure-based modeling of SARS-CoV-2 peptide/HLA-A02 antigens. *bioRxiv*.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436.

Nielsen, M., Lundegaard, C., Worning, P., Lauemøller, S. L., Lamberth, K., Buus, S., Brunak, S., and Lund, O. (2003). Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science*, 12(5):1007–1017.

Niven, T. and Kao, H.-Y. (2019). Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

O'Donnell, T., Rubinsteyn, A., and Laserson, U. (2020a). A model of antigen processing improves prediction of MHC I-presented peptides. *bioRxiv*.

O'Donnell, T. J., Rubinsteyn, A., Bonsack, M., Riemer, A. B., Laserson, U., and Hammerbacher, J. (2018). MHCflurry: open-source class I MHC binding affinity prediction. *Cell Systems*, 7(1):129–132.

O'Donnell, T. J., Rubinsteyn, A., and Laserson, U. (2020b). MHCflurry 2.0: Improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. *Cell Systems*, 11(1):42–48.

Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization. *Distill*.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*.

Papernot, N., Carlini, N., Goodfellow, I., Feinman, R., Faghri, F., Matyasko, A., Hambardzumyan, K., Juang, Y.-L., Kurakin, A., Sheatsley, R., Garg, A., and Lin, Y.-C. (2017). cleverhans v2.0.0: an adversarial machine learning library. *arXiv:1610.00768*.

Pardi, N., Hogan, M. J., Pelc, R. S., Muramatsu, H., Andersen, H., DeMaso, C. R., Dowd, K. A., Sutherland, L. L., Scearce, R. M., Parks, R., et al. (2017). Zika virus protection by a single low-dose nucleoside-modified mRNA vaccination. *Nature*, 543(7644):248–251.

Pardi, N., Hogan, M. J., Porter, F. W., and Weissman, D. (2018). mRNA vaccines — a new era in vaccinology. *Nature Reviews Drug Discovery*, 17(4):261–279.

Pardieck, I. N., van der Sluis, T. C., van der Gracht, E. T., Veerkamp, D. M., Behr, F. M., van Duikeren, S., Beyrend, G., Rip, J., Nadafi, R., Beyranvand Nejad, E., et al. (2022). A third vaccination with a single T cell epitope confers protection in a murine model of SARS-CoV-2 infection. *Nature Communications*, 13(1):3966.

Paston, S. J., Brentville, V. A., Symonds, P., and Durrant, L. G. (2021). Cancer vaccines, adjuvants, and delivery systems. *Frontiers in Immunology*, 12:627932.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*.

Patel, N. V. (2017). *Why Doctors Aren't Afraid of Better, More Efficient AI Diagnosing Cancer*. Accessed September 27, 2020.

Pavord, S., Scully, M., Hunt, B. J., Lester, W., Bagot, C., Craven, B., Rampotas, A., Ambler, G., and Makris, M. (2021). Clinical features of vaccine-induced immune thrombocytopenia and thrombosis. *New England Journal of Medicine*, 385(18):1680–1689.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peters, B., Nielsen, M., and Sette, A. (2020). T cell epitope predictions. *Annual Review of Immunology*, 38(1):123–145.

Poran, A., Harjanto, D., Malloy, M., Rooney, M. S., Srinivasan, L., and Gaynor, R. B. (2020). Sequence-based prediction of vaccine targets for inducing T cell responses to SARS-CoV-2 utilizing the bioinformatics predictor RECON. *bioRxiv*.

Prachar, M., Justesen, S., Steen-Jensen, D. B., Thorgrimsen, S. P., Jurgons, E., Winther, O., and Bagger, F. O. (2020). COVID-19 vaccine candidates: Prediction and validation of 174 SARS-CoV-2 epitopes. *bioRxiv*.

Radford, A., Jozefowicz, R., and Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *arXiv:1704.01444*.

Rajan, S., Khunti, K., Alwan, N., Steves, C., MacDermott, N., Morsella, A., Angulo, E., Winkelmann, J., Bryndová, L., Fronteira, I., et al. (2021). In the wake of the pandemic: Preparing for Long COVID. World Health Organization (WHO) Policy Brief 39.

Ramaiah, A. and Arumugaswami, V. (2020). Insights into cross-species evolution of novel human coronavirus 2019-nCoV and defining immune determinants for vaccine development. *bioRxiv*.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2018). Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv:1806.00451*.

Redd, A. D., Nardin, A., Kared, H., Bloch, E. M., Pekosz, A., Laeyendecker, O., Abel, B., Fehlings, M., Quinn, T. C., and Tobian, A. A. (2021). CD8+ T-cell responses in COVID-19 convalescent individuals target conserved epitopes from multiple prominent SARS-CoV-2 circulating variants. *Open Forum Infectious Diseases*, 8(7):ofab143.

Reynisson, B., Alvarez, B., Paul, S., Peters, B., and Nielsen, M. (2020a). NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Research*, 48(W1):W449–W454.

Reynisson, B., Barra, C., Kaabinejadian, S., Hildebrand, W. H., Peters, B., and Nielsen, M. (2020b). Improved prediction of MHC II antigen presentation through integration and motif deconvolution of mass spectrometry MHC eluted ligand data. *Journal of Proteome Research*, 19(6):2304–2315.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Riley, T. P., Keller, G. L., Smith, A., Devlin, J. R., Davancaze, L. M., Arbuiso, A. A., and Baker, B. M. (2019). Structure based prediction of neoantigen immunogenicity. *Frontiers in Immunology*, 10:2047.

Rist, M. J., Theodossis, A., Croft, N. P., Neller, M. A., Welland, A., Chen, Z., Sullivan, L. C., Burrows, J. M., Miles, J. J., Brennan, R. M., et al. (2013). HLA peptide length preferences control CD8+ T cell responses. *Journal of Immunology*, 191(2):561–571.

Rizzo, M. L. and Székely, G. J. (2016). Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38.

Rosenfeld, A., Zemel, R., and Tsotsos, J. K. (2018). The elephant in the room. *arXiv preprint arXiv:1808.03305*.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. (2017). Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2662–2670.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Saha, R. and Prasad, B. V. (2020). In silico approach for designing of a multi-epitope based vaccine against novel Coronavirus (SARS-COV-2). *bioRxiv*.

Sahin, U., Derhovanessian, E., Miller, M., Kloke, B.-P., Simon, P., Löwer, M., Bukur, V., Tadmor, A. D., Luxemburger, U., Schrörs, B., et al. (2017). Personalized RNA mutanome vaccines mobilize poly-specific therapeutic immunity against cancer. *Nature*, 547(7662):222–226.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. (2016). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673.

Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems*.

Schultheiß, C., Paschold, L., Simnica, D., Mohme, M., Willscher, E., von Wenserski, L., Scholz, R., Wieters, I., Dahlke, C., Tolosa, E., et al. (2020). Next-generation sequencing of T and B cell receptor repertoires from COVID-19 patients showed signatures associated with severity of disease. *Immunity*, 53(2):442–455.

Seabold, S. and Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference*, pages 92–96.

Sekine, T., Perez-Potti, A., Rivera-Ballesteros, O., Strålin, K., Gorin, J.-B., Olsson, A., Llewellyn-Lacey, S., Kamal, H., Bogdanovic, G., Muschiol, S., et al. (2020). Robust T cell immunity in convalescent individuals with asymptomatic or mild COVID-19. *Cell*, 183:158–168.

Sette, A., Vitiello, A., Reherman, B., Fowler, P., Nayersina, R., Kast, W. M., Melief, C., Oseroff, C., Yuan, L., Ruppert, J., et al. (1994). The relationship between class I binding affinity and immunogenicity of potential cytotoxic T cell epitopes. *Journal of Immunology*, 153(12):5586–5592.

Sha, Y. and Wang, M. D. (2017). Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*.

Shetty, R., Schiele, B., and Fritz, M. (2019). Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8226.

Shree, T. (2022). Can B cell-deficient patients rely on COVID-19 vaccine-induced T-cell immunity? *British Journal of Haematology*, 197(6):659–661.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International Conference on Machine Learning*.

Shuai, H., Chan, J. F.-W., Yuen, T. T.-T., Yoon, C., Hu, J.-C., Wen, L., Hu, B., Yang, D., Wang, Y., Hou, Y., et al. (2021). Emerging SARS-CoV-2 variants expand species tropism to murines. *EBioMedicine*, 73:103643.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations*.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Simpson, B., Dutil, F., Bengio, Y., and Cohen, J. P. (2019). GradMask: Reduce overfitting by regularizing saliency. *arXiv preprint arXiv:1904.07478*.

Singh, A., Thakur, M., Sharma, L. K., and Chandra, K. (2020a). Designing a multi-epitope peptide-based vaccine against SARS-CoV-2. *bioRxiv*.

Singh, K. K., Mahajan, D., Grauman, K., Lee, Y. J., Feiszli, M., and Ghadiyaram, D. (2020b). Don't judge an object by its context: Learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11070–11078.

Sirignano, J. A., Sadhwani, A., and Giesecke, K. (2018). Deep learning for mortgage risk. *arXiv:1607.02470*.

Smola, A. J., Vishwanathan, S., and Hofmann, T. (2005). Kernel methods for missing variables. In *Artificial Intelligence and Statistics*.

Snyder, T. M., Gittelman, R. M., Klinger, M., May, D. H., Osborne, E. J., Taniguchi, R., Zahid, H. J., Kaplan, I. M., Dines, J. N., Noakes, M. N., et al. (2020). Magnitude and dynamics of the T-Cell response to SARS-CoV-2 infection at both individual and population levels. *medRxiv*.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015). Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Srivastava, S., Verma, S., Kamthania, M., Kaur, R., Badyal, R. K., Saxena, A. K., Shin, H.-J., Kolbe, M., and Pandey, K. (2020). Structural basis to design multi-epitope vaccines against Novel Coronavirus 19 (COVID19) infection, the ongoing pandemic emergency: an in silico approach. *bioRxiv*.

Stirling, D. R., Swain-Bowden, M. J., Lucas, A. M., Carpenter, A. E., Cimini, B. A., and Goodman, A. (2021). CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics*, 22:1–11.

Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackermann, Z., et al. (2020). A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.

Strobelt, H., Gehrmann, S., Pfister, H., and Rush, A. (2018). LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, pages 667–676.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*.

Swank, Z., Senussi, Y., Manickas-Hill, Z., Yu, X. G., Li, J. Z., Alter, G., and Walt, D. R. (2023). Persistent circulating severe acute respiratory syndrome coronavirus 2 spike is associated with post-acute coronavirus disease 2019 sequelae. *Clinical Infectious Diseases*, 76(3):e487–e490.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Tahir ul Qamar, M., Rehman, A., Ashfaq, U. A., Awan, M. Q., Fatima, I., Shahid, F., and Chen, L.-L. (2020). Designing of a next generation multiepitope based vaccine (MEV) against SARS-COV-2: Immunoinformatics and in silico approaches. *bioRxiv*.

Tommasi, T., Patricia, N., Caputo, B., and Tuytelaars, T. (2017). A deeper look at dataset bias. In *Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer.

Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528. IEEE.

Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970.

Tramer, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In *USENIX Security Symposium*.

Tregoning, J. S., Flight, K. E., Higham, S. L., Wang, Z., and Pierce, B. F. (2021). Progress of the COVID-19 vaccine effort: viruses, vaccines and variants versus efficacy, effectiveness and escape. *Nature Reviews Immunology*, 21(10):626–636.

Trolle, T., McMurtrey, C. P., Sidney, J., Bardet, W., Osborn, S. C., Kaever, T., Sette, A., Hildebrand, W. H., Nielsen, M., and Peters, B. (2016). The length distribution of class I-restricted T cell epitopes is determined by both peptide supply and MHC allele-specific binding preference. *Journal of Immunology*, 196(4):1480–1487.

Tseng, C.-T., Sbrana, E., Iwata-Yoshikawa, N., Newman, P. C., Garron, T., Atmar, R. L., Peters, C. J., and Couch, R. B. (2012). Immunization with SARS coronavirus vaccines leads to pulmonary immunopathology on challenge with the SARS virus. *PLoS One*, 7(4):e35421.

Tseng, C.-T. K., Huang, C., Newman, P., Wang, N., Narayanan, K., Watts, D. M., Makino, S., Packard, M. M., Zaki, S. R., Chan, T.-s., et al. (2007). Severe acute respiratory syndrome coronavirus infection of mice transgenic for the human Angiotensin-converting enzyme 2 virus receptor. *Journal of Virology*, 81(3):1162–1173.

Vashi, Y., Jagrit, V., and Kumar, S. (2020). Understanding the B and T cells epitopes of spike protein of severe respiratory syndrome coronavirus-2: A computational way to predict the immunogens. *bioRxiv*.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Vita, R., Mahajan, S., Overton, J. A., Dhanda, S. K., Martini, S., Cantrell, J. R., Wheeler, D. K., Sette, A., and Peters, B. (2019). The immune epitope database (IEDB): 2018 update. *Nucleic Acids Research*, 47(D1):D339–D343.

Viviano, J. D., Simpson, B., Dutil, F., Bengio, Y., and Cohen, J. P. (2021). Saliency is a possible red herring when diagnosing poor generalization. In *International Conference on Learning Representations*.

Walls, A. C., Park, Y.-J., Tortorici, M. A., Wall, A., McGuire, A. T., and Veesler, D. (2020). Structure, function, and antigenicity of the SARS-CoV-2 spike glycoprotein. *Cell*, 181(2):281–292.

Walsh, E. E., Frenck Jr, R. W., Falsey, A. R., Kitchin, N., Absalon, J., Gurtman, A., Lockhart, S., Neuzil, K., Mulligan, M. J., Bailey, R., et al. (2020). Safety and immunogenicity of two RNA-based Covid-19 vaccine candidates. *New England Journal of Medicine*, 383(25):2439–2450.

Wang, Q., Qiu, Y., Li, J.-Y., Zhou, Z.-J., Liao, C.-H., and Ge, X.-Y. (2020). A unique protease cleavage site predicted in the spike protein of the novel pneumonia coronavirus (2019-ncov) potentially related to viral transmissibility. *Virologica Sinica*, pages 1–3.

Wang, Y., Huang, M., Zhao, L., et al. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Wibmer, C. K., Ayres, F., Hermanus, T., Madzivhandila, M., Kgagudi, P., Oosthuysen, B., Lambson, B. E., de Oliveira, T., Vermeulen, M., van der Berg, K., Rossouw, T., Boswell, M., Ueckermann, V., Meiring, S., von Gottberg, A., Cohen, C., Morris, L., Bhiman, J. N., and Moore, P. L. (2021). SARS-CoV-2 501Y.V2 escapes neutralization by South African COVID-19 donor plasma. *Nature Medicine*, 27(4):622–625.

Willett, B. J., Grove, J., MacLean, O. A., Wilkie, C., De Lorenzo, G., Furnon, W., Cantoni, D., Scott, S., Logan, N., Ashraf, S., et al. (2022). SARS-CoV-2 Omicron is an immune escape variant with an altered cell entry pathway. *Nature Microbiology*, 7(8):1161–1179.

Wolfert, M. A. and Boons, G.-J. (2013). Adaptive immune activation: glycosylation does matter. *Nature Chemical Biology*, 9(12):776–784.

Wrapp, D., Wang, N., Corbett, K. S., Goldsmith, J. A., Hsieh, C.-L., Abiona, O., Graham, B. S., and McLellan, J. S. (2020). Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science*, 367(6483):1260–1263.

Wu, M., Hughes, M. C., Parbhoo, S., Zazzi, M., Roth, V., and Doshi-Velez, F. (2017). Beyond sparsity: Tree regularization of deep models for interpretability. In *NIPS TIML Workshop*.

Yarmarkovich, M., Warrington, J. M., Farrel, A., and Maris, J. M. (2020). Identification of SARS-CoV-2 vaccine epitopes predicted to induce long-term population-scale immunity. *Cell Reports Medicine*.

Yazdani, Z., Rafiei, A., Yazdani, M., and Valadan, R. (2020). Design an efficient multi-epitope peptide vaccine candidate against SARS-CoV-2: An in silico analysis. *bioRxiv*.

Yoshikawa, N., Yoshikawa, T., Hill, T., Huang, C., Watts, D. M., Makino, S., Milligan, G., Chan, T., Peters, C. J., and Tseng, C.-T. K. (2009). Differential virological and immunological outcome of severe acute respiratory syndrome coronavirus infection in susceptible and resistant transgenic mice expressing human angiotensin-converting enzyme 2. *Journal of Virology*, 83(11):5451–5465.

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Medicine*, 15(11):e1002683.

Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *arXiv:1212.5701*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*.

Zeng, H., Edwards, M. D., Liu, G., and Gifford, D. K. (2016). Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics*, 32(12):i121.

Zeng, H. and Gifford, D. K. (2019). Quantification of uncertainty in peptide-MHC binding prediction improves high-affinity peptide selection for therapeutic design. *Cell Systems*, 9(2):159–166.

Zhang, L., Lin, D., Sun, X., Curth, U., Drosten, C., Sauerhering, L., Becker, S., Rox, K., and Hilgenfeld, R. (2020a). Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Science*, 368(6489):409–412.

Zhang, Q., Yang, Y., Wu, Y. N., and Zhu, S. (2018). Interpreting CNNs via decision trees. *arXiv:1802.00121*.

Zhang, S., Zhang, H., and Zhao, J. (2009). The role of CD4 T cell help for CD8 CTL activation. *Biochemical and Biophysical Research Communications*, 384(4):405–408.

Zhang, X., Wu, S., Wu, B., Yang, Q., Chen, A., Li, Y., Zhang, Y., Pan, T., Zhang, H., and He, X. (2021). SARS-CoV-2 Omicron strain exhibits potent capabilities for immune evasion and viral entrance. *Signal Transduction and Targeted Therapy*, 6(1):430.

Zhang, Y., Zhao, W., Mao, Y., Wang, S., Zhong, Y., Su, T., Gong, M., Lu, X., Cheng, J., and Yang, H. (2020b). Site-specific N-glycosylation characterization of recombinant SARS-CoV-2 spike proteins using high-resolution mass spectrometry. *bioRxiv*.

Zhao, J., Zhao, J., Mangalam, A. K., Channappanavar, R., Fett, C., Meyerholz, D. K., Agnihothram, S., Baric, R. S., David, C. S., and Perlman, S. (2016). Airway memory CD4+ T cells mediate protective immunity against emerging respiratory coronaviruses. *Immunity*, 44(6):1379–1391.