Query lower bounds for log-concave sampling

Citation: Chewi, Sinho, de Dios Pont, Jaume, Li, Jerry, Lu, Chen and Narayanan, Shyam. 2024. "Query lower bounds for log-concave sampling." Journal of the ACM.

As Published: 10.1145/3673651

Publisher: Association for Computing Machinery

Persistent URL: https://hdl.handle.net/1721.1/155703

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Query lower bounds for log-concave sampling

SINHO CHEWI, Institute for Advanced Study, Princeton, United States
JAUME DE DIOS PONT, ETH Zürich, Zürich, Switzerland
JERRY LI, Microsoft Research, Redmond, United States
CHEN LU, Massachusetts Institute of Technology, Cambridge, United States
SHYAM NARAYANAN, Massachusetts Institute of Technology, Cambridge, United States

Log-concave sampling has witnessed remarkable algorithmic advances in recent years, but the corresponding problem of proving *lower bounds* for this task has remained elusive, with lower bounds previously known only in dimension one. In this work, we establish the following query lower bounds: (1) sampling from strongly log-concave and log-smooth distributions in dimension $d \geq 2$ requires $\Omega(\log \kappa)$ queries, which is sharp in any constant dimension, and (2) sampling from Gaussians in dimension d (hence also from general log-concave and log-smooth distributions in dimension d) requires $\widetilde{\Omega}(\min(\sqrt{\kappa} \log d, d))$ queries, which is nearly sharp for the class of Gaussians. Here κ denotes the condition number of the target distribution. Our proofs rely upon (1) a multiscale construction inspired by work on the Kakeya conjecture in geometric measure theory, and (2) a novel reduction that demonstrates that block Krylov algorithms are optimal for this problem, as well as connections to lower bound techniques based on Wishart matrices developed in the matrix-vector query literature.

CCS Concepts: • **Theory of computation → Random walks and Markov chains**; **Design and analysis of algorithms**.

Additional Key Words and Phrases: block Krylov, log-concave sampling, matrix-vector queries

1 Introduction

We study the problem of sampling from a target distribution on $\mathbb{R}^d$ given query access to its unnormalized density. This is a fundamental algorithmic primitive arising in diverse fields, such as Bayesian inference, numerical simulation, and randomized algorithms [Robert and Casella 2004]. Recently, there has been considerable progress in developing faster algorithms for this problem, particularly in the case where the target distribution is log-concave. In large part, these results have been achieved by exploiting the rich interplay between optimization and sampling [Jordan et al. 1998; Wibisono 2018], leading to novel sampling schemes inpsired by classical optimization methods [Bernton 2018; Chewi et al. 2020; Lee et al. 2021b; Ma et al. 2021; Zhang et al. 2020], as well as new quantitative convergence guarantees for sampling [Dalalyan 2017; Durmus et al. 2019].

In light of such results, many prior works (e.g., [Chatterji et al. 2022; Cheng et al. 2018; Lee et al. 2021a]) have raised the foundational question of whether the algorithmic upper bounds are tight. However, there is still a dearth of lower bounds for log-concave sampling. This lies in stark contrast to the analogous setting of

The results of this work were first announced in an extended abstract at FOCS 2023.

Authors' Contact Information: Sinho Chewi, Institute for Advanced Study, Princeton, New Jersey, United States; e-mail: schewi@ias.edu; Jaume de Dios Pont, ETH Zürich, Zürich, Switzerland; e-mail: jdedios@math.ucla.edu; Jerry Li, Microsoft Research, Redmond, Washington, United States; e-mail: jerrl@microsoft.com; Chen Lu, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States; e-mail: chenl819@mit.edu; Shyam Narayanan, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States; e-mail: shyamsn@mit.edu.

ACM 1557-735X/2024/6-ART
https://doi.org/10.1145/3673651

convex optimization, in which the query complexity has been tightly characterized for a plethora of function classes [Nemirovskij and Yudin 1983; Nesterov 2018]. Such lower bounds yield important insights into the limitations of our existing algorithms and provide guidance towards identifying optimal ones.

Given the deep connections between the two fields, it is natural to ask why optimization lower bounds cannot be converted into sampling lower bounds. One way to do so is to directly reduce from optimization, as was done in [Gopi et al. 2022]. However, as we are interested in the intrinsic complexity of sampling, we make the standard assumption that the mode of the target distribution to zero to remove the optimization component of the sampling task, which rules out this approach. Another avenue is to borrow the techniques used for optimization lower bounds, but there are several obstructions to doing so. First, most optimization lower bounds hold against (classes of) deterministic algorithms and proceed by constructing specific adversarial functions [Bubeck 2015; Nesterov 2018]. In contrast, lower bounds for randomized algorithms are relatively recent and still not fully understood [Woodworth and Srebro 2017], which poses a major challenge for sampling algorithms, since they are inherently randomized. Second, whereas optimization constructions can employ local perturbations to hide the minima, sampling constructions need to hide the bulk of the mass of the target distribution, making them surprisingly delicate.

We now describe the problem in more detail. We consider the canonical setting in which target distribution π on $\mathbb{R}^d$ is α-strongly log-concave and β-log-smooth, with its mode located at the origin. Namely, we assume $\pi \propto \exp(-V)$, where the potential $V : \mathbb{R}^d \to \mathbb{R}$ is twice continuously differentiable, α-strongly convex, β-smooth, and $\nabla V(0) = 0$. We let $\kappa := \beta/\alpha$ denote the *condition number* of π. We study algorithms in which the sampler is given query access to V and ∇V, and the goal is to produce a sample whose law is close to π in total variation distance. The complexity of the algorithm is measured by the number of queries made. Note that this oracle model captures the majority of sampling algorithms used in practice, including the unadjusted Langevin algorithm, Hamiltonian Monte Carlo, Metropolized random walks, and hit-and-run.

Despite the intense research activity centered on log-concave sampling, only a handful of works address the lower bound question, and the majority of them are either algorithm-specific or pertain to auxiliary problems such as estimation of the normalizing constant; see Section 1.2 for related work. To the best of our knowledge, currently the only general log-concave sampling lower bound is that of [Chewi et al. 2022b], which establishes a sharp query lower bound of order $\Omega(\log\log\kappa)$ in dimension one. However, that work leaves open the question of obtaining stronger lower bounds in higher dimension, which is the more relevant case for applications. Even beyond the log-concave setting, we are aware of only one other work that obtains query lower bounds for sampling: the recent result of [Chewi et al. 2023] is incomparable to the present work, as it considers a different setting, and we discuss it further in Section 1.2. Overall, the lack of sampling lower bounds points to a lack of tools for addressing this problem and motivates the present work.

1.1 Our contributions

In this paper, we make significant progress on this problem by proving new lower bounds for sampling which reach beyond the one-dimensional setting considered in [Chewi et al. 2022b]. In fact, for some settings of interest, our lower bounds match existing upper bounds up to constants, and we therefore obtain some of the first *tight* complexity results for sampling from log-concave distributions in dimension $d > 1$. We obtain lower bounds in two regimes:

Lower bounds in low dimension. Our first lower bound gives a tight characterization of the complexity of log-sampling in any constant dimension $d \geq 2$. We show:

Theorem 1.1 (informal, see Theorem 3.1). *For any dimension $d \geq 2$, any sampler for d-dimensional log-concave distributions with condition number κ requires $\Omega(\log\kappa)$ queries.*

Note that this result is exponentially stronger than the $\Omega(\log\log\kappa)$ lower bound in the univariate case [Chewi et al. 2022b]. Moreover, when the dimension d is held fixed, we obtain a matching $O(\log\kappa)$ algorithmic upper bound, based on folklore ideas from the classical literature on sampling from convex bodies (Theorem A.4). Together with the result of [Chewi et al. 2022b] for $d = 1$, this settles the complexity of log-concave sampling in constant dimension.

On a technical level, the lower bound is based on a novel construction inspired by work on the Kakeya conjecture in harmonic analysis, which we believe may be of independent interest. We give a detailed description of the construction in Section 3.

Lower bounds in high dimension. Our second set of lower bounds applies to the high-dimensional setting and implies that when the dimension is sufficiently large, a polynomial dependence on the condition number κ is unavoidable (in contrast to Theorem 1.1, which only gives a logarithmic dependence on κ in low dimension). In fact, our lower bounds hold for the special case of sampling from Gaussians, for which they are nearly tight. We first prove the following theorem.

Theorem 1.2 (informal, see Corollary 4.4). *Any sampler for centered d-dimensional Gaussians with condition number κ requires $\Omega(\min(\sqrt{\kappa}, d))$ queries.*

We emphasize the fact that in our setting, the Gaussians are centered. Note that if the Gaussians were allowed to have varying means, then one can deduce a sampling lower bound by reducing the optimization task of minimizing a convex quadratic function $x \mapsto \langle (x-x_\star), \Sigma^{-1}(x-x_\star)\rangle$ to the task of sampling from the corresponding Gaussian $\mathcal{N}(x_\star, \Sigma)$. However, as previously alluded to, this does not address the inherent difficulty of the sampling problem.

The proof of Theorem 1.2 rests upon an elegant technique developed in the literature on the matrix-vector query model (see Section 1.2) in which the conditioning properties and sharp characterizations of the eigenvalue distribution of Wishart matrices are used to produce difficult lower bound instances for various tasks. We adapt this method to our context by reducing the task of inverse trace estimation to sampling (see Theorem 4.1).

As we show in Appendix B, the lower bound is nearly tight over the class of Gaussians, as it is possible to sample from a Gaussian using $O(\min(\sqrt{\kappa}\log d, d))$ queries using the block Krylov method. However, note that the lower bound from Theorem 1.2 does not match the block Krylov upper bound, and the lower bound of Theorem 1.2 is vacuous when κ is constant. In particular, it leaves open the possibility that the complexity of sampling from well-conditioned Gaussians is dimension-free. While such dimension-free rates are possible in convex optimization, our next result shows that the same is in fact not possible for log-concave sampling:

Theorem 1.3. (informal, see Theorem 5.22) *Let d be sufficiently large, and let $\kappa \le d^{1/5-\delta}$. Then, any sampler for d-dimensional Gaussians with condition number κ requires $\Omega_\delta(\sqrt{\kappa}\log d)$ queries.*

In the regime for which Theorem 1.3 is valid, the lower bound matches the block Krylov upper bound up to constant factors, and hence we settle the complexity of sampling from Gaussians in this regime. Moreover, Theorem 1.3 implies the *first* dimension-dependent lower bounds for general log-concave sampling. We conjecture that Theorem 1.3 holds for all κ for which $\sqrt{\kappa}\log d \le d$, and we leave this question for future work.

Although Theorem 1.3 may appear to only be a mild improvement over Theorem 1.2, analyzing this regime is quite delicate, and we believe that the tools based on Wishart matrices employed in the proof of Theorem 1.2 may be insufficient to reach Theorem 1.3. Instead, we prove Theorem 1.3 by first establishing sharp lower bounds on the performance of block Krylov algorithms for the sampling task, and then providing a novel reduction (Lemma 5.16) which shows that block Krylov algorithms are optimal for this task. This reduction is quite general, and as the block Krylov algorithm and the matrix-vector query model are of wide interest in scientific computing and numerical linear algebra, we believe that our reduction may be broadly useful for tackling other problems in this space.

We remark that a concise way of summarizing Theorems 1.2 and 1.3 if we do not care about lower order terms is that sampling from Gaussians requires $\widetilde{\Omega}(\min(\sqrt{\kappa}\log d, d))$ queries, where we write $f = \widetilde{\Omega}(g)$ to mean $f = \Omega(g \log^{-O(1)}(g))$.

Remark 1.4. Our results do not settle the complexity of zeroth-order methods, such as the ball walk or the Metropolized random walk, since one expects stronger lower bounds in the zeroth-order setting. In particular, in the settings of Theorems 1.2 and 1.3, we conjecture that a lower bound of $\Omega(\min(\sqrt{\kappa}d, d^2))$ holds.

1.2 Related work

There is a vast literature on from sampling log-concave (and non-log-concave) distributions, and a full survey is beyond the scope of this paper. For a detailed exposition, see e.g. [Chewi 2024].

Lower bounds for log-concave sampling. As previously mentioned, the only unconditional lower bound against log-concave sampling is by [Chewi et al. 2022b] for the one-dimensional setting, where the tight bound is $\Theta(\log\log\kappa)$. Other prior work on sampling lower bounds has fallen largely into one of several categories. One line of work studies lower bounds against a specific class of algorithm such as underdamped Langevin [Cao et al. 2021] or MALA [Chewi et al. 2021; Lee et al. 2021a; Wu et al. 2022]. However, these lower bounds techniques are tailored to the restricted class of algorithms that they consider and are not suitable for proving general query lower bounds. Another line of work considers lower bounds against computing normalizing constants [Ge et al. 2020; Rademacher and Vempala 2008]. The work [Talwar 2019] also investigates the computational complexity of sampling.

We mention two further lower bounds in different settings. The work of [Chatterji et al. 2022] proves a lower bound against stochastic gradient oracles, and the work of [Gopi et al. 2022] proves a lower bound on the number of individual function value (i.e., zeroth-order) queries needed to sample from a density of the form $\exp(-\sum_{i\in I} f_i + \mu\,\|\cdot\|^2)$, where each f_i is convex, Lipschitz, and whose domain is the unit ball. In contrast, we consider deterministic, first-order oracle access. Moreover, their considerations are somewhat orthogonal to ours: [Chatterji et al. 2022] focuses more on the role of noise, whereas we consider exact gradient access; and the lower bound of [Gopi et al. 2022] applies a direct reduction from optimization, which is also not in the spirit of the present work (in particular, we explicitly set the mode of the target distribution to zero).

Finally, we also mention the recent work [Chewi et al. 2023], which proves query lower bounds for non-log-concave sampling in a different metric (the Fisher information). This work is inspired by the corresponding upper bounds of [Balasubramanian et al. 2022] and can be viewed as lower bounds against *local mixing*.

Upper bounds for log-concave sampling. Starting with the seminal papers of [Dalalyan 2017; Dalalyan and Tsybakov 2012; Durmus and Moulines 2017], there has been a flurry of recent work on proving non-asymptotic guarantees for log-concave sampling, with iteration complexities that scale polynomially in the condition number and dimension. This includes analyses for the classical Langevin dynamics [Altschuler and Talwar 2023; Balasubramanian et al. 2022; Chewi et al. 2022a; Dalalyan and Karagulyan 2019; Durmus et al. 2019; Vempala and Wibisono 2019; Wibisono 2018], mirror and proximal methods [Ahn and Chewi 2021; Chen et al. 2022; Chen and Eldan 2022; Chewi et al. 2020; Fan et al. 2023; Gatmiry and Vempala 2022; Jiang 2021; Lee et al. 2021b; Li et al. 2022; Salim and Richtarik 2020; Wibisono 2019; Zhang et al. 2020], the Metropolis-adjusted Langevin algorithm (MALA) [Altschuler and Chewi 2024; Chen et al. 2020; Chewi et al. 2021; Dwivedi et al. 2018; Lee et al. 2020; Wu et al. 2022], and many others [Cheng et al. 2018; Dalalyan and Riou-Durand 2020; Ding et al. 2021; Ma et al. 2021; Shen and Lee 2019].

Our upper bound for sampling from Gaussians (Theorem B.3) is closely related to the use of the conjugate gradient algorithm for sampling from Gaussians [Nishimura and Suchard 2022]. Also, our $O(\log\kappa)$ upper bound algorithm is closely related to rounding procedures which have been previously used in the convex body sampling literature (see, e.g., [Lovász and Vempala 2006]).

Matrix-vector product query model. While matrix-vector queries have been studied in scientific computing for decades (e.g., [Bai et al. 1996]), they have only been studied in the theoretical computer science literature recently, with a fully formalized model described in [Sun et al. 2019]. The most relevant works to ours are those that study the matrix-vector query complexity of spectral properties, such as estimating top eigenvectors [Braverman et al. 2020; Simchowitz et al. 2018], trace and matrix norms [Dharangutte and Musco 2021; Hutchinson 1990; Meyer et al. 2021; Rashtchian et al. 2020; Wimmer et al. 2014], the full eigenspectrum [Braverman et al. 2022; Cohen-Steiner et al. 2018], and low-rank approximation [Bakshi et al. 2022; Musco and Musco 2015]. We remark that the non-adaptive matrix-vector product model is closely related to sketching, which has enjoyed a large body of work (see, e.g., [Woodruff 2014] for a survey).

2 Technical overview

Here we summarize the main technical ideas used to prove our lower bounds. For details, see Section 3 for Theorem 1.1, Section 4 for Theorem 1.2, and Section 5 for Theorem 1.3.

Notably, the proofs of our three main theorems rely on different methods; this is because we can take advantage of additional tools from the matrix-vector query literature in the Gaussian setting which do not apply to our construction for Theorem 1.1. It is conceivable that Theorem 1.3 could eventually encompass Theorem 1.2 by removing the restriction $\kappa \le d^{1/5-\delta}$, but this would require a more detailed analysis which is beyond the scope of the current paper.

2.1 Geometric construction in low dimension

Theorem 1.1 is proved with a construction in dimension two. For convenience, in this section we use radial coordinates to denote points in $\mathbb{R}^2$, so $\omega := (x, y) = (r, \theta)$, where $r \in \mathbb{R}_+$ and $\theta \in [0, 2\pi)$. We denote sectors of $\mathbb{R}^2$ enclosed by angles θ_1 and θ_2 as $S(\theta_1, \theta_2) := \{(r, \theta) \in \mathbb{R}^2 : \theta \in [\theta_1, \theta_2]\}$, and denote bounded sectors as $S_{\mathrm{bdd}}(\theta_1, \theta_2, r) := \{(r', \theta) \in \mathbb{R}^2 : \theta \in [\theta_1, \theta_2],\ r' \le r\}$.

The argument is information-theoretic in nature. We will construct a family of strongly log-concave and log-smooth distributions $\{\pi_1, \ldots, \pi_m\}$, where each $\pi_b \propto \exp(-V_b)$, which satisfies two key properties. First, different distributions π_b and $\pi_{b'}$ are well separated in total variation distance; and second, if b is chosen uniformly at random from $[m]$, then querying the potential $(V_b(\omega), \nabla V_b(\omega))$ at any $\omega \in \mathbb{R}^2$ will reveal $O(1)$ bits of information about b. The lower bound in Theorem 1.1 follows readily from the existence of such a family, provided that m and κ are polynomially related. On the one hand, because the distributions are well-separated in total variation, if we can sample well from the distribution π_b using queries, we can identify the index b with high probability. On the other hand, because there are m distributions and every query reveals $O(1)$ bits of information about b, we need at least $\Omega(\log m) = \Omega(\log \kappa)$ queries to identify b, which results in a $\Omega(\log \kappa)$ query lower bound for log-concave sampling.

How do we construct such a family? A first attempt is to consider distributions supported on thin convex sets that have no overlap. For $b = \frac{1}{\kappa}, \frac{2}{\kappa}, \ldots, 1$, let $\pi_b = \mathsf{uniform}(\mathcal{Z}_b)$, where $\mathcal{Z}_b = S_{\mathrm{bdd}}(\frac{\pi}{2}\, b, \frac{\pi}{2}\,(b + \frac{1}{2\kappa}), 1)$, and the size of the family is $m = \lfloor \kappa \rfloor$. The potential V_b is the convex indicator of $\mathcal{Z}_b$, i.e., it is 0 on $\mathcal{Z}_b$ and $+\infty$ outside. Morally, the distributions π_b can be thought of as having condition number κ.

This family does satisfy the two properties needed for the lower bound: different distributions are certainly well-separated because they have disjoint supports; and when we query any potential V_b at a point $\omega \in \mathbb{R}^2$, we always receive one bit of information: whether or not ω lies in the support of π_b. However, the distributions in this family are neither strongly log-concave nor log-smooth. It is easy to make them strongly log-concave while still satisfying the desired properties: we can adjust the distributions by adding the same quadratic function $\frac{\|\cdot\|^2}{2}$ to all of the potentials V_b. But it is much harder to make this family log-smooth.

One way to make this construction log-smooth is to let the potentials V_b grow slowly (linearly) to infinity outside of the their zero sets $\mathcal{Z}_b$, which leads to a modified second attempt: for $m = \kappa^{\Omega(1)}$, $b = \frac{1}{m}, \dots, 1$, let π_b have potential $V_b = \tilde{V}_b + \frac{\|\cdot\|^2}{2\kappa^{O(1)}}$, where $\mathcal{Z}_b = S(\frac{\pi}{2}\,b, \frac{\pi}{2}\,(b + \frac{1}{2m}))$, and $\tilde{V}_b(\omega) = \kappa \operatorname{dist}(\omega, \mathcal{Z}_b)$. Note that the potentials V_b are in fact still not smooth at the boundaries of the sets $\mathcal{Z}_b$, but this can be fixed by mollifying V_b. The distributions in this family will be well-separated, because an $\Omega(1)$ fraction of the mass of π_b will lie in $\mathcal{Z}_b$, and the sets $\mathcal{Z}_b$ are disjoint for different b. Unfortunately, this family no longer reveals $O(1)$ bits per query: for any $\omega \in \mathbb{R}^2$, we can identify b with a single query to $(V_b(\omega), \nabla V_b(\omega))$, because either $\omega \in \mathcal{Z}_b$, or $\nabla V_b(\omega)$ reveals the direction of $\mathcal{Z}_b$, and in both cases the index b itself is identified.

We can reduce the information revealed by queries by more carefully controlling the growth of $\tilde{V}_b$, so that the further away a point ω lies from $\mathcal{Z}_b$, the fewer the number of bits will be revealed by $(\tilde{V}_b(\omega), \nabla\tilde{V}_b(\omega))$. This motivates a third attempt at the construction. For $m = 2^N = \kappa^{\Omega(1)}$, $b = \frac{1}{m}, \dots, 1 - \frac{1}{m}$, let $b = 0.b_1 \dots b_N$ be the binary expansion of b, and let $[b]_k = 0.b_1 \dots b_k$ be the truncation of b up to the k-th bit. For $k = 1, \dots, N$, let $\mathcal{Z}_{k,b}^{\text{radial}} = S(\frac{\pi}{2}\,[b]_k, \frac{\pi}{2}\,([b]_k{+}2^{-k}))$, and let $\phi_{k,b}^{\text{radial}}(x) = \kappa^{O(1)}\,2^{-k} \operatorname{dist}(x, \mathcal{Z}_{k,b}^{\text{radial}})$. Finally, let $V_b^{\text{radial}} = \frac{\|\cdot\|^2}{2\kappa^{O(1)}} + \tilde{V}_b^{\text{radial}}$, where

$$\tilde{V}_b^{\text{radial}} = \max_{k=1,\dots,N} \phi_{k,b}^{\text{radial}}\,.$$

The potentials V_b^{radial} will again have to be mollified to be made smooth. It turns out that the potentials $\tilde{V}_b^{\text{radial}}$ will grow fast enough outside $\mathcal{Z}_{N,b}^{\text{radial}}$ such that the distributions will be well-separated. It also turns out that queries indeed reveal $O(1)$ bits of information on average. This can be seen as follows: note that the sets $\mathcal{Z}_{k,b}^{\text{radial}}$ are sectors such that $\mathcal{Z}_{k,b}^{\text{radial}} \supset \mathcal{Z}_{k+1,b}^{\text{radial}}$, and as k increases, $\mathcal{Z}_{k,b}^{\text{radial}}$ becomes thinner around the ray $\{\theta = \frac{\pi}{2}\,b\}$; also note that as k increases, the growth rate of $\phi_{k,b}^{\text{radial}}$ outside its zero set $\mathcal{Z}_{k,b}^{\text{radial}}$ is decreasing; these two properties imply that if we query a point $\omega = (r, \theta)$ that is far from the sector $\mathcal{Z}_{i,b}^{\text{radial}}$ (in the sense that $\theta \notin [\frac{\pi}{2}\,[b]_i{-}100{\cdot}2^{-i}, \frac{\pi}{2}\,[b]_i{+}100{\cdot}2^{-i}]$), then the value of $\tilde{V}_b^{\text{radial}}(\omega)$ will not depend on any $\phi_{k,b}^{\text{radial}}$ for $k > i$, and hence querying $\tilde{V}_b^{\text{radial}}(\omega)$ will only reveal b up to the i-th bit. As a result, if b is chosen uniformly, then for a fixed query ω with high probability we will have $\omega \notin \mathcal{Z}_{k,b}^{\text{radial}}$ for any $k = O(1)$, so the query will only reveal $O(1)$ bits of information about b.

Yet this construction fails because of the mollification step, which we have so far ignored. To make the potentials V_b smooth, we will instead take $V_b = \chi_\delta * \tilde{V}_b^{\text{radial}} + \frac{\|\cdot\|^2}{2\kappa^{O(1)}}$, where χ_δ is supported on a ball of radius $\delta < 2^{-2N}$. We would hope that the potential $\chi_\delta * \tilde{V}_b^{\text{radial}}$ still satisfies the property that querying a point $\omega = (r, \theta)$ that is far from $\mathcal{Z}_{i,b}^{\text{radial}}$ only reveals b up to the i-th bit. When r is not too close to the origin (say $r > 100 \cdot 2^{-i}$), this is indeed still true: if ω satisfies $\theta \notin [\frac{\pi}{2}\,[b]_i - 200 \cdot 2^{-i}, \frac{\pi}{2}\,[b]_i + 200 \cdot 2^{-i}]$, then the entire δ-neighbourhood of ω will satisfy $\theta \notin [\frac{\pi}{2}\,[b]_i - 100 \cdot 2^{-i}, \frac{\pi}{2}\,[b]_i + 100 \cdot 2^{-i}]$, so the value of $\tilde{V}_b^{\text{radial}}$ on the δ-neighbourhood of ω will not depend on any $\phi_{k,b}^{\text{radial}}$ for $k > i$, hence the value of $(\chi_\delta * \tilde{V}_b^{\text{radial}})(\omega)$ will also not reveal any information of b beyond the i-th bit. But when ω is very close to the origin ($r < \delta$), the δ-neighbourhood of ω will intersect $\mathcal{Z}_{N,b}^{\text{radial}}$, which means that the value of $(\chi_\delta * \tilde{V}_b^{\text{radial}})(\omega)$ will depend on $\phi_{k,b}^{\text{radial}}$ for all k and hence on all bits of b. In other words, mollification leaks information around the origin. As a result, if we query points δ-close to the origin, we will again identify b in a single query.

The way to resolve the leakage at the origin is to create a branching structure, such that all V_b are equal near the origin so that no information is leaked at small scales, and such that far away from the origin V_b is small around the ray $\{\theta = \frac{\pi}{2}\,b\}$ so that π_b still concentrates around different sectors. We keep the choices of m and b from the previous construction. The potentials will be $V_b = \chi_\delta * \tilde{V}_b + \frac{\|\cdot\|^2}{2\kappa^{O(1)}}$, where $\tilde{V}_b = \max_{k=1,\dots,N} \phi_{k,b}$, and

$\phi_{k,b}(\omega) = \kappa^{O(1)}\, 2^{-k}\, \mathrm{dist}(\omega, \mathcal{Z}_{k,b})$. The zero set $\mathcal{Z}_{k,b}$, instead of being a radial sector like $\mathcal{Z}_{k,b}^{\mathrm{radial}}$, is now thickened adaptively.

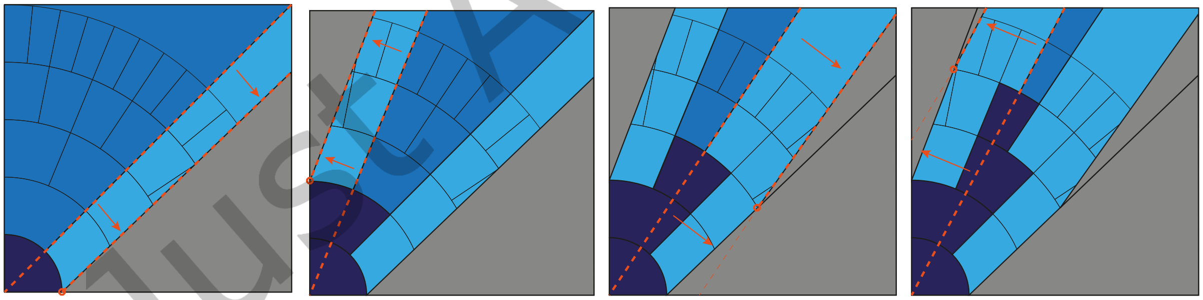
We intuitively describe how to generate $\mathcal{Z}_{k,b}$. Each $\mathcal{Z}_{k,b}$ will be a thickening of $\mathcal{Z}_{k,b}^{\mathrm{radial}}$, by simply including all points within some distance d_k of $\mathcal{Z}_{k,b}^{\mathrm{radial}}$. We define $\mathcal{Z}_{\le k,b} := \bigcap_{k' \le k} \mathcal{Z}_{k',b}$: note that each $\mathcal{Z}_{\le k,b}$ is getting smaller as k increases, and $\mathcal{Z}_{\le N,b}$ is the zero set of $\tilde{V}_b$.

Consider some radii $r_0 < r_1 < r_2 < \dots$. To generate $\mathcal{Z}_{1,b}$, we thicken $\mathcal{Z}_{1,b}^{\mathrm{radial}}$ (corresponding to the radial sector matching on the first bit), so that it contains $S_{\mathrm{bdd}}(0, \pi/2, r_0)$ (corresponding to the quarter-circle near the origin). This avoids leaking information near the origin, as every x within radius r will be in $\mathcal{Z}_{1,b}$, which means $\phi_{1,b}$ will also be 0. Indeed, we can thicken $\mathcal{Z}_{1,b}^{\mathrm{radial}}$ just the right amount so that it contains $S_{\mathrm{bdd}}(0, \pi/2, r_0)$. For the concrete example where $N = 4$, and $b = 0.1010$, we show a description of $\mathcal{Z}_{1,b}$ in Figure 1a: we shade $S_{\mathrm{bdd}}(0, \pi/2, r_0)$ in dark blue, $\mathcal{Z}_{1,b}^{\mathrm{radial}} = S(\pi/4, \pi/2)$ in medium blue, and the additional thickening required in light blue.

To generate $\mathcal{Z}_{k,b}$ for $k \ge 2$, we thicken a much thinner angular sector. This ensures that at large radii, the arc of $\mathcal{Z}_{k,b}$ is not too big. We will inductively thicken $\mathcal{Z}_{k,b}$ by some amount d_k just enough to contain $\mathcal{Z}_{k-1,b} \cap S_{\mathrm{bdd}}(0, \pi/2, r_{k-1})$. Consider one more example for $k = 2$ (again for $N = 4$, and $b = 0.1010$), in Figure 1b. Note that $\mathcal{Z}_{2,b}^{\mathrm{radial}}$ is the sector $S(\frac{\pi}{4}, \frac{3\pi}{8})$ (shaded in medium blue), and the thickened region (in light blue emanating from both sides of the sector) is just enough to capture all of $\mathcal{Z}_{1,b}$ that was within radius r_1. However, for larger radii, $\mathcal{Z}_{2,b}$ is much thinner than $\mathcal{Z}_{1,b}$. In addition, if we know the first bit $b_1 = 1$, then querying V_b anywhere in $\{r \le r_1\}$ will not reveal any information about the second bit b_2. This is because either we were in $\mathcal{Z}_{1,b}$ which only depends on b_1 (in which case $\phi_{1,b} = \phi_{2,b} = 0$ as we thickened to make sure $\mathcal{Z}_{2,b} \supset Z_{1,b} \cap S_{\mathrm{bdd}}(0, \pi/2, r_1)$), or we weren't, in which case $\phi_{1,b}$ grows much more quickly than $\phi_{2,b}$.

We can also continue this process inductively for $k = 3, 4$ (Figures 1c and 1d): we show $\mathcal{Z}_{\le k,b}$. The intuition for why this prevents leaking of information near the origin is that even if k is large, $\mathcal{Z}_{k,b}$ in the smaller-radius regions is decided by $\mathcal{Z}_{k',b}$ for $k' \ll k$, so we cannot learn any later bits.

The comparisons of $\mathcal{Z}_{k,b}^{\mathrm{radial}}$ and $\mathcal{Z}_{\le k,b}$ for $b = 0.1010$ and for all $k \le 4$ are shown together in Figure 1. The picture is not to scale, and the radial arcs represent the radii $r_i = 2^i r_0$, for $i = 0, \dots, 4$.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$ (d) $k = 4$

Fig. 1. Comparison of $\mathcal{Z}_{\le k,b}^{\mathrm{radial}}$ (the sector in medium blue) with $\mathcal{Z}_{k,b}$ (union of dark, medium, and light blue), for $k = 1, 2, 3, 4$, and $b = 0.1010$. Dark blue represents the larger angular sectors closer to the origin, and light blue represents the additional fattening from taking sumsets. Each $\mathcal{Z}_{k,b}$ is constructed by thickening $\mathcal{Z}_{k,b}^{\mathrm{radial}}$ enough (illustrated by the red arrows) such that no information about the k-th bit is revealed close to the origin, but $\mathcal{Z}_{k,b}$ continues to get thinner at large radii.

The construction of $\mathcal{Z}_{\le k,b}$ means that for $k > 1$, querying $\phi_{k,b}$ within $\{r \le 2^{k-1} r_0\}$ will not reveal the k-th bit, and so even querying the mollified $\chi_\delta * \phi_{k,b}$ within $\{r \le 2^{k-2} r_0\}$ will not reveal the k-th bit, which stops information leaking near the origin.

Since $\tilde{V}_b = \max_{k=1,\ldots,N} \phi_{k,b}$, the zero set of $\tilde{V}_b$ coincides with $\mathcal{Z}_{\leq N,b}$, and for the choice of $b = 0.1010$, this is shown in the first panel of Figure 2. It turns out that each π_b will concentrate around the zero set of $\tilde{V}_b$, and the other panels of Figure 2 show these zero sets for seven different values of b in the set $\{\frac{1}{16}, \ldots, \frac{15}{16}\}$ at larger scales. We can see that far out from the origin the zero sets become well-separated, and hence the distributions are well-separated in total variation.

We already discussed how the thickening of $\mathcal{Z}_{k,b}$ means that querying $\phi_{k,b}$, and hence $\tilde{V}_b$, near the origin will not reveal the higher bits of b. For query points $\omega = (r, \theta)$ where r is large, the same analysis on $\tilde{V}_b^{\text{radial}}$ tells us that $\tilde{V}_b(x)$ (even after mollification) will reveal $O(1)$ bits of information about b when b is chosen uniformly. As mentioned earlier, such a family of distributions readily leads to a sampling lower bound of $\Omega(\log m)$, where m is the size of the family. Since we can choose $m = \kappa^{\Omega(1)}$, this leads to the $\Omega(\log \kappa)$ lower bound. Details of the proof can be found in Section 3.

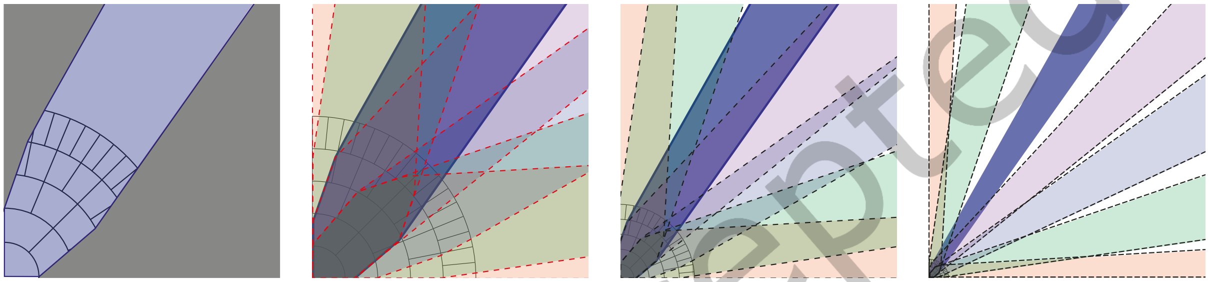


Fig. 2. Zeros sets of $\tilde{V}_b$. The first panel shows the zero set for $b = 0.1010$. The other panels show the zeros sets for different values of b at different scales. Note that far away from the origin the zero sets become well-separated, which leads to the distributions being well-separated in total variation. Note that if b, b' match in the first ℓ bits, then they will agree up to the ℓ-th circle, as those circles only depend on $\mathcal{Z}_{\leq \ell,b}$ even for ℓ much less than K.

Connections to Kakeya constructions. The construction outlined above is related to Perron's construction [Perron 1928] of Besicovich (Kakeya) sets known as *Perron trees*. Kakeya sets are sets with area zero that contain the translation of a unit segment in any direction. While Kakeya sets over finite fields have been investigated before in theoretical computer science, e.g., [Dvir 2009; Jukna 2011; Saraf and Sudan 2008], our construction is inspired by Kakeya sets over continuous domains, namely $\mathbb{R}^2$. To our knowledge, this is one of the first applications of these geometric ideas to theoretical computer science.

There are many similarities between our construction and that of Perron. Perron's construction proceeds by the method of sprouting. Sprouting is an iterative process in which, at each step, one adds further and further smaller triangles to the pre-existing construction. The figure is then rescaled in order to have height 1. The construction after n steps contains 2^n triangles of small aperture $\Omega(2^{-n})$, and has area $O(n^{-1})$. We do a similar process in the definition of our sets $\mathcal{Z}_{k,b}$, and indeed, ultimately our hard instance has a very similar tree-like structure.

While we were inspired by the construction of Perron trees, there are also key differences between our hard instance and Perron's construction. Indeed, in our setting, we need to minimize overlap (so that the resulting distributions are well-separated) while simultaneously ensuring that information is not leaked by queries. In contrast, Kakeya sets are explicitly designed to maximize overlap. Secondly, the iterates of Perron trees are convex sets, not convex functions. One must turn these convex sets into convex functions somehow. This is additionally complicated by the fact that these iterates are not nested. In our construction, we must take great care to create nested convex sets, so that the resulting functions are convex and still maintain the structure of the sets.

2.2 Lower bounds for sampling from Gaussians

We now turn to our lower bounds against sampling from Gaussians. Recall that our goal is to provide a lower bound on sampling from a Gaussian $\mathcal{N}(0,\Sigma)$, where $\Lambda \coloneqq \Sigma^{-1}$ has condition number κ. Note that the corresponding potential is $V(x) = \frac{1}{2}\langle x, \Lambda x\rangle$, and we are allowed zeroth-order and first-order queries, which means for a query x, we receive $x^\top \Lambda x$ and Λx. Hence, adaptive queries are equivalent to adaptive matrix-vector product computations with Λ.

The first observation we make is that we can reduce the problem of sampling from the Gaussian to estimating the trace of Σ. This is because if X is a sample from a distribution which is close in total variation distance to $\mathcal{N}(0,\Sigma)$, then $\|X\|_2^2 \approx \mathrm{tr}(\Sigma)$ with high probability. Therefore, it suffices to demonstrate a lower bound for the following problem: given matrix-vector product computations with Λ, approximately compute $\mathrm{tr}(\Lambda^{-1})$.

2.2.1 Lower bound via Wishart matrices. For any d, let $W \in \mathbb{R}^{d\times d}$ have the $\mathrm{Wishart}(d)$ distribution. That is, $W = XX^\top$, where $X \in \mathbb{R}^{d\times d}$ has i.i.d. $\mathcal{N}(0, 1/d)$ entries. We take W to be the precision matrix, $\Lambda = W$. Our first lower bound shows that $\Omega(d)$ matrix-vector queries with W are necessary to estimate the trace of W^{-1} even to constant multiplicative accuracy, with constant success probability (Theorem 4.2). Since the condition number of W is $\Theta(d^2)$ with high probability, we obtain one extreme of the claimed lower bound $\Omega(\min(\sqrt{\kappa}, d))$. The general lower bound for all κ then follows from a padding argument.

This lower bound approach is inspired by [Braverman et al. 2020], which proved a query lower bound for estimating the minimum eigenvalue of W. Their approach relies on the fact that if we condition on any sequence of $(1-\Omega(1))\,d$ adaptive queries, the posterior distribution of the remaining eigenvalues behaves similarly to the original distribution of the eigenvalues of W. In addition, while the smallest eigenvalue of W is usually about $1/d^2$, its distribution has heavy tails: with probability $\Theta(\sqrt{\varepsilon})$, the smallest eigenvalue of W is below ε/d^2. Consequently, even conditioned on $d/2$ adaptive queries, we are unable to learn the minimum eigenvalue up to a constant factor with high probability.

In our setting, we instead wish to show that learning the trace of W^{-1} is hard. However, the smallest eigenvalue of the Wishart matrix is so small that with high probability, $\mathrm{tr}(W^{-1}) = \Theta(\lambda_{\min}(W)^{-1})$. While most of the time the trace is $O(d^2)$, with probability $\Theta(\sqrt{\varepsilon})$ the posterior distribution of the smallest eigenvalue of W after our adaptive queries may be ε/d^2. Hence, we will be unable to determine whether the trace is $\le O(d^2)$ or $\ge \Omega(d^2/\varepsilon)$ with high probability.

This lower bound technique is clean and nearly optimal, but as previously mentioned it is vacuous (of constant order) when $\kappa = O(1)$, whereas we expect the complexity of the problem to increase as $d \to \infty$. To tackle this setting, we introduce a second approach.

2.2.2 Lower bounds via reduction to block Krylov. Our second technique works in two parts. First, we show that for a specific hard distribution over instances, any block Krylov-style algorithm requires $\Omega(\min(\sqrt{\kappa}\log d, d))$ queries to estimate $\mathrm{tr}(\Sigma)$. Then, we show a general purpose reduction which demonstrates that for this hard instance (and indeed, any rotationally invariant instance), block Krylov methods are actually optimal.

Lower bound for block Krylov algorithms. Recall the block Krylov technique: the algorithm chooses K i.i.d. random vectors $v_1, \dots, v_K \sim \mathcal{N}(0, I)$, and computes $\Lambda^j v_k$ for all $j \le T, k \le K$. This can be done using KT adaptive queries, by querying $\Lambda^j v_k$ to learn $\Lambda^{j+1} v_k$. For our purposes, it suffices to consider block Krylov algorithms with $K = T$ and to prove a lower bound on the smallest number K needed to successfully estimate $\mathrm{tr}(\Sigma)$, for $\Sigma = \Lambda^{-1}$.

We will construct two diagonal matrices D, D' with all eigenvalues between 1 and κ, such that $\mathrm{tr}(D^{-1})$ and $\mathrm{tr}((D')^{-1})$ are sufficiently different. In addition, if Λ, Λ' are random rotations of D, D', respectively, then $\{\Lambda^j v_k\}_{j,k\le K}$ and $\{(\Lambda')^j v_k\}_{j,k\le K}$ are hard to distinguish for $K \le c\sqrt{\kappa}\log d$ for a small constant c (Lemma 5.8). Thus, unless $K \ge \Omega(\sqrt{\kappa}\log d)$, we cannot estimate the trace.

To explain the intuition behind Lemma 5.8, we first consider what happens if we only have $\{\Lambda^j v\}_{j\leq K}$ for a single random vector v (i.e., power method). Letting $\lambda_1, \dots, \lambda_d$ be the eigenvalues of Λ, we have $\Lambda^j v = \sum_{i=1}^d \lambda_i^j \alpha_i u_i$, where u_i is the i-th eigenvector of Λ and $v = \sum_{i=1}^d \alpha_i u_i$. Intuitively, the only information we obtain from these vectors are their pairwise inner products, since we could have randomly rotated Λ. Therefore, the only information we have is $\langle \Lambda^j v, \Lambda^{j'} v\rangle = \sum_{i=1}^d \lambda_i^{j+j'} \alpha_i^2$, which is the set $\{\sum_{i=1}^d \lambda_i^j \alpha_i^2\}_{j\leq 2K}$. Since v is random, we may think of all of the α_i^2 as 1 for simplicity, and so we know $\{\sum_{i=1}^d \lambda_i^j\}_{j\leq 2K}$. Our goal is to use this information to learn $\operatorname{tr}(\Lambda^{-1}) = \sum_{i=1}^d \lambda_i^{-1}$.

We connect this to the problem of estimating $1/x$ as a linear combination of $1, x, x^2, \dots, x^K$, a classic problem in approximation theory that is often tackled with *Chebyshev polynomials*. Indeed, this relation to Chebyshev polynomials is the main tool in the analysis of essentially all Krylov methods. In our setting, as we desire lower bounds, we apply the fact that Chebyshev polynomials are *optimal* in generating certain approximations. More concretely, suppose that there are only K distinct eigenvalues $\lambda_1, \dots, \lambda_K$, with each λ_i having some multiplicity N_i. Since we want to show that estimating $\operatorname{tr}(\Lambda^{-1})$ is hard, this amounts to showing that knowing $\sum_{i=1}^K N_i \lambda_i^j$ for $0 \leq j \leq K$ is insufficient to learn $\sum_{i=1}^K N_i/\lambda_i$. We express this as a linear program (if we relax the N_i to be reals), the dual of which precisely captures whether $1/x$ can be approximated well by a degree-K polynomial at $\lambda_1, \dots, \lambda_K$ (Proposition 5.5). If we choose the λ_i to be the local extrema of a degree-K Chebyshev polynomial, shifted so that $\lambda_1 = 1$ and $\lambda_K = \kappa$, then it is known that one cannot estimate $1/x$ up to error $d^{-\Omega(1)}$ at these points (which is needed for trace estimation), unless $K \geq \Omega(\sqrt{\kappa}\log d)$. At a high level, this is the reason why we need $\Omega(\sqrt{\kappa}\log d)$ iterations of power method.

For general block Krylov algorithms, the algorithm obtains $\langle v_\ell, \Lambda^j v_k\rangle$, for $0 \leq j \leq K$ and $1 \leq k, \ell \leq K$. Now, the information that the algorithm sees is captured by the matrices $\{\langle v_\ell, \Lambda^j v_k\rangle\}_{k,\ell\leq K}$, for $j = 1, \dots, K$. Here, we show that provided K is sufficiently small compared to d, we can find choices of multiplicities $N_1, \dots, N_K$ and $N_1', \dots, N_K'$, such that the corresponding matrices D, D' have significantly different traces (i.e., $\sum_{i=1}^K (N_i - N_i')/\lambda_i$ is large) but the information from queries is not enough to distinguish between Λ and Λ', which we establish via a coupling argument.

Reduction to block Krylov algorithms. The argument outlined above shows block Krylov algorithms with $K = o(\sqrt{\kappa}\log d)$ cannot distinguish between two families of randomly rotated matrices with difference traces (Λ coming from D and Λ' coming from D'), and hence cannot solve the trace estimation task. Our next technical contribution is a reduction which allows us to simulate the output of any *adaptive* algorithm with K queries on our hard instance, given only the responses to a block Krylov algorithm. Thus, a lower bound against block Krylov methods translates into a lower bound against any query algorithm. We now give a high-level description of the reduction.

Since we prove lower bounds based on randomized constructions, it suffices to consider adaptive deterministic algorithms, i.e., each query v_k is a deterministic function of the previous queries and oracle outputs. The difficulty of proving such a lower bound against such an algorithm is the adaptivity of the queries, which makes it difficult to reason about how much information the algorithm has learned. However, since our lower bound construction for block Krylov algorithms is rotationally invariant, intuitively the adaptivity does not help: the algorithm may as well query a random direction which it has not yet explored.

However, this intuition is not entirely correct: if the algorithm has previously queried a vector v and received the information Λv, then it may useful to query Λv in order to receive the information $\Lambda^2 v$, instead of querying a completely random new direction. Indeed, computing powers $v, \Lambda v, \Lambda^2 v, \dots$ is precisely the essence of the power method, as discussed above. To account for this, we move to the following stronger oracle model: if the algorithm has selected vectors $v_1, \dots, v_k$, then at iteration k it receives all of the information $(\Lambda^i v_j)_{i+j\leq k}$ for free. Now, there is provably no benefit to querying vectors which lie in the span of the previous queries and oracle outputs.

Recall that our goal is to argue that an adaptive deterministic algorithm can be simulated by an algorithm which simply makes i.i.d. Gaussian queries $z_1, z_2, \ldots, z_K$, in the following sense. In the stronger oracle model, at iteration k, the adaptive algorithm has made queries $(v_1^{\mathsf{alg}}, \ldots, v_k^{\mathsf{alg}})$ and received information $(\Lambda^i v_j^{\mathsf{alg}})_{i+j \le k}$ and it picks a new vector v_{k+1} which lies orthogonal to its received information. Suppose that using only the Gaussian queries $z_1, z_2, \ldots, z_k$, we have simulated queries $v_1^{\mathsf{sim}}, v_2^{\mathsf{sim}}, \ldots, v_k^{\mathsf{sim}}$ which are equivalent to the execution of the adaptive algorithm in the sense that the law of the information $(\Lambda^i v_j^{\mathsf{sim}})_{i+j \le k}$ is precisely the same as the law of the algorithm's information $(\Lambda^i v_j^{\mathsf{alg}})_{i+j \le k}$. Since the algorithm is deterministic, v_k^{alg} is a function $v_k((\Lambda^i v_j^{\mathsf{alg}})_{i+j<k})$ of algorithm's accumulated information. Thus, in order to simulate the adaptive algorithm for one more step, it is natural to consider taking $v_k^{\mathsf{sim}} := v_k((\Lambda^i v_j^{\mathsf{sim}})_{i+j<k})$. However, we will be unable to compute $\Lambda^i v_k^{\mathsf{sim}}$ for any $i \ge 1$, because the simulation must be based on the Gaussian queries $z_1, z_2, \ldots, z_k$, whereas this definition of v_k^{sim} requires making queries at $v_1^{\mathsf{sim}}, v_2^{\mathsf{sim}}, \ldots, v_{k-1}^{\mathsf{sim}}$.

Thus far, we have not invoked the rotational invariance of Λ, which is crucial to the argument. The key is that although we cannot directly take $v_k((\Lambda^i v_j^{\mathsf{sim}})_{i+j<k})$ to be our next simulated point, we can *rotate* $\tilde{v}_k$ into $v_k((\Lambda^i v_j^{\mathsf{sim}})_{i+j<k})$ via a unitary matrix U_k; moreover, we can arrange that U_k fixes all of the previous information $(\Lambda^i v_j^{\mathsf{sim}})_{i+j<k}$, because $v_k((\Lambda^i v_j^{\mathsf{sim}})_{i+j<k})$ lies orthogonal to this information (recall, we can assume that each deterministic function $v_k(\cdot)$ outputs a vector orthogonal to its inputs, due to our choice of oracle model). The intuition is that due to the rotational invariance of Λ, then conditioned on the data $(\Lambda^i v_j^{\mathsf{sim}})_{i+j<k}$, the distribution of Λ is still rotationally invariant on the orthogonal subspace of the data; hence, $U_k \tilde{v}_k = v_k((\Lambda^i v_j^{\mathsf{sim}})_{i+j<k})$ ought to have the same law as $\tilde{v}_k$, i.e., querying the completely random direction $\tilde{v}_k$ is just as good as querying according to what the adaptive algorithm specifies.

Unfortunately there are further difficulties to overcome with this approach. Namely, suppose that we define each simulated point v_k^{sim} to be the output $U_k \tilde{v}_k$ of a rotation matrix applied to $\tilde{v}_k$. We would like to take U_k such that $U_k \tilde{v}_k = v_k((\Lambda^i v_j^{\mathsf{sim}})_{i+j<k})$ but this is no longer computable based on $(\Lambda^i \tilde{v}_j)_{i+j<k}$. However, we note that $\Lambda^i v_j^{\mathsf{sim}} = \Lambda^i U_j \tilde{v}_j = U_j \tilde{\Lambda}^i \tilde{v}_j$ where $\tilde{\Lambda} := U_j^\mathsf{T} \Lambda U_j$. This shows that $\Lambda^i v_j^{\mathsf{sim}}$ is computed from the query of $\tilde{v}_j$, not on the original matrix Λ but on the modified matrix $\tilde{\Lambda}$, together with the matrix U_j. Since we hope that $\tilde{\Lambda}$ has the same law as Λ, then this is good enough for the purposes of simulating the adaptive algorithm. Actually, in order for the induction to work out, it becomes clear that we need to define a sequence of matrices $\Lambda_1, \Lambda_2, \ldots, \Lambda_k$, where each Λ_k is related to the previous Λ_{k-1} via $\Lambda_k = U_k^\mathsf{T} \Lambda_{k-1} U_k$, and U_k is chosen such that $v_k^{\mathsf{sim}} = U_k \tilde{v}_k = v_k((\Lambda_{k-1}^i v_j^{\mathsf{sim}})_{i+j<k})$. Then, we must argue that the simulated sequence $v_1^{\mathsf{sim}}, v_2^{\mathsf{sim}}, \ldots, v_k^{\mathsf{sim}}$ has the same law as the algorithm's sequence $v_1^{\mathsf{alg}}, v_2^{\mathsf{alg}}, \ldots, v_k^{\mathsf{alg}}$.

This last step, however, turns out to be delicate. Indeed, although it is obvious that for a *fixed* orthogonal matrix U', the law of Λ is the same as the law of $(U')^\mathsf{T} \Lambda U'$, the rotation matrices U_k we choose in the above argument are dependent on the previous queries and oracle outputs, and are hence dependent on Λ itself. In the presence of such dependence, it is not obvious why the law of Λ_k should be the same as the law of Λ, and to address this we prove a conditioning lemma in Section 5.3.2. Once the conditioning lemma is proved, the remainder of the proof follows along the lines just described, and the details of the induction are carried out in Section 5.3.3.

3 A general sampling lower bound in dimension two

3.1 Overview

Our goal is to show the following theorem:

Theorem 3.1 (lower bound in dimension two). *There is a universal constant $\varepsilon_0 > 0$ such that the following holds. The query complexity of sampling from the class of distributions $\pi \propto \exp(-V)$ on $\mathbb{R}^2$ such that V is 1-strongly convex, κ-smooth, and minimized at 0, with accuracy ε_0 in total variation distance, is at least $\Omega(\log \kappa)$.*

The strategy to do so will be to construct a finite family $\mathcal{S}$ of potentials in the given class which satisfies the following two properties:

- The potentials are *hard to identify via queries* (in the sense of Definition 3.6 below), and therefore any algorithm must query V at $\Omega(\log \kappa)$ points in order to identify which $V \in \mathcal{S}$ the algorithm is querying.
- The potentials are *well-separated* (in the sense of Definition 3.7 below), which loosely means that they have mostly non-overlapping support and hence (by Proposition 3.8) a single sample from $\pi \propto \exp(-V)$ suffices to identify $V \in \mathcal{S}$ with constant probability.

Before describing the potentials $\mathcal{S}$ in more detail, we note some basic definitions.

Definition 3.2. Given two functions $f, g : \mathbb{R}^d \to \mathbb{R}$, the *convolution* $f * g$ is the function defined as $(f * g)(x) := \int_{\mathbb{R}^d} f(y) g(x-y) dy$, for all $x \in \mathbb{R}^d$.

Definition 3.3. For $\delta > 0$, we define χ_δ to be the *indicator function of the ball* B_δ of radius δ around the origin. By this, we mean $\chi_\delta(x) = 1$ if $\|x\|_2 \le \delta$, and $\chi_\delta(x) = 0$ otherwise.

The family $\mathcal{S}$ of potentials will have cardinality $\kappa^{\Omega(1)}$, so that identification of the potential requires $\Omega(\log \kappa)$ bits of information. Actually, by rescaling the potentials, it suffices for each potential V to be $\kappa^{-O(1)}$-convex and $\kappa^{O(1)}$-smooth. Our eventual construction also satisfies the following properties.

- Each $V \in \mathcal{S}$ is of the form $V = \tilde{V} * \chi_\delta + \|\cdot\|^2/(2\kappa^{O(1)})$, where $\tilde{V} : \mathbb{R}^2 \to \mathbb{R}$ is a convex, non-negative, and piecewise linear potential, and δ will have scale $\delta = \kappa^{-\Theta(1)}$.
- Each $V \in \mathcal{S}$ is zero in a small neighbourhood of a ray ℓ emanating from the origin, and grows fast outside of this ray; hence, the potentials are well-separated.
- Suppose that ℓ, ℓ' are the rays corresponding to two potentials $V, V' \in \mathcal{S}$. At distances from ℓ and ℓ' that are much larger than the angle $\angle(\ell, \ell')$, the potentials V, V' are exactly equal. This is the property makes the potentials hard to identify via queries.

Throughout the proof, we assume that κ is sufficiently large, $\kappa \ge \Omega(1)$.

3.2 Definitions and the information-theoretic argument

Definition 3.4 (density and normalizing constant). Given a strictly convex function $V : \mathbb{R}^d \to \mathbb{R}$, we denote by P_V the probability distribution with density $Z^{-1} \exp(-V)$ w.r.t. Lebesgue measure, where $Z := \int \exp(-V)$ is the normalizing constant. In an abuse of notation, we also use P_V to refer to the density itself.

Definition 3.5 (queries and extended oracle). For a fixed potential V, and given a query $x \in \mathbb{R}^d$, the extended oracle responds with $V(B_\delta(x_1))$, which consists of the value of V for all points in the ball of radius δ centered at x. For a sequence of (possibly adaptive and randomized) queries $x_1, \dots, x_n$ and observations $V(B_\delta(x_1)), \dots, V(B_\delta(x_n))$, we denote the information from the i-th query by $\xi_i := \{x_i, V(B_\delta(x_i))\}$, and the information from all the queries by

$$\xi_{1:n} := \{\xi_1, \dots, \xi_n\}\,.$$

Note that the extended oracle in Definition 3.5 provides more information (the set of values of the potential in some ball around the query point x) to the algorithm than our original first-order query model, from which the algorithm only observes $(V(x), \nabla V(x))$ at the query x. A lower bound for sampling in this stronger query model clearly implies a lower bound in the original query model. We consider the stronger model out of technical convenience, as this notion is robust to the mollification in the construction of the potentials.

Definition 3.6 (hard to identify via queries). A finite set $\mathcal{S}$ of potentials in $\mathbb{R}^d$ is called *$\mathcal{I}$-hard to identify with queries at scale δ* if the following holds: for $V \sim \mathsf{uniform}(\mathcal{S})$, any sequence of queries $x_1, \dots, x_n$ to the extended oracle made by a deterministic adaptive algorithm satisfies

$$I(\xi_{1:n}; V) \le \mathcal{I}\, n\,,$$

where I denotes the mutual information.

Definition 3.7 (well-separated set). A set $\mathcal{S}$ of potentials is *well-separated* if there is a family of measurable sets $(\Omega_V)_{V\in\mathcal{S}}$ where the sets Ω_V are disjoint, and a universal constant $c > 0$ such that

$$P_V(\Omega_V) \ge c\,, \qquad \text{for all } V \in \mathcal{S}\,.$$

The motivation for this definition is the following fact:

Proposition 3.8 (one sample identifies well-separated distributions). *Let $\mathcal{S}$ be a well-separated set of potentials and conditionally on $V \sim \mathsf{uniform}(\mathcal{S})$, suppose that X is a sample from a probability measure $\widehat{P}_V$ which is at most $\frac{c}{2}$ away from P_V in total variation distance. Then,*

$$\mathbb{P}\{X \in \Omega_V\} \ge \frac{c}{2}\,.$$

Proof. By conditioning on V,

$$\mathbb{P}\{X \in \Omega_V\} = \mathbb{E}\,\mathbb{P}\{X \in \Omega_V \mid V\} = \mathbb{E}\,\widehat{P}_V(\Omega_V) \ge \mathbb{E}\big[P_V(\Omega_V) - \|P_V - \widehat{P}_V\|_{\mathrm{TV}}\big] \ge \frac{c}{2}\,,$$

which is what we wanted to show. □

This shows that the minimum-distance estimator

$$\widehat{V} \coloneqq \operatorname*{arg\,min}_{V\in\mathcal{S}} \inf_{z\in\Omega_V} \|X - z\| \tag{1}$$

succeeds at estimating the randomly drawn V with constant probability. On the other hand, we have Fano's inequality from information theory.

Theorem 3.9 (Fano's inequality, [Cover and Thomas 2006, Theorem 2.10.1]). *Suppose that $\mathcal{S}$ is a finite set and $V \sim \mathsf{uniform}(\mathcal{S})$. Suppose that $\widehat{V}$ is any estimator which is based on some data ξ. Then,*

$$\mathbb{P}\{\widehat{V} \ne V\} \ge 1 - \frac{I(\xi; V) + \log 2}{\log|\mathcal{S}|}\,.$$

Fano's inequality enables us to reduce Theorem 3.1 to the following proposition:

Proposition 3.10 (well-separated set which is hard to identify via queries). *Let $\kappa \ge \Omega(1)$. Then, there is a set $\mathcal{S}$ of potentials such that:*

(1) All elements of $\mathcal{S}$ are $\kappa^{-O(1)}$-convex and $\kappa^{O(1)}$-smooth, and have their minimum at zero.
(2) $\mathcal{S}$ has cardinality $\kappa^{\Omega(1)}$.
(3) $\mathcal{S}$ is well-separated with $c = \Omega(1)$.
(4) $\mathcal{S}$ is hard to identify via queries at scale $\delta = \kappa^{-\Theta(1)}$, and with $\mathcal{I} = O(1)$.

Proof of Theorem 3.1. Suppose that there is a sampling algorithm which, given any target distribution $\pi \propto \exp(-V)$ on $\mathbb{R}^2$ such that V is 1-strongly convex, $\bar{\kappa}$-smooth, and minimized at 0, outputs a sample X whose law is ε_0 close in total variation distance to π using $n(\bar{\kappa})$ queries to the extended oracle. Let $\mathcal{S}$ be the family in Proposition 3.10. By choosing $\varepsilon_0 = c/2 = \Omega(1)$ and rescaling the potentials accordingly, then Proposition 3.8

implies that the sampling algorithm can identify $V \sim \mathsf{uniform}(\mathcal{S})$ using $n(\bar\kappa)$ queries with constant probability, where $\bar\kappa = \kappa^{O(1)}$. Namely, for the estimator $\widehat{V}$ in (1),

$$\mathbb{P}\{\widehat{V} = V\} \ge \frac{c}{2} = \Omega(1)\,. \tag{2}$$

On the other hand, we can prove a lower bound for the error probability of any estimator $\widehat{V}$ constructed using adaptive queries. First we assume that the estimator is deterministic given previous queries. Because the set $\mathcal{S}$ is hard to identify, by Fano's inequality (Theorem 3.9) we have

$$\mathbb{P}\{\widehat{V} \ne V\} \ge 1 - \frac{I(\xi_{1:n(\bar\kappa)}; V) + \log 2}{\log|\mathcal{S}|} \ge 1 - \frac{\mathcal{I}\, n(\bar\kappa) + \log 2}{\log|\mathcal{S}|} = 1 - \Omega\Big(\frac{n(\bar\kappa)}{\log\kappa}\Big)\,, \tag{3}$$

for all $n(\bar\kappa) \le c\,|\mathcal{S}| = O(\log\kappa)$. If the estimator is instead randomized, it depend on a random seed ζ that is independent of V. In this case, the same argument as above conditional on ζ gives

$$\mathbb{P}\{\widehat{V} \ne V \mid \zeta\} \ge 1 - \Omega\Big(\frac{n(\bar\kappa)}{\log\kappa}\Big)\,.$$

Taking expectation over ζ, we see that (3) holds also for randomized algorithms. Combined with (2), we see that $n(\bar\kappa) \ge \Omega(\log\kappa) = \Omega(\log\bar\kappa)$. □

3.3 Reductions and properties of the construction

Recall from Section 3.1 that each $V \in \mathcal{S}$ is of the form $V = \tilde{V} * \chi_\delta + \|\cdot\|^2/(2\kappa^{O(1)})$. In this section, we reduce the desired properties of $\mathcal{S}$, namely that $\mathcal{S}$ is well-separated and hard to identify via queries, to geometric properties of the potentials summarized in Proposition 3.11 below.

By increasing κ by a factor of at most two, which will not harm the final lower bound, we can assume that $\kappa = 2^N$ for some positive integer N. We also set $\delta := \kappa^{-5}$. Let B_N denote the set of binary strengths of length N. For each $b \in B_N$ and $\ell \in [N]$, we let $[b]_\ell := 0.00b_1 \dots b_\ell$ in binary representation, and set $[b] := [b]_N$.

Proposition 3.11 (geometric properties). *There are functions $\tilde{V}_b$, for $b \in B_N$, such that:*

(P0) *$\tilde{V}_b$ is convex and $\kappa^{O(1)}$-smooth on average at scale $\delta = \kappa^{-5}$, i.e., $\tilde{V}_b * \chi_\delta$ is $\kappa^{O(1)}$-smooth, and attains its minimum $V_b(0) = 0$ at zero.*

(P1) *The zero set $\mathcal{Z}_b := \{\tilde{V}_b = 0\}$ contains the $10^3\delta$-neighborhood of the set*

$$\tilde{\mathcal{Z}}_b := \{(x, \beta x) \in \mathbb{R}^2 \mid x \ge 0,\ [b] - 2^{-N} \le \beta \le [b] + 2^{-N}\}\,, \tag{4}$$

and is contained in the 1-neighbourhood of $\tilde{\mathcal{Z}}_b$.

(P2) *Moreover, for all $x, y \in \mathbb{R}^2$,*

$$\tilde{V}_b(x, y) \ge \kappa^4\,\big(\mathrm{dist}((x, y), \tilde{\mathcal{Z}}_b) - 1\big)_+\,.$$

(P3) *If b, b' coincide in the first ℓ bits then $\tilde{V}_b$ and $\tilde{V}_{b'}$ coincide in the set*

$$\Big\{(x, y) \in \mathbb{R}^2 \;\Big|\; x < \frac{1}{4}\,2^{-3N} \text{ or } |y - [b]_\ell\, x| > 100 \cdot 2^{-\ell} x\Big\}\,.$$

We check that these properties imply that Proposition 3.10 holds.

Proof of Proposition 3.10. Let $\mathcal{S}$ be the collection of potentials $V_b := \tilde{V}_b * \chi_\delta + \|\cdot\|^2/(2\kappa^{16})$ for $b \in B_N$, where $\{\tilde{V}_b : b \in B_N\}$ are the functions from Proposition 3.11. We now verify the four properties of Proposition 3.10.

Proof of 1. By (P0), we know that $\tilde{V}_b$ is convex, which implies that $\tilde{V}_b * \chi_\delta$ is also convex. Therefore, V_b is κ^{-16}-strongly convex. In addition, by (P0), $\tilde{V}_b * \chi_\delta$ is $\kappa^{O(1)}$-smooth, which means that V_b is $\kappa^{O(1)} + \kappa^{-16} \le \kappa^{O(1)}$-smooth.

Proof of 2. By construction, $|\mathcal{S}| = \kappa$.

Proof of 3. We now show that $\mathcal{S}$ is c-separated. For any string b, recall the definition of $\tilde{\mathcal{Z}}_b$ from (4). Define the set

$$\Omega_b \coloneqq \{(x, \beta x) \in \mathbb{R}^2 \mid x \geq 2^{-3N},\ [b] - 0.4 \cdot 2^{-N} \leq \beta \leq [b] + 0.4 \cdot 2^{-N}\}\,.$$

It is clear that $\{\Omega_b : b \in B_N\}$ is a family of disjoint sets. By (P1) we know that the zero set $\mathcal{Z}_b$ of $\tilde{V}_b$ contains a $10^3\delta$-neighborhood of $\tilde{\mathcal{Z}}_b$. Since $\Omega_b \subset \tilde{\mathcal{Z}}_b$, it follows that $\tilde{V}_b * \chi_\delta = 0$ on Ω_b.

Let $\tilde{\Omega}_b \coloneqq \{(x,y) \in \Omega_b : \|(x,y)\| \leq \kappa^8\}$. Note that the full set of points (x,y) with $\|(x,y)\| \leq \kappa^8$ has volume $\pi\kappa^{16}$, and Ω_b is a sector of the plane with arc $\Theta(2^{-N})$, minus a small set of points (specifically, the points in the sector with $x \leq 2^{-3N}$, which also means $y \leq O(2^{-3N})$). Therefore, the volume of $\tilde{\Omega}_b$ is $\Theta(\kappa^{16} \cdot 2^{-N}) = \Theta(\kappa^{15})$. In addition, all points $(x,y) \in \tilde{\Omega}$ have $V_b(x,y) = -\|(x,y)\|^2/(2\kappa^{16}) \geq -1/2$. Hence,

$$\int_{\Omega_b} \exp(-V_b) \geq \int_{\tilde{\Omega}_b} \exp(-V_b) \geq \Omega(\kappa^{15})\,. \tag{5}$$

Next, we bound the full integral of $\exp(-V_b)$ across $\mathbb{R}^d$ by splitting $\mathbb{R}^d$ into four regions $\mathbb{R}^d = \tilde{\mathcal{Z}}_b \cup \Psi_{1,b} \cup \Psi_{2,b} \cup \Psi_{3,b}$, defined as follows:

- $\Psi_{1,b} \coloneqq \{(x,y) \in \mathbb{R}^2 \setminus \tilde{\mathcal{Z}}_b : \mathrm{dist}((x,y), \tilde{\mathcal{Z}}_b) \leq 2,\ \|(x,y)\| \leq \kappa^9\}$.
- $\Psi_{2,b} \coloneqq \{(x,y) \in \mathbb{R}^2 \setminus (\tilde{\mathcal{Z}}_b \cup \Psi_{1,b}) : \|(x,y)\| \leq \kappa^9\}$.
- $\Psi_{3,b} = \mathbb{R}^2 \setminus (\tilde{\mathcal{Z}}_b \cup \Psi_{1,b} \cup \Psi_{2,b})$.

Note that all points $\Psi_{3,b}$ have norm at least κ^9. To show that most of the mass of P_{V_b} is concentrated on $\tilde{\mathcal{Z}}_b$, we must show that the integrals over $\Psi_{1,b}$, $\Psi_{2,b}$, and $\Psi_{3,b}$ are small. In a nutshell, the integral over $\Psi_{1,b}$ is small because the 2-neighborhood of $\tilde{\mathcal{Z}}_b$ is small (relative to the size of $\tilde{\mathcal{Z}}_b$ itself); the integral over $\Psi_{2,b}$ is small because $\tilde{V}_b$ increases rapidly outside $\tilde{\mathcal{Z}}_b$; and the integral over $\Psi_{3,b}$ is small because the Gaussian part of V_b is small over this region.

On these four regions, we have the following bounds. First, $\int_{\mathbb{R}^2} \exp(-\|\cdot\|^2/(2\kappa^{16})) = 2\pi\kappa^{16}$. Therefore, since the sector $\tilde{\mathcal{Z}}_b$ has arc $\Theta(2^{-N})$, by rotational symmetry

$$\int_{\tilde{\mathcal{Z}}_b} \exp(-V_b) \leq \int_{\tilde{\mathcal{Z}}_b} \exp\bigl(-\frac{\|\cdot\|^2}{2\kappa^{16}}\bigr) \leq O(2^{-N}) \int_{\mathbb{R}^2} \exp\bigl(-\frac{\|\cdot\|^2}{2\kappa^{16}}\bigr) \leq O(\kappa^{15})\,.$$

Note that $\Psi_{1,b}$ consists of two strips adjacent to $\tilde{\mathcal{Z}}_b$, where each strip has width 2 and length $O(\kappa^9)$, together with a piece of area $O(1)$ near the origin. Thus, $\mathrm{vol}(\Psi_{1,b}) \leq O(\kappa^9)$, yielding

$$\int_{\Psi_{1,b}} \exp(-V_b) \leq \mathrm{vol}(\Psi_{1,b}) \leq O(\kappa^9)\,.$$

Next, for $(x,y) \in \mathbb{R}^2$ such that $\mathrm{dist}((x,y), \tilde{\mathcal{Z}}_b) \geq 3/2$, by (P2) we have $\tilde{V}_b(x,y) \geq \kappa^4$. After mollification at scale $\delta \leq 1/2$, we conclude that $\tilde{V}_b * \chi_\delta \geq \kappa^4$ on $\Psi_{2,b}$. In addition, $\Psi_{2,b}$ is contained in the ball of radius κ^9, so the volume of $\Psi_{2,b}$ is at most $\pi\kappa^{18}$. Therefore,

$$\int_{\Psi_{2,b}} \exp(-V_b) \leq \pi\kappa^{18} \exp(-\kappa^4)\,.$$

Finally, all points in $\Psi_{3,b}$ have ℓ_2 norm at least κ^9, so

$$\int_{\Psi_{3,b}} \exp(-V_b) \leq \iint_{\|(x,y)\| \geq \kappa^9} \exp\bigl(-\frac{\|(x,y)\|^2}{2\kappa^{16}}\bigr) \leq O(\kappa^8)\exp\bigl(-\Omega(\kappa^2)\bigr)\,,$$

by standard Gaussian tail estimates. Therefore,

$$\int_{\mathbb{R}^2} \exp(-V_b) \le O\Big(\kappa^{15} + \kappa^9 + \exp\big(-\Omega(\kappa^4)\big) + \exp\big(-\Omega(\kappa^2)\big)\Big) \le O(\kappa^{15})\,. \tag{6}$$

Overall, (5) and (6) together imply that $P_{V_b}(\Omega_b) \ge \Omega(1)$, i.e., $\mathcal{S}$ is $\Omega(1)$-well-separated.

Proof of 4. Finally, we show that $\mathcal{S}$ is hard to identify via queries at scale $\delta = \kappa^{-\Theta(1)}$ with $\mathcal{I} = O(1)$. We consider b drawn uniformly at random from B_N.

First, however, we need to extend (P3) to V_b (i.e., taking into account the mollification at scale δ). We claim that if b, b' coincide in the first ℓ bits, then V_b and $V_{b'}$ coincide in the set

$$\big\{(x,y) \in \mathbb{R}^2 \bigm| x < \frac{1}{8}\, 2^{-3N} \text{ or } |y - [b]_\ell\, x| > 200 \cdot 2^{-\ell} x\big\}\,. \tag{7}$$

In light of (P3), it suffices to show that if (x,y) lies in this set and $\|(x',y') - (x,y)\| \le \delta$, then $x' < \frac{1}{4}\, 2^{-3N}$ or $|y' - [b]_\ell\, x'| > 100 \cdot 2^{-\ell}\, x'$. In other words, the δ-neighborhood of (7) is contained in the set in (P3). In the first case, $x' < \frac{1}{4}\, 2^{-3N}$ follows if $\delta < \frac{1}{8}\, 2^{-3N}$, but since $\delta = \kappa^{-5} = 2^{-5N}$ this holds for large κ. In the second case,

$$|y' - [b]_\ell\, x'| \ge |y - [b]_\ell\, x| - \delta - [b]_\ell\, \delta \ge 200 \cdot 2^{-\ell}\, x - 2\delta\,.$$

This is greater than $100 \cdot 2^{-\ell}\, x$ provided that $2\delta \le 100 \cdot 2^{-\ell}\, x$, but this follows because $\delta = 2^{-5N}$ and $x \ge \frac{1}{8}\, 2^{-3N}$ (as we are in the negation of the first case). In fact, by replacing δ with 2δ, the same argument shows that for all (x,y) lying in the set (7), we have $V_b(B_\delta(x,y)) = V_{b'}(B_\delta(x,y))$. Note also that (7) shows that it is useless to query any points (x,y) with $x < \frac{1}{8}\, 2^{-3N}$, so for the remainder of the proof we assume that the algorithm does not do so.

We now move to a stronger oracle model. Namely, given a query point $(x,y) \in \mathbb{R}^2$, let ℓ be the largest integer such that $|y - [b]_\ell\, x| \le 200 \cdot 2^{-\ell}\, x$. Then, the oracle outputs $\hat{\xi} := [b]_{\ell+1}$, i.e., the oracle reveals the first $\ell + 1$ bits of b. To see that this new oracle is indeed stronger, observe that we can simulate the previous oracle using the revealed bits $[b]_{\ell+1}$; namely, pick any bit string b' which is consistent, in the sense that $[b']_{\ell+1} = [b]_{\ell+1}$. Then, by the choice of ℓ, we have $|y - [b]_{\ell+1}\, x| > 200 \cdot 2^{-(\ell+1)}\, x$, so that $V_b(B_\delta(x,y)) = V_{b'}(B_\delta(x,y))$, and hence we can output $V_b(B_\delta(x,y))$ given knowledge of $[b]_{\ell+1}$. It therefore suffices to bound the mutual information $I(\hat{\xi}_{1:n}; b)$ where $\hat{\xi}_{1:n}$ denotes the output of the stronger oracle on a sequence of adaptive but deterministic queries $(x_1,y_1),\dots,(x_n,y_n)$.

We can then write

$$\begin{aligned} I(\hat{\xi}_{1:n}; b) &= \sum_{i=1}^n I(\hat{\xi}_i; b \mid \hat{\xi}_{1:i-1}) && (8)\\ &= \sum_{i=1}^n \{H(\hat{\xi}_i \mid \hat{\xi}_{1:i-1}) - H(\hat{\xi}_i, \mid \hat{\xi}_{1:i-1}, b)\} && (9)\\ &\le \sum_{i=1}^n H(\hat{\xi}_i \mid \hat{\xi}_{1:i-1})\,, && (10) \end{aligned}$$

where $H(\cdot \mid \cdot)$ denotes the conditional entropy. The first line follows from the chain rule for mutual information, the second line follows from definition of mutual information, and third line follows from non-negativity of conditional entropy. Thus, we are done if we can show that $H(\hat{\xi}_i \mid \hat{\xi}_{1:i-1}) \le O(1)$, for all $i \le c\,|\mathcal{S}|$.

Conditionally on any particular realization of $\hat{\xi}_{1:i-1}$, let ℓ_0 denote the number of bits of b revealed thus far and let $[b_0]_{\ell_0}$ denote the revealed bits. Clearly the bit string b is uniformly distributed on the set B'_N of bit strings b' with $[b']_{\ell_0} = [b_0]_{\ell_0}$. Also, since we have assumed that the algorithm's queries are deterministic given the past history, the next query point (x_i, y_i) is deterministic. Then, the conditional probability that $\ell \ge \ell_0$ bits are revealed by the next query is

$$\begin{aligned}
&\mathbb{P}\{200 \cdot 2^{-\ell}\, x_i < |y_i - [b]_\ell\, x_i| \le 200 \cdot 2^{-(\ell-1)}\, x_i \mid \hat{\xi}_{1:i-1}\} \\
&\qquad \le \mathbb{P}\Big\{\frac{y_i}{x_i} - 200 \cdot 2^{-(\ell-1)} \le [b]_\ell \le \frac{y_i}{x_i} + 200 \cdot 2^{-(\ell-1)} \Bigm| \hat{\xi}_{1:i-1}\Big\}\,.
\end{aligned}$$

This is the probability that a uniformly chosen element of B'_N belongs to an interval of length $\Theta(2^{-\ell})$. Since there are $2^{N-\ell_0}$ elements of B'_N, and $\Theta(2^{N-\ell})$ of them belong to any fixed interval of length $\Theta(2^{-\ell})$, we conclude that the above probability is $O(2^{-(\ell-\ell_0)})$.

We then have

$$H(\hat{\xi}_i \mid \hat{\xi}_{1:i-1}) \le \mathbb{E} \sum_{\ell \ge \ell_0} (\ell - \ell_0)\, O(2^{-(\ell-\ell_0)}) \le O(1)\,, \tag{11}$$

where the expectation is taken over ℓ_0 (which depends on the realization of $\hat{\xi}_{1:i-1}$). Substituting the above bound into (10), we conclude that $I(\xi_{1:n}; b) = O(n)$, which implies that $\mathcal{S}$ is indeed hard to identify via queries. □

3.4 Construction of the distributions

This section contains the proof of Proposition 3.11.

For integers $1 \le k \le N$, let $[b]_k$ be the number $0.00b_1 b_2 \dots b_k$ in binary representation, and let $[b]_k := [b] := [b]_N$ for $k \ge N$. Define

$$\phi_{k,b}(x, y) := \big(|y - [b]_k\, x| - (2^{-k}\, x + 2^{-(3N-k)})\big)_+\,. \tag{12}$$

We also write $\phi_k := \phi_{k,b}$ when b is clear from context. For $x \ge 0$, the function ϕ_k essentially measures the distance to the set

$$\{(x, [b]_k\, x + \xi_k) \in \mathbb{R}^2 : x \ge 0,\ |\xi_k| \le 2^{-k}\, x + 2^{-(3N-k)}\}\,.$$

Finally, we define the potential

$$\tilde{V}_b(x, y) := 2^{7N} \max_{k=1,\dots,N} 2^{-k} \phi_k(x, y)\,. \tag{13}$$

Proof of Proposition 3.11. We prove that the construction (13) satisfies each of the four properties in turn.

Proof of Property (P0). The convexity of $\tilde{V}_b$ follows because each ϕ_k is convex. To check that $\tilde{V}_b$ is $\kappa^{O(1)}$-smooth on average, using the compositionality of the maximum (i.e., $\max(a, \max(b, c)) = \max(a, b, c)$) we see that that $\tilde{V}_b$ can be written as a maximum of affine functions, each of slope $\kappa^{O(1)}$; hence, $\tilde{V}_b$ is $\kappa^{O(1)}$-Lipschitz. Differentiating under the integral,

$$\nabla(\tilde{V}_b * \chi_\delta)(x, y) = \iint_{B_\delta} \nabla \tilde{V}_b(x + u, y + v)\, \mathrm{d}u\, \mathrm{d}v = \iint \nabla \tilde{V}_b\, \mathbb{1}_{B_\delta(x,y)}\,,$$

where the expression makes sense because $\tilde{V}_b$ is Lipschitz and hence differentiable almost everywhere by Rademacher's theorem, and the absolute continuity of $\tilde{V}_b$ ensures the validity of the fundamental theorem of calculus. Then, by Hölder's inequality,

$$\begin{aligned}
\|\nabla(\tilde{V}_b * \chi_\delta)(x, y) - \nabla(\tilde{V}_b * \chi_\delta)(x', y')\| &\le \big(\sup \|\nabla \tilde{V}_b\|\big)\, \|\mathbb{1}_{B_\delta(x,y)} - \mathbb{1}_{B_\delta(x',y')}\|_{L^1} \\
&\le \kappa^{O(1)} \operatorname{vol}\big(B_\delta(x, y) \mathbin{\triangle} B_\delta(x', y')\big)\,.
\end{aligned}$$

By elementary considerations, the volume of the symmetric difference between the balls is bounded by $O(\kappa^{O(1)}\, \|(x, y) - (x', y')\|)$, and therefore $\nabla(\tilde{V}_b * \chi_\delta)$ is $\kappa^{O(1)}$-Lipschitz.

Finally, it is obvious that $\tilde{V}_b \ge 0$ and $\tilde{V}_b = 0$ at the origin.

Proof of Property (P1)**.** We only need to verify that any point (x, y) which is $10^3\delta$-close to $\tilde{\mathcal{Z}}_b$ satisfies $\tilde{V}_b(x, y) = 0$, as the second part of Property (P1) is automatically implied by Property (P2). For such a point (x, y), there exists (x', y') such that

$$x' \geq 0\,, \qquad |x' - x| \wedge |y' - y| \leq 10^3\delta\,, \qquad \text{and} \qquad |y' - [b]\,x'| \leq 2^{-N}x'\,.$$

This also implies $|y' - [b]_k\,x'| \leq 2^{-k}\,x'$ for all $1 \leq k \leq N$, since $|[b]_k - [b]| \leq 2^{-k} - 2^{-N}$. Therefore, for all $1 \leq k \leq N$, $|y - [b]_k\,x| \leq 2^{-k}\,(x + 10^3\delta) + 2 \cdot 10^3\delta \leq 2^{-k}x + 2^{-(3N-k)}$, since $\delta = 2^{-5N}$. By the definition (13) of $\tilde{V}_b$ and the definition of ϕ_k in (12), it follows that $\tilde{V}_b(x, y) = 0$.

Proof of Property (P2)**.** We just need to check that

$$2^{6N}\phi_N(x, y) \geq \kappa^4\,\big(\operatorname{dist}((x, y), \tilde{\mathcal{Z}}_b) - 1\big)_+\,,$$

or equivalently, $2^{2N}\phi_N(x, y) \geq (\operatorname{dist}((x, y), \tilde{\mathcal{Z}}_b) - 1)_+$. We first consider the case when $x \geq 0$, and we may assume that $(x, y) \notin \tilde{\mathcal{Z}}_b$ as otherwise the claim is obvious. If (x, y) has distance Δ to its closest point in $\tilde{\mathcal{Z}}_b$, then any y' such that $(x, y') \in \tilde{\mathcal{Z}}_b$ must satisfy $|y - y'| \geq \Delta$. Applying this to $y' = [b]\,x \pm 2^{-N}\,x$, we obtain

$$\operatorname{dist}\big((x, y), \tilde{\mathcal{Z}}_b\big) \leq |y - [b]\,x + 2^{-N}x| \wedge |y - [b]\,x - 2^{-N}\,x| = |y - [b]\,x| - 2^{-N}\,x\,.$$

In turn, it implies that $\phi_N(x, y) \geq (\operatorname{dist}((x, y), \tilde{\mathcal{Z}}_b) - 2^{-(3N-k)})_+ \geq (\operatorname{dist}((x, y), \tilde{\mathcal{Z}}_b) - 1)_+$.

If $x < 0$, then $\operatorname{dist}((x, y), \tilde{\mathcal{Z}}_b) \leq \|(x, y)\| \leq \sqrt{2}\,\max(|x|, |y|)$. Then, for N large,

$$\begin{aligned}
2^{2N}\phi_N(x, y) &= 2^{2N}\,\big(|y - [b]\,x| - 2^{-N}\,x - 2^{-(3N-k)}\big)_+ \\
&= 2^{N-1/2}\,\big(2^{N+1/2}\,|y - [b]\,x| + \sqrt{2}\,|x| - 2^{-(2N-k)+1/2}\big)_+ \\
&\geq 2^{N-1/2}\,\Big(2^{3/2}\,\max\big(0, |y| - \frac{1}{2}\,|x|\big) + \sqrt{2}\,|x| - 1\Big)_+ \\
&\geq 2^{N-1/2}\,\big(\sqrt{2}\,\max(|x|, |y|) - 1\big)_+ \geq \big(\operatorname{dist}((x, y), \tilde{\mathcal{Z}}_b) - 1\big)_+\,.
\end{aligned}$$

Proof of Property (P3). The last property follows from Proposition 3.12 below, because if b, b' agree on the first ℓ bits, then on the set in the statement of Property (P3),

$$\tilde{V}_b = 2^{7N}\max_{k=1,\dots,N} 2^{-k}\phi_{k,b} = 2^{7N}\max_{k=1,\dots,\ell} 2^{-k}\phi_{k,b} = 2^{7N}\max_{k=1,\dots,\ell} 2^{-k}\phi_{k,b'} = 2^{7N}\max_{k=1,\dots,N} 2^{-k}\phi_{k,b'} = \tilde{V}_{b'}\,.$$

The second and fourth equalities invoke Proposition 3.12, and the third equality uses the fact that $\phi_{k,b}$ only depends on b through $[b]_k$. This completes the proof. □

Proposition 3.12 (potentials agree if bits agree). *Let $S_\ell(b)$ be the set*

$$S_\ell(b) \coloneqq \Big\{(x, y) \in \mathbb{R}^2 : x < \frac{1}{4}\,2^{-3N} \text{ or } |y - [b]_\ell\,x| \geq 100 \cdot 2^{-\ell}x\Big\}\,.$$

Then, for $x, y \in S_\ell(b)$,

$$\max_{k=1,\dots,N} 2^{-k}\phi_k(x, y) = \max_{k=1,\dots,\ell} 2^{-k}\phi_k(x, y)\,.$$

In turn, Proposition 3.12 follows by induction from:

Proposition 3.13 (induction). *If $(x, y) \in S_\ell(b)$, and for some $k > \ell$ we have $\phi_k(x, y) > 0$, then $\phi_k(x, y) \leq 2\phi_{k-1}(x, y)$.*

Proof. First, we may assume that $x > 0$. This is because if $x \le 0$,

$$\begin{aligned}\phi_{k-1}(x,y) &\ge |y - [b]_k\, x| - |[b]_{k-1} - [b]_k|\, |x| - 2^{-(k-1)}\, x - 2^{-(3N-k+1)}\\ &\ge |y - [b]_k\, x| + 2^{-k} x - 2^{-(k-1)} x - 2^{-(3N-k+1)}\\ &= |y - [b]_k\, x| - 2^{-k} x - 2^{-(3N-k+1)}\\ &\ge \phi_k(x,y)\,,\end{aligned}$$

since we are assuming $\phi_k(x,y) > 0$.

Now, since $x > 0$, we start by estimating

$$\begin{aligned}\phi_{k-1}(x,y) &\ge |y - [b]_k\, x| - |[b]_{k-1} - [b]_k|\, x - 2^{-(k-1)} x - 2^{-(3N-k+1)}\\ &\ge |y - [b]_k\, x| - 3 \cdot 2^{-k} x - 2^{-(3N-k+1)}\\ &= \phi_k(x,y) - 2 \cdot 2^{-k} x + 2^{-(3N-k+1)}\end{aligned}$$

and

$$\phi_k(x,y) = |y - [b]_k\, x| - (2^{-k} x + 2^{-(3N-k)})\,.$$

First, suppose that $x \le \frac{1}{4}\, 2^{-3N}$. Then, $2^{-(3N-k+1)} \ge 2 \cdot 2^{-k} x$, so in fact $\phi_{k-1}(x,y) \ge \phi_k(x,y)$. Alternatively, if $x \ge \frac{1}{4}\, 2^{-3N}$ and $|y - [b]_\ell\, x| \ge 100 \cdot 2^{-\ell} x$, then

$$\begin{aligned}2\phi_{k-1}(x,y) &\ge 2\,|y - [b]_\ell\, x| - 2\,|[b]_\ell - [b]_{k-1}|\, x - 4 \cdot 2^{-k} x - 2^{-(3N-k)}\\ &\ge 2\,|y - [b]_\ell\, x| - 6 \cdot 2^{-\ell} x - 2^{-(3N-k)}\,,\\ \phi_k(x,y) &\le |y - [b]_\ell\, x| + |[b]_\ell - [b]_k|\, x - 2^{-k} x - 2^{-(3N-k)}\\ &\le |y - [b]_\ell\, x| + 2^{-\ell} x - 2^{-(3N-k)}\,.\end{aligned}$$

As a result, when $|y - [b]_\ell\, x| \ge 100 \cdot 2^{-\ell} x$, we see that $\phi_{k-1}(x,y) \ge \frac{1}{2}\, \phi_k(x,y)$. □

4 A lower bound for sampling from Gaussians via Wishart matrices

We define $W \sim \mathsf{Wishart}(d)$ to mean $W = XX^\mathsf{T}$ where each entry of $X \in \mathbb{R}^{d\times d}$ is $\mathcal{N}(0, \frac{1}{d})$. We aim to prove the following two theorems, which together imply a query complexity lower bound for sampling from Gaussians.

Theorem 4.1 (reducing inverse trace estimation to sampling). *Let $\delta > 0$. There is a universal constant $c > 0$ (depending only on δ) such that the following hold. Suppose that $d \ge c^{-1}$ and there exists a query algorithm such that, for any Gaussian target distribution $\pi := \mathcal{N}(0,\Sigma)$ in $\mathbb{R}^d$ with $cd^{-2}\, I_d \preceq \Sigma^{-1} \preceq c^{-1}\, I_d$, outputs a sample from a distribution $\widehat{\pi}$ such that either $\|\widehat{\pi} - \pi\|_{\mathrm{TV}} \le c$ or $\sqrt{cd^{-2}}\, W_2(\widehat{\pi}, \pi) \le c$, using n queries to π.*

Then, given $W \sim \mathsf{Wishart}(d)$, there exists an algorithm which makes at most $c^{-1} n$ matrix-vector queries to W and outputs an estimator $\widehat{\mathrm{tr}}$ such that $\frac{1}{2}\,\mathrm{tr}(W^{-1}) \le \widehat{\mathrm{tr}} \le 2\,\mathrm{tr}(W^{-1})$ with probability at least $1 - \delta$.

Theorem 4.2 (lower bound for inverse trace estimation). *Let $W \sim \mathsf{Wishart}(d)$ for $d \ge 2$. For any $C > 0$, there exists $\delta > 0$ (depending only on C) such that any algorithm which makes n matrix-vector queries to W and outputs an estimator $\widehat{\mathrm{tr}}$ such that $C^{-1}\,\mathrm{tr}(W^{-1}) \le \widehat{\mathrm{tr}} \le C\,\mathrm{tr}(W^{-1})$ with probability at least $1 - \delta$ must use $n \ge \Omega(d)$ queries.*

Remark 4.3. Suppose that we want to sample from a target distribution π which is α-strongly log-concave. It is straightforward to check that total variation guarantees are invariant under rescaling the target (replacing π with $S_\# \pi$, where $S : \mathbb{R}^d \to \mathbb{R}^d$ is the scaling map $Sx := \zeta x$ for some $\zeta > 0$), whereas Wasserstein guarantees are not. Instead, the scale-invariant quantity is $\sqrt{\alpha}\, W_2$, which is what appears in Theorem 4.1.

Consider the class of centered Gaussian distributions on $\mathbb{R}^d$ which are α-strongly log-concave and β-log-smooth; let $\kappa := \beta/\alpha$ denote the condition number. Let $\mathscr{C}_{\mathrm{G,d}}(\kappa, d, \varepsilon)$ denote the query complexity of outputting a sample which is ε-close in the metric d to a target distribution in this class, where d is one of the scale-invariant distances $\mathrm{d} \in \{\mathrm{TV}, \sqrt{\alpha}\, W_2\}$. Then, Theorems 4.1 and 4.2 (with $C = 2$ and δ, c being universal constants) show that for $d \ge c^{-1}$,

$$\mathscr{C}_{\mathrm{G,d}}(c^{-2}d^2, d, c) \ge \Omega(d)\,. \tag{14}$$

By embedding the construction into higher dimensions, we obtain the following corollary.

Corollary 4.4 (query lower bound via Wishart matrices). *For* $\mathrm{d} \in \{\mathrm{TV}, \sqrt{\alpha}\, W_2\}$, *there is a universal constant* $c > 0$ *such that*

$$\mathscr{C}_{\mathrm{G,d}}(\kappa, d, c) \ge \Omega\big(\sqrt{\kappa} \wedge d\big)\,.$$

Proof. If $\kappa \ge c^{-2}d^2$, then (14) yields

$$\mathscr{C}_{\mathrm{G,d}}(\kappa, d, c) \ge \Omega(d) \ge \Omega\big(\sqrt{\kappa} \wedge d\big)\,.$$

Otherwise, if $\kappa \le c^{-2}d^2$, let $d_\star$ be the largest integer such that $\kappa \ge c^{-2}d_\star^2$. Then, by embedding the $d_\star$-dimensional construction into dimension d,

$$\mathscr{C}_{\mathrm{G,d}}(\kappa, d, c) \ge \mathscr{C}_{\mathrm{G,d}}(\kappa, d_\star, c) \ge \Omega(d_\star) \ge \Omega\big(\sqrt{\kappa} \wedge d\big)\,,$$

which concludes the proof. □

4.1 Reducing inverse trace estimation to sampling

In this section, we prove Theorem 4.1, which is based on the concentration of the squared norm of a Gaussian. We recall the following identity:

Lemma 4.5 (concentration of the squared norm). *Let* $Z \sim \mathcal{N}(0, \Sigma)$. *Then,*

$$\operatorname{var}(\|Z\|^2) = 2\,\|\Sigma\|_{\mathrm{HS}}^2\,,$$

where $\|\Sigma\|_{HS} = \sqrt{\sum_{i,j} \Sigma_{i,j}^2}$ *is the* Hilbert-Schmidt norm *(equivalently, the* Frobenius *norm) of* Σ.

Proof. Note that since all quantities are rotationally invariant, we may assume without loss of generality that Σ is diagonal. Then the equality claimed is just the variance of a non-homogenous chi-squared random variable. □

We now prove Theorem 4.1.

Proof of Theorem 4.1. Let $W \sim \mathrm{Wishart}(d)$ and let $\Sigma := W^{-1}$. By Proposition 4.8, there exists $c > 0$ (depending only on δ) such that with probability at least $1 - \delta/3$, it holds that

$$c d^{-2}\, I_d \preceq \Sigma^{-1} \preceq c^{-1}\, I_d\,.$$

We work on the event $\mathcal{E}$ that this holds.

Case 1: total variation distance. From Lemma 4.5 and Chebyshev's inequality, we deduce that if $Z_1, \dots, Z_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$ and $\widehat{\mathrm{tr}}_\star := m^{-1} \sum_{i=1}^m \|Z_i\|^2$,

$$\mathbb{P}\Big\{\big|\widehat{\mathrm{tr}}_\star - \operatorname{tr}\Sigma\big| \ge \frac{1}{2}\operatorname{tr}\Sigma\Big\} \le \frac{\operatorname{var}\widehat{\mathrm{tr}}_\star}{(\operatorname{tr}\Sigma)^2/4} = \frac{8}{m} \cdot \frac{\operatorname{tr}(\Sigma^2)}{\operatorname{tr}(\Sigma)^2} \le \frac{8}{m}\,.$$

Take $m \ge 48/\delta$ so that this probability is at most $\delta/3$. Conditionally on W, let $\widehat{\pi}_W$ denote the law of the sample X of the algorithm when run on the target $\mathcal{N}(0, \Sigma)$. By running the sampling algorithm m times, we can obtain i.i.d. samples $X_1, \dots, X_m \overset{\text{i.i.d.}}{\sim} \widehat{\pi}_W$. Then, for $\widehat{\operatorname{tr}} := m^{-1} \sum_{i=1}^m \|X_i\|^2$,

$$\begin{aligned}
\mathbb{P}\Bigl\{\bigl|\widehat{\operatorname{tr}} - \operatorname{tr}\Sigma\bigr| \ge \frac{1}{2}\operatorname{tr}\Sigma\Bigr\} &\le \mathbb{P}(\mathcal{E}^{\mathrm{c}}) + \mathbb{P}\Bigl\{\bigl|\widehat{\operatorname{tr}} - \operatorname{tr}\Sigma\bigr| \ge \frac{1}{2}\operatorname{tr}\Sigma,\ \mathcal{E}\Bigr\} \\
&\le \frac{\delta}{3} + \mathbb{E}\Bigl[\mathbb{P}\Bigl\{\bigl|\widehat{\operatorname{tr}}_\star - \operatorname{tr}\Sigma\bigr| \ge \frac{1}{2}\operatorname{tr}\Sigma \Bigm| W\Bigr\}\,\mathbb{1}_{\mathcal{E}}\Bigr] + \mathbb{E}\bigl[\|\widehat{\pi}_W^{\otimes m} - \mathcal{N}(0,\Sigma)^{\otimes m}\|_{\mathrm{TV}}\,\mathbb{1}_{\mathcal{E}}\bigr] \\
&\le \frac{\delta}{3} + \frac{\delta}{3} + cm\,.
\end{aligned}$$

If we choose $c \le \delta/(3m)$, then $\widehat{\operatorname{tr}}$ is an estimator of $\operatorname{tr}(W^{-1})$ with multiplicative error at most 2 which succeeds with probability at least $1 - \delta$. Note that both c and m depend only on δ.

Case 2: Wasserstein distance. Consider a coupling of X and Z such that, conditionally on W, we have $\mathbb{E}[\|X - Z\|^2 \mid W] = \mathbb{E}[W_2^2(\widehat{\pi}_W, \mathcal{N}(0,\Sigma)) \mid W]$. Let $(X_1, Z_1), \dots, (X_m, Z_m)$ be i.i.d. copies of this coupling. Also, let $\mathcal{E}'$ denote the event that $\lambda_{\min}(W^{-1}) \ge \bar{c}d^2$, where $\bar{c}$ is a constant depending only on δ, chosen so that $\mathbb{P}(\mathcal{E}'^{\mathrm{c}}) \le \delta/3$ using Proposition 4.8. Then, conditionally on W in the event $\mathcal{E} \cap \mathcal{E}'$,

$$\begin{aligned}
\mathbb{E}\bigl[|\widehat{\operatorname{tr}} - \operatorname{tr}\Sigma| \bigm| W\bigr] &\le \mathbb{E}\bigl[|\widehat{\operatorname{tr}} - \widehat{\operatorname{tr}}_\star| \bigm| W\bigr] + \mathbb{E}\bigl[|\widehat{\operatorname{tr}}_\star - \operatorname{tr}\Sigma| \bigm| W\bigr] \\
&\le \mathbb{E}\bigl[|\widehat{\operatorname{tr}} - \widehat{\operatorname{tr}}_\star| \bigm| W\bigr] + \frac{2\operatorname{tr}\Sigma}{\sqrt{m}}\,,
\end{aligned}$$

where we used Lemma 4.5. Using $\|x\|^2 - \|y\|^2 = \langle x - y, x + y\rangle$, for any $\lambda > 0$,

$$\begin{aligned}
\mathbb{E}\bigl[|\widehat{\operatorname{tr}} - \widehat{\operatorname{tr}}_\star| \bigm| W\bigr] &\le \mathbb{E}\bigl[|\,\|X\|^2 - \|Z\|^2\,| \bigm| W\bigr] \le \mathbb{E}\bigl[\|X - Z\|^2 \bigm| W\bigr] + 2\,\mathbb{E}\bigl[|\langle X - Z, Z\rangle| \bigm| W\bigr] \\
&\le (1+\lambda)\,\mathbb{E}\bigl[\|X - Z\|^2 \bigm| W\bigr] + \frac{1}{\lambda}\,\mathbb{E}\bigl[\|Z\|^2 \bigm| W\bigr] \\
&\le (1+\lambda)\,c^3 d^2 + \frac{\operatorname{tr}\Sigma}{\lambda} \le (1+\lambda)\,\frac{c^3}{\bar{c}}\operatorname{tr}\Sigma + \frac{\operatorname{tr}\Sigma}{\lambda}\,.
\end{aligned}$$

If we take $\lambda = 18/\delta$, $m \ge (36/\delta)^2$, and if c is sufficiently small (depending only on δ), we obtain

$$\mathbb{E}\bigl[|\widehat{\operatorname{tr}} - \operatorname{tr}\Sigma| \bigm| W\bigr] \le \frac{\delta\operatorname{tr}\Sigma}{6}\,.$$

By Markov's inequality,

$$\begin{aligned}
\mathbb{P}\Bigl\{\bigl|\widehat{\operatorname{tr}} - \operatorname{tr}\Sigma\bigr| \ge \frac{1}{2}\operatorname{tr}\Sigma\Bigr\} &\le \mathbb{P}(\mathcal{E}^{\mathrm{c}}) + \mathbb{P}(\mathcal{E}'^{\mathrm{c}}) + \mathbb{E}\Bigl[\mathbb{P}\Bigl\{\bigl|\widehat{\operatorname{tr}} - \operatorname{tr}\Sigma\bigr| \ge \frac{1}{2}\operatorname{tr}\Sigma \Bigm| W\Bigr\}\,\mathbb{1}_{\mathcal{E}\cap\mathcal{E}'}\Bigr] \\
&\le \frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3} \le \delta\,.
\end{aligned}$$

We conclude as before. □

4.2 Lower bound for inverse trace estimation

In this section, we prove Theorem 4.2. The idea is that due to the heavy tails of $\lambda_{\min}(W^{-1})$ implied by Proposition 4.8, with some small probability δ, $\operatorname{tr}(W^{-1})$ will be very large. An algorithm for inverse trace estimation which succeeds with probability at least $1 - \delta$ must be able to detect this event, and we show that this requires making $\Omega(d)$ queries.

The key technical tools are the following propositions, due to [Braverman et al. 2020].

Proposition 4.6 ([Braverman et al. 2020, Lemma 3.4]). *Let $W \sim \mathsf{Wishart}(d)$. Then, for any sequence of $n < d$ (possibly adaptive) queries $v_1, \ldots, v_n$ and responses $w_1 = Wv_1, \ldots, w_n = Wv_n$, there exists an orthogonal matrix $V \in \mathbb{R}^{d\times d}$ and matrices $Y_1 \in \mathbb{R}^{n\times n}$, $Y_2 \in \mathbb{R}^{(d-n)\times n}$ that only depend on $v_1, \ldots, v_n, w_1, \ldots, w_n$, such that VWV^T has the block form*

$$VWV^\mathsf{T} = \begin{bmatrix} Y_1Y_1^\mathsf{T} & Y_1Y_2^\mathsf{T} \\ Y_2Y_1^\mathsf{T} & Y_2Y_2^\mathsf{T} + \widetilde{W} \end{bmatrix}.$$

Here, conditionally on $v_1, \ldots, v_n, w_1, \ldots, w_n$, the matrix $\widetilde{W}$ has the $\mathsf{Wishart}(d-n)$ distribution.

Proposition 4.7 ([Braverman et al. 2020, Lemma 3.5]). *For any matrices $Y_1 \in \mathbb{R}^{n\times n}$, $Y_2 \in \mathbb{R}^{(d-n)\times n}$, and any symmetric matrix $\widetilde{W} \in \mathbb{R}^{(d-n)\times(d-n)}$, it holds that*

$$\lambda_{\min}\Bigl(\begin{bmatrix} Y_1Y_1^\mathsf{T} & Y_1Y_2^\mathsf{T} \\ Y_2Y_1^\mathsf{T} & Y_2Y_2^\mathsf{T} + \widetilde{W} \end{bmatrix}\Bigr) \le \lambda_{\min}(\widetilde{W}).$$

We are now ready to prove Theorem 4.2. Note that this result is very similar to that of [Braverman et al. 2020], except that we work with the inverse trace rather than the minimum eigenvalue.

Proof of Theorem 4.2. Let $\delta > 0$ be chosen later. We first argue that $\widehat{\mathrm{tr}}$ must not be too large. Applying Proposition 4.9, we conclude that there is a universal constant $C' > 0$ such that $\mathrm{tr}(W^{-1}) \le C'd^2$ with probability at least $1/2$. Hence,

$$\begin{aligned} \mathbb{P}\bigl\{\widehat{\mathrm{tr}} \le CC'd^2\bigr\} &\ge \mathbb{P}\bigl\{\mathrm{tr}(W^{-1}) \le C'd^2 \text{ and } \widehat{\mathrm{tr}} \le C\,\mathrm{tr}(W^{-1})\bigr\} \\ &\ge \mathbb{P}\{\mathrm{tr}(W^{-1}) \le C'd^2\} - \mathbb{P}\{\widehat{\mathrm{tr}} > C\,\mathrm{tr}(W^{-1})\} \ge \frac{1}{2} - \delta\,. \end{aligned}$$

Next, suppose for the sake of contradiction that $n \le d/2$. Let $\mathscr{F}_n$ denote the σ-algebra generated by the information available to the algorithm up to iteration n, that is, the queries $v_1, \ldots, v_n$, the responses $w_1, \ldots, w_n$, and any external randomness used by the algorithm (which is independent of W). Applying Propositions 4.6 and 4.7,

$$\begin{aligned} \mathbb{P}\bigl\{\widehat{\mathrm{tr}} < C^{-1}\,\mathrm{tr}(W^{-1})\bigr\} &\ge \mathbb{P}\bigl\{\widehat{\mathrm{tr}} \le CC'd^2 \text{ and } \lambda_{\min}(W^{-1}) > C^2C'd^2\bigr\} \\ &\ge \mathbb{P}\bigl\{\widehat{\mathrm{tr}} \le CC'd^2 \text{ and } \lambda_{\min}(\widetilde{W}^{-1}) > C^2C'\,d^2\bigr\} \\ &= \mathbb{E}\bigl[\mathbb{1}\{\widehat{\mathrm{tr}} \le CC'd^2\}\,\mathbb{P}\{\lambda_{\min}(\widetilde{W}^{-1}) \ge C^2C'd^2 \mid \mathscr{F}_n\}\bigr]\,. \end{aligned}$$

According to Proposition 4.6, conditionally on $\mathscr{F}_n$, $\widetilde{W}$ has the $\mathsf{Wishart}(d-n)$ distribution. By applying Proposition 4.8,

$$\mathbb{P}\{\lambda_{\min}(\widetilde{W}^{-1}) \ge C^2C'd^2 \mid \mathscr{F}_n\} \ge \mathbb{P}\{\lambda_{\min}(\widetilde{W}^{-1}) \ge 4C^2C'\,(d-n)^2 \mid \mathscr{F}_n\} \gtrsim \frac{1}{C\sqrt{C'}}\,.$$

Therefore,

$$\mathbb{P}\bigl\{\widehat{\mathrm{tr}} < C^{-1}\,\mathrm{tr}(W^{-1})\bigr\} \gtrsim \mathbb{P}\bigl\{\widehat{\mathrm{tr}} \le CC'd^2\bigr\}\,\frac{1}{C\sqrt{C'}} \ge \frac{1/2-\delta}{C\sqrt{C'}}\,,$$

which is larger than δ provided that δ is chosen sufficiently small (depending only on C). This contradicts the success probability of the algorithm, and hence we deduce that $n \ge d/2$. □

4.3 Useful facts about Wishart matrices

We collect together useful facts about Wishart matrices which are used in the proofs.

Proposition 4.8 (extreme singular values of a Gaussian matrix). *Let $W \sim \mathsf{Wishart}(d)$. For any $x \in [0,1]$,*

$$\mathbb{P}\Big\{\lambda_{\min}(W) \le \frac{x}{d^2}\Big\} \asymp \sqrt{x}\,.$$

Also, there is a universal constant $C > 0$ such that

$$\mathbb{P}\{\lambda_{\max}(W) \ge C\,(1+t)\} \le 2\exp(-dt)\,.$$

Proof. See, e.g., [Edelman 1989, Theorem 5.1] and [Vershynin 2018, Theorem 4.4.5]. □

Proposition 4.9 (bound on the inverse trace). *Let $W \sim \mathsf{Wishart}(d)$. Then, for any $\delta > 0$, with probability at least $1-\delta$, it holds that $\operatorname{tr}(W^{-1}) \le C_\delta d^2$ where C_δ is a constant depending only on δ.*

Proof. According to [Szarek 1991, Theorem 1.2], there is a universal constant $C > 0$ such that for each $j = 1, \dots, d$ and $\alpha \ge 0$,

$$\mathbb{P}\Big\{\frac{1}{\lambda_j(W)} \ge \frac{d^2}{\alpha^2 j^2}\Big\} \le (C\alpha)^{j^2}\,.$$

Let $\alpha < 1/C$ and let $E_\alpha \coloneqq \{1/\lambda_j(W) \ge d^2/(\alpha^2 j^2)$ for some $j = 1, \dots, d\}$. By the union bound,

$$\mathbb{P}(E_\alpha) \le \sum_{j=1}^d (C\alpha)^{j^2} \lesssim \frac{1}{\sqrt{\log(1/(C\alpha))}}\,.$$

On the event E_α^{c},

$$\operatorname{tr}(W^{-1}) \le \sum_{j=1}^d \frac{d^2}{\alpha^2 j^2} = \frac{\pi^2 d^2}{6\alpha^2}\,,$$

which is the claimed result upon taking α sufficiently small. □

Remark 4.10. The proof only shows that $\mathbb{P}\{\operatorname{tr}(W^{-1}) \ge \eta d^2\} \lesssim 1/\sqrt{\log \eta}$ for $\eta \gg 1$, which is not enough to conclude that $\mathbb{E}\operatorname{tr}(W^{-1})$ is finite. In fact, it holds that $\mathbb{E}\operatorname{tr}(W^{-1}) = \infty$, which can already be seen from Proposition 4.8.

5 A lower bound for sampling from Gaussians via reduction to block Krylov

In this section, we prove Theorem 1.3. Our proof procedes in two parts: we first show a lower bound against the block Krylov method, and then a reduction showing that an arbitrary adaptive algorithm can be simulated via a block Krylov method.

5.1 Preliminaries

We first record some important facts that we will use later on. The following is a standard approximation-theoretic result:

Proposition 5.1 ([Sachdeva and Vishnoi 2014, Proposition 2.4, rephrased]). *Let T_K be the degree-K Chebyshev polynomial, and let $1 = \beta_1 > \cdots > \beta_{K+1} = -1$ be the set of real values β such that $T_K(\beta) \in \{-1, 1\}$. Then, for any real degree-K polynomial p such that $|p(\beta_i)| \le 1$ for all β_i, we have $|p(x)| \le |T_K(x)| \le (|x| + \sqrt{x^2-1})^K$ for all $|x| > 1$.*

Let $c_0 > 0$ be a constant to be chosen later. The above proposition immediately implies:

Corollary 5.2 (approximation error). *Suppose that $K \le c_0\sqrt{\kappa}\log d$. Then, there exist $\kappa = \lambda_1 > \cdots > \lambda_{K+2} = 1$ (that only depend on K and κ) such that for any real degree-K polynomial P, $\max_{1\le i\le K+2} |\frac{1}{\lambda_i} - P(\lambda_i)| \ge d^{-2c_0 - O(1/\sqrt{\kappa})}/\kappa$.*

Proof. Set $\beta_1, \dots, \beta_{K+2}$ to be the solutions of $T_{K+1} \in \{-1, 1\}$, and for each $1 \le i \le K{+}2$, set $\lambda_i := \frac{(\kappa-1)}{2}\,(\beta_i{+}1){+}1$; by construction, $\kappa = \lambda_1 > \cdots > \lambda_{K+2} = 1$. Given any polynomial Q of degree at most $K+1$, note that if $|Q(\lambda_i)| \le 1$ for all i, then the polynomial p given by $p(x) := Q(\frac{\kappa-1}{2}\,(x+1)+1)$ satisfies $|p(\beta_i)| \le 1$ for all i. By Proposition 5.1, for $x_0 := -(1 + \frac{2}{\kappa-1})$,

$$|Q(0)| = |p(x_0)| \le \left(|x_0| + \sqrt{x_0^2 - 1}\right)^{K+1} \le \left(1 + \frac{2}{\sqrt{\kappa}} + O\Big(\frac{1}{\kappa}\Big)\right)^{K+1}$$

$$< \exp\Big(\Big(\frac{2}{\sqrt{\kappa}} + O\Big(\frac{1}{\kappa}\Big)\Big)\,\big(c_0\sqrt{\kappa}\log d + 1\big)\Big) = d^{2c_0 + O(1/\sqrt{\kappa})}\,.$$

Next, for a degree-K polynomial P, consider $Q(x) := d^{2c_0+O(1/\sqrt{\kappa})}\,(1 - xP(x))$. Note that Q has degree $K+1$ and $|Q(0)| = d^{2c_0+O(1/\sqrt{\kappa})}$, which implies that $|Q(\lambda_i)| > 1$ for some i, which in turn shows that $|\frac{1}{\lambda_i} - P(\lambda_i)| \ge d^{-2c_0-O(1/\sqrt{\kappa})}/\kappa$. □

We also introduce random matrix ensembles that are used in the proof, together with basic facts and properties.

Interestingly, as in the previous section, Wishart matrices are also useful for understanding block Krylov algorithms, but for a completely different reason. This time, we will study inner products between random vectors, which is also captured by a Wishart matrix. We denote by $\mathsf{Wishart}(K, N)$ the law of the random matrix $XX^\top \in \mathbb{R}^{K\times K}$, where the entries of $X \in \mathbb{R}^{K\times N}$ are i.i.d. *standard* Gaussians. Note that this is a different convention from the previous section, in which each entry of X was i.i.d. $\mathcal{N}(0, \frac{1}{d})$.

We also define the Gaussian orthogonal ensemble (GOE) of size K, denoted $\mathsf{GOE}(K)$. This is the law of a random symmetric matrix $G \in \mathbb{R}^{K\times K}$ where each diagonal entry $G_{i,i}$ is distributed as $\mathcal{N}(0, 1)$, and each off-diagonal entry $G_{i,j} = G_{j,i}$ is distributed as $\mathcal{N}(0, \frac{1}{2})$. Also, the entries $\{G_{i,j} : 1 \le i \le K,\ j \le i\}$ are independent.

A long line of work (see, e.g., [Brennan et al. 2021; Bubeck et al. 2016; Bubeck and Ganguly 2018; Jiang and Li 2015; Mikulincer 2022; Rácz and Richey 2019]) shows that when $N \gg K^3$, the Wishart ensemble is well-approximated by a scaled and shifted GOE, a fact which we shall invoke in the sequel.

Lemma 5.3 (equivalence of Wishart and GOE). *Let $W \sim \mathsf{Wishart}(K, N)$ be drawn from the Wishart distribution, and let W_0 be drawn from the distribution of symmetric matrices where the diagonal and above-diagonal entries are mutually independent, each diagonal entry is drawn as $\mathcal{N}(N, 2N)$, and each above-diagonal entry is drawn as $\mathcal{N}(0, N)$. (Equivalently, we can write $W_0 = NI + \sqrt{2N}\,G$, where $G \sim \mathsf{GOE}(K)$.) Then,*

$$\|\mathrm{law}(W) - \mathrm{law}(W_0)\|_{\mathrm{TV}} \le O\Big(\frac{K^{3/2}}{N^{1/2}}\Big)\,.$$

Finally, we also require the following basic linear algebraic fact:

Proposition 5.4 (rotating the right singular vectors). *Let $V, V' \in \mathbb{R}^{K\times N}$ be such that $VV^\top = (V')(V')^\top$. Then, there exists an orthogonal matrix $U \in \mathbb{R}^{N\times N}$ such that $VU = V'$.*

5.2 Lower bound against block Krylov algorithms

We start with the following proposition, which will be useful in establishing the existence of matrices with different inverse traces but which generate similar power method iterates.

Proposition 5.5 (polynomial approximation and duality). *Suppose that $K \le c_0\sqrt{\kappa}\log d$. Then, there exist $\kappa = \lambda_1 > \lambda_2 > \cdots > \lambda_{K+2} = 1$ and non-negative real numbers $x_1, \dots, x_{K+2}; x'_1, \dots, x'_{K+2}$, such that:*

(1) For all $0 \le j \le K$, $\sum_{i=1}^{K+2} x_i \lambda_i^j = \sum_{i=1}^{K+2} x_i' \lambda_i^j$.
(2) $\sum_{i=1}^{K+2} x_i = \sum_{i=1}^{K+2} x_i' = d$.
(3) $\sum_{i=1}^{K+2} x_i/\lambda_i - \sum_{i=1}^{K+2} x_i'/\lambda_i \ge 2d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa$.

Proof. If we fix the values of the λ_i to be the choices in Corollary 5.2, this becomes a linear program in the variables $\{x_i\}_{i=1}^{K+2}, \{x_i'\}_{i=1}^{K+2}$. By writing $\mathbf{x} = (x_1, \dots, x_{K+2}, x_1', \dots, x_{K+2}')$, our goal is to maximize $\mathbf{c}^\top \mathbf{x}$ over $\mathbf{x} \ge 0$ subject to $A\mathbf{x} = \mathbf{b}$. In our case, we set

$$\mathbf{c} := \begin{bmatrix} \lambda_1^{-1} \\ \vdots \\ \lambda_{K+2}^{-1} \\ -\lambda_1^{-1} \\ \vdots \\ -\lambda_{K+2}^{-1} \end{bmatrix}, \qquad A := \begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & -1 & \cdots & -1 \\ \lambda_1 & \cdots & \lambda_{K+2} & -\lambda_1 & \cdots & -\lambda_{K+2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \lambda_1^K & \cdots & \lambda_{K+2}^K & -\lambda_1^K & \cdots & -\lambda_{K+2}^K \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 2d \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

We can consider the dual linear program, and by strong duality this maximization is equivalent to minimizing $\mathbf{b}^\top \mathbf{y}$ over $\mathbf{y}$ such that $A^\top \mathbf{y} \ge \mathbf{c}$. By writing $\mathbf{y} = (z, y_0, y_1, \dots, y_K)$, this means we wish to minimize $2dz$ subject to $z + (y_0 + y_1\lambda_i + \cdots + y_K\lambda_i^K) \ge \frac{1}{\lambda_i}$ and $z - (y_0 + y_1\lambda_i + \cdots + y_K\lambda_i^K) \ge -\frac{1}{\lambda_i}$ for all $1 \le i \le K+2$. Equivalently, we wish to minimize $2dz$ subject to the existence of a polynomial P of degree at most K (with coefficients $y_0, \dots, y_K$) such that $z \ge |\frac{1}{\lambda_i} - P(\lambda_i)|$ for all $i \le K+2$.

The minimum for the dual linear program (and thus the maximum for the primal linear program), is $2d \inf_{P \in \mathcal{P}_K} \max_{1 \le i \le K+2} |\frac{1}{\lambda_i} - P(\lambda_i)|$, where $\mathcal{P}_K$ is the set of polynomials of degree at most K with real coefficients. By Corollary 5.2, this quantity is at least $2d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa$. □

We note that a slightly strengthened version of Proposition 5.5 holds. Let $0 < c_1 < 1$.

Corollary 5.6 (existence of good solutions). *Proposition 5.5 holds, where we also ensure that each* x_i *and* x_i' *is at least* $\frac{d}{2(K+2)}$ *and* $\frac{|x_i - x_i'|}{x_i} \le \frac{2c_1}{1-c_1}$, *though the right-hand side of the third condition becomes* $c_1 d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa$.

Proof. First, replace every x_i with $\frac{1}{2}(x_i + \frac{d}{K+2})$ and x_i' with $\frac{1}{2}(x_i' + \frac{d}{K+2})$. Then, we have that the replaced x_i, x_i' are at least $\frac{d}{2(K+2)}$, and the remaining statements in Proposition 5.5 hold, except the third which has the right-hand side replaced with $d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa$.

Next, replace every x_i with $\tilde{x}_i := \frac{1+c_1}{2} x_i + \frac{1-c_1}{2} x_i'$, and every x_i' with $\tilde{x}_i' := \frac{1+c_1}{2} x_i' + \frac{1-c_1}{2} x_i$. We still have that every $\tilde{x}_i, \tilde{x}_i'$ is at least $\frac{d}{2(K+2)}$, the first two conditions still hold, and the right-hand side of third condition is now $c_1 d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa$. Finally, note that $|\tilde{x}_i - \tilde{x}_i'| \le c_1 |x_i - x_i'|$, whereas $\tilde{x}_i \ge \frac{1-c_1}{2}(x_i + x_i')$. This implies that $\frac{|\tilde{x}_i - \tilde{x}_i'|}{\tilde{x}_i} \le \frac{2c_1}{1-c_1}$. □

We now have the necessary tools to prove our lower bound against block Krylov algorithms. Before doing so, we establish that there exist diagonal matrices D, D' which have substantially different inverse traces, but block Krylov algorithms cannot distinguish between them. To prove our actual lower bound, we show the same claim holds even if D, D' are randomly rotated, and the inverse trace difference is enough for a single sample to distinguish between them.

Lemma 5.7 (construction of diagonal matrices). *Suppose that* $K \le c_0\sqrt{\kappa}\log d$ *and* $K \le O(d)$. *Then, there exist diagonal matrices* $D, D' \in \mathbb{R}^{d \times d}$ *with all diagonal entries between* 1 *and* κ *with the following properties.*

(1) $|\operatorname{tr}(D^{-1}) - \operatorname{tr}(D'^{-1})| \ge c_1 d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa - 2(K+2)$.

(2) Consider sampling K d-dimensional random vectors $v^{(1)}, \dots, v^{(K)} \overset{i.i.d.}{\sim} \mathcal{N}(0, I_d)$. Then, the distributions of $\{\langle v^{(k)}, D^j\, v^{(\ell)}\rangle\}_{j \le K+2;\, k,\ell \le K}$ and $\{\langle v^{(k)}, D'^j\, v^{(\ell)}\rangle\}_{j \le K+2; k,\ell \le K}$ differ in total variation distance by at most $O(c_1 K^3 + K^3/d^{1/2})$.

Proof. Choose $\{x_i\}_{i=1}^{K+2}$, $\{x_i'\}_{i=1}^{K+2}$, and $\{\lambda_i\}_{i=1}^{K=2}$ satisfying Corollary 5.6. Define integers $\{N_i\}_{i=1}^{K+2}$ such that each N_i is either $\lfloor x_i \rfloor$ or $\lceil x_i \rceil$ and $\sum_{i=1}^{K+2} N_i = d$; define $\{N_i'\}_{i=1}^{K+2}$ similarly in terms of $\{x_i'\}_{i=1}^{K+2}$. We let D, D' be diagonal matrices such that for all i, D has N_i diagonal entries equal λ_i, and D' has N_i' diagonal entries equal to λ_i. Now, let $v^{(1)}, \dots, v^{(K)} \in \mathbb{R}^d$ be K random vectors drawn i.i.d. from $\mathcal{N}(0, I_d)$ (and define $v^{(1)\prime}, \dots, v^{(K)\prime}$ similarly). For $1 \le i \le K+2$, we define $v^{(k,i)}$ to be the projection of $v^{(k)}$ onto the dimensions corresponding to the diagonal entry λ_i for D. Note that $\{v^{(k,i)}\}_{i \le K+2,\, k \le K}$ are independent, and $v^{(k,i)} \sim \mathcal{N}(0, I_{N_i})$. Likewise, define $\{v^{(k,i)\prime}\}_{i \le K+2,\, k \le K}$ accordingly in terms of D'.

Note that $\operatorname{tr}(D^{-1}) - \operatorname{tr}(D'^{-1}) = \sum_{i=1}^{K+2} N_i/\lambda_i - \sum_{i=1}^{K+2} N_i'/\lambda_i$. Since $|N_i - x_i|, |N_i' - x_i'| \le 1$, and since each $\lambda_i \ge 1$, it implies

$$\operatorname{tr}(D^{-1}) - \operatorname{tr}(D'^{-1}) \ge \frac{c_1\, d^{1-2c_0 - O(1/\sqrt{\kappa})}}{\kappa} - 2\,(K+2)\,.$$

Next, we let $W^{(i)}$ represent the $K \times K$ matrix with entries $W_{k,\ell}^{(i)} = \langle v^{(k,i)}, v^{(\ell,i)}\rangle$ and define $W^{(i)\prime}$ similarly. Note that the matrices $W^{(i)}, W^{(i)\prime}$ over all i are independent. In addition, $W^{(i)}$ has the $\mathsf{Wishart}(K, N_i)$ distribution, and $W^{(i)\prime}$ has the $\mathsf{Wishart}(K, N_i')$ distribution. In addition, for any $k, \ell \le K$ and $j \le T$, we have that $\langle v^{(k)}, D^j\, v^{(\ell)}\rangle = \sum_{i=1}^{K+2} \lambda_i^j W_{k,\ell}^{(i)}$.

Now, we attempt to design a coupling between the matrices $\{W^{(i)}\}_{i=1}^{K+2}$ and $\{W^{(i)\prime}\}_{i=1}^{K+2}$ such that $W^{(i)} - W^{(i)\prime} = (x_i - x_i')\, I_K$ for all $i \le K+2$, with high probability. Note that this implies our claim, due to Corollary 5.6. To design this coupling, first note that by Lemma 5.3, if we draw $Z^{(i)} \sim N_i\, I_K + \sqrt{2N_i}\,\mathsf{GOE}(K)$, then $\|\mathrm{law}(W^{(i)}) - \mathrm{law}(Z^{(i)})\|_{\mathrm{TV}} \le O(K^{3/2}/N_i^{1/2})$, and a similar statement holds if we define $Z^{(i)\prime}$ and compare its law to that of $W^{(i)\prime}$.

Note that the entries of $Z^{(i)}$ and $Z^{(i)\prime}$ are independent (apart from the requirement of symmetry), so we will attempt a coupling between the entries $Z_{k,\ell}^{(i)}$ and $Z_{k,\ell}^{(i)\prime}$. For $k < \ell$, since $Z_{k,\ell}^{(i)} \sim \mathcal{N}(0, N_i)$ and $Z_{k,\ell}^{(i)\prime} \sim \mathcal{N}(0, N_i')$, the total variation distance between their distributions is bounded up to a constant, using Corollary 5.6, by

$$\Bigl|\frac{N_i'}{N_i} - 1\Bigr| \le \Bigl|\frac{x_i'}{x_i} - 1\Bigr| + \Bigl|\frac{N_i' - x_i'}{N_i}\Bigr| + \Bigl|\frac{x_i'\,(N_i - x_i)}{N_i x_i}\Bigr| \le O\bigl(c_1 + \frac{K}{d}\bigr)$$

under our assumptions. Therefore, we can couple $Z_{k,\ell}^{(i)}$ and $Z_{k,\ell}^{(i)'}$ such that they fail to coincide with this probability. For $k = \ell$, we have $Z_{k,k}^{(i)} \sim \mathcal{N}(N_i, 2N_i)$ and $Z_{k,k}^{(i)\prime} + x_i - x_i' \sim \mathcal{N}(N_i' + x_i - x_i', 2N_i')$. The total variation distance between their distributions is bounded by a constant times

$$\Bigl|\frac{N_i'}{N_i} - 1\Bigr| + \frac{|N_i' - x_i' + x_i - N_i|}{\sqrt{N_i}} \le O\bigl(c_1 + \frac{K^{1/2}}{d^{1/2}}\bigr)\,.$$

Therefore, we can couple the two random variables together so that $Z_{k,k}^{(i)} = Z_{k,k}^{(i)\prime} + x_i - x_i'$ fails with the above probability.

By a union bound, the coupling $Z^{(i)} = Z^{(i)\prime} + (x_i - x_i')\, I_K$ for all i fails with probability at most

$$O\Bigl(K^3\,\bigl(c_1 + \frac{K}{d}\bigr) + K^2\,\bigl(c_1 + \frac{K^{1/2}}{d^{1/2}}\bigr)\Bigr) = O\bigl(c_1 K^3 + \frac{K^{5/2}}{d^{1/2}}\bigr)\,.$$

Combining this with comparison between the Wishart and GOE ensembles and another union bound, we obtain the result. □

Finally, we are able to prove our main lower bound against block Krylov algorithms.

Lemma 5.8 (lower bound against block Krylov algorithms). *Let κ, K, D, D' be as in Lemma 5.7. Then, let U be a uniformly random orthogonal matrix in $\mathbb{R}^{d\times d}$, and let $\Lambda = U^\intercal DU$ and $\Lambda' = U^\intercal D'U$. Let $v^{(1)}, \dots, v^{(K)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$. Then, for any $\delta > 0$, provided $K \le O_\delta(\sqrt{\kappa}\log d)$ and $\kappa \le d^{1/5-\delta}$, the distributions of $\{\Lambda^j v^{(k)}\}_{j\le(K+2)/2,\ k\le K}$ and $\{\Lambda'^j v^{(k)}\}_{j\le(K+2)/2;\ k\le K}$ differ in total variation distance by at most $o(1)$. On the other hand, drawing a sample either from $\mathcal{N}(0,\Lambda^{-1})$ or $\mathcal{N}(0,\Lambda'^{-1})$ can, with probability $1-o(1)$, distinguish between the two cases.*

Proof. From Lemma 5.7, there is a coupling such that the tuples $\{\langle v^{(k)}, D^j\, v^{(\ell)}\rangle\}_{j\le K+2,\ k,\ell\le K}$ and $\{\langle v^{(k)\prime}, D'^j\, v^{(\ell)\prime}\rangle\}_{j\le K+2,\ k,\ell\le K}$ are equal with high probability. In particular, it holds that

$$\langle D^i\, v^{(k)}, D^j\, v^{(\ell)}\rangle = \langle D'^i\, v^{(k)\prime}, D'^j\, v^{(\ell)\prime}\rangle \qquad \text{for all } i, j \le (K+2)/2 \text{ and } k \le K$$

with high probability. By Proposition 5.4, there is a unitary matrix U_0 such that $D'^j\, v^{(k)\prime} = U_0 D^j\, v^{(k)}$ for all $j \le (K+2)/2$ and all $k \le K$ with high probability. Note then that the tuples $\{U^\intercal D^j U\, U^\intercal v^{(k)}\}_{j\le(K+2)/2,\ k\le K}$ and $\{U^\intercal U_0^\intercal D'^j U_0 U\, U^\intercal U_0^\intercal v^{(k)\prime}\}_{j\le(K+2)/2,\ k\le K}$ are equal with high probability, and this is a coupling which witnesses the fact that

$$\big\|\mathrm{law}\big(\{\Lambda^j v^{(k)}\}_{j\le(K+2)/2,\ k\le K}\big) - \mathrm{law}\big(\{\Lambda'^j v^{(k)}\}_{j\le(K+2)/2,\ k\le K}\big)\big\|_{\mathrm{TV}} \le O(c_1 K^3 + K^3/d^{1/2})\,.$$

Finally, we note that from a single sample it is easy to distinguish between $\mathcal{N}(0,\Lambda^{-1})$ and $\mathcal{N}(0,\Lambda'^{-1})$. This is because if $X \sim \mathcal{N}(0,\Lambda^{-1})$, then $\mathbb{E}[\|X\|^2] = \operatorname{tr}(\Lambda^{-1}) = \operatorname{tr}(D^{-1}) = \sum_{i=1}^{K+2} N_i/\lambda_i$, but one checks that $\operatorname{var}(\|X\|^2) = O(\sum_{i=1}^{K+2} N_i/\lambda_i^2) \le O(d)$. Likewise, if $X' \sim \mathcal{N}(0,\Lambda'^{-1})$, then we have $\mathbb{E}[\|X'\|^2] = \sum_{i=1}^{K+2} N_i'/\lambda_i$ but $\operatorname{var}(\|X'\|^2) = O(d)$. So, the difference in their expectations at least $c_1 d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa - 2\,(K+2)$, whereas the standard deviations are bounded by $O(d^{1/2})$.

To finish the proof, we must choose the values of c_0 and c_1. We require the following conditions:

(1) $c_1 K^3 = o(1)$.
(2) $K^3/d^{1/2} = o(1)$.
(3) $d^{1/2} = o(c_1 d^{1-2c_0-O(1/\sqrt{\kappa})}/\kappa - 2\,(K+2))$.

For the second condition, we can assume $\kappa \le d^{1/3}/\log^4(d)$. To satisfy the first condition, we can set $c_1 = 1/(\kappa^{3/2}\log^4(d))$. Finally, if κ is sufficiently large and if c_0 is chosen depending on δ, then the third condition requires $\sqrt{\kappa}\log d + d^{1/2} = o(d^{1-\delta}/\kappa^{5/2})$, and it suffices for $\kappa \le d^{1/5-\delta}$. □

Remark 5.9. We did not attempt to optimize the exponent in the condition $\kappa \le d^{1/5-\delta}$. Indeed, by using the chain rule for the KL divergence rather than a union bound in the proof of Lemma 5.7, we believe that the total variation bound can be improved to $O(c_1 K^{3/2} + K^{5/2}/d^{1/2})$, and a back-of-the-envelope calculation suggests that this could improve the condition to $\kappa \le d^{2/7-\delta}$. Nevertheless, this falls short of capturing the full regime $\sqrt{\kappa}\log d \le O(d)$, and we leave this as an open question.

5.3 Reduction to block Krylov algorithms

In this section, we show that in order to prove a lower bound for sampling from Gaussians against any query algorithm, it suffices to prove a lower bound against block Krylov algorithms.

5.3.1 Setup. Let $\Lambda = U^\intercal DU$, where D is a (possibly random) diagonal matrix, U is a Haar-random orthogonal matrix, and U and D are independent. We consider the following model, which is a strengthening of the matrix-vector product model:

Definition 5.10 (extended oracle model). Given $K \in \mathbb{N}$, for all $k \in [K]$, the algorithm chooses a new query point v_k, and receives the information $\{\Lambda^i v_j\}_{(i,j)\in H_k}$, where $H_k := \{(i,j) : i + j \le k+1, i \ge 0, 1 \le j \le k\}$ is a set of ordered pairs of nonnegative integers. We use the following notation $\{\Lambda^i v_j\}_S$ for any set S to denote $\{\Lambda^i v_j\}_{(i,j)\in S}$.

This is clearly a stronger oracle model than before, so a lower bound against algorithms in the extended oracle model implies a lower bound against algorithms in the original matrix-vector model.

Definition 5.11 (adaptive deterministic algorithm). An *adaptive deterministic algorithm* $\mathcal{A}$ that makes K extended oracle queries (see Definition 5.10) is given by a deterministic collection of functions $v_1, v_2(\cdot), \ldots, v_K(\cdot)$, where v_1 is constant and each $v_k(\cdot)$ is a function of $\frac{k\,(k+1)}{2} - 1$ inputs. This corresponds to a sequence of queries where the k-th query $v_k(\{\Lambda^i v_j\}_{H_{k-1}})$ is chosen adaptively based on the information available to the algorithm at the start of iteration k. (Note that v_1 has no inputs.) When the choice of the inputs is clear from context, we may simply write $v_k = v_k(\{\Lambda^i v_j\}_{H_{k-1}})$.

In the extended oracle model, the next lemma shows that we can assume that each v_k is a unit vector orthogonal to its inputs.

Lemma 5.12 (extended oracle and orthogonal queries). *For $k \in [2, K]$, let v_k be as stated in Definition 5.11 and let $\{\Lambda^i v_j\}_{H_{k-1}}$ be as stated in Definition 5.10. Then, v_k is orthogonal to the subspace spanned by the vectors in $\{\Lambda^i v_j\}_{H_{k-1}}$.*

Proof. Assume for sake of contradiction that this were not the case. Then, we can decompose $v_k = \sum_{(i,j)\in H_{k-1}} c_{i,j}\Lambda^i v_j + c^\perp v_k^\perp$ where $v_k^\perp$ is a unit vector orthogonal to $\{\Lambda^i v_j\}_{H_{k-1}}$ and each $c_{i,j}$ and $c^\perp$ is a scalar. At the end of iteration k, the new information obtained by the algorithm is $\{\Lambda^i v_j\}_{i+j=k+1, j\le k}$. For all $(i,j) \ne (1,k)$, the new information does not depend on v_k. Also, $\Lambda v_k = \sum_{(i,j)\in H_{k-1}} c_{i,j}\Lambda^{i+1} v_j + c^\perp \Lambda v_k^\perp$, where each $\Lambda^{i+1} v_j$ is information obtained by the algorithm at the end of iteration $k+1$ regardless (due to our extended query model). Since $(i+1, j) \in H_k$ if $(i,j) \in H_{k-1}$, and since $(1,k) \in H_k$, this expression shows that the algorithm would receive the same amount of information (or more, if $c^\perp = 0$) if it queries $v_k^\perp$ instead of v_k. Applying this reasoning inductively proves the claim. □

We compare to a *block Krylov algorithm*, which makes i.i.d. standard Gaussian queries $z_1, \ldots, z_K$ and then receives $\{\Lambda^i z_j\}$ for all $i, j \le K$. Recall that a block Krylov algorithm does not make *adaptive* queries, so it is easier to prove lower bounds against block Krylov algorithms. Our goal is to now show that block Krylov algorithms can simulate an adaptive deterministic algorithm.

5.3.2 Conditioning lemma.

We start by proving a general conditioning lemma which will be invoked repeatedly in the reduction to block Krylov algorithms. This lemma roughly shows that if the adaptive algorithm knows $\{\Lambda^i v_j\}_{H_k}$, the posterior distribution of Λ given $\{\Lambda^i v_j\}_{H_k}$ is indeed rotationally symmetric on the orthogonal complement $\{\Lambda^i v_j\}_{H_k}$.

We will use the notation $\overset{\mathrm{d}}{=}$ to denote that two random variables are equal in probability distribution (possibly conditioned on other information).

Lemma 5.13 (conditioning lemma, preliminary version). *Let U be a Haar-random orthogonal matrix, and $\Lambda = U^\top D U$, where D is a (possibly random) positive diagonal matrix. Suppose that $\mathcal{A}$ is an adaptive deterministic algorithm that generates extended oracle queries $v_1, \ldots, v_K$, and after the k-th query knows $\Lambda^i v_j$ for all $(i,j) \in H_k$. For any integer $m \ge 1$, let k be the integer such that $\frac{k(k+1)}{2} \le m < \frac{(k+1)(k+2)}{2}$, i.e., m is at least the k-th triangular number but less than the $(k+1)$-th triangular number. Consider the order of vectors $v_1, \Lambda v_1, v_2, \Lambda^2 v_1, \Lambda v_2, v_3, \Lambda^3 v_1, \ldots$ (this enumerates $\Lambda^i v_j$ in order of $i + j$, breaking ties with smaller values of j first). Let W_m be the set of first m of* *these vectors and X_k be the set $\{v_1, \dots, v_k\}$. Let V be a Haar-random orthogonal matrix fixing W_m and acting on the orthogonal complement $W_m^\perp$. Then, $(X_k, U) \overset{\mathrm{d}}{=} (X_k, UV)$.*

Before proving this lemma, we note that since the algorithm is deterministic and D is fixed, W_m and X_k are deterministic functions of Λ, and thus of U. Hence, we can write $v_k(U'), W_m(U'), X_k(U')$ to be the v_k, W_m, X_k that would have been generated if we started with $\Lambda' = (U')^\intercal DU'$. (If no argument is given, v_k, W_m, X_k are assumed to mean $v_k(U), W_m(U), X_k(U)$, respectively.) We note the following proposition.

Proposition 5.14 (fixing the first m queries and responses). *Suppose that V is any orthogonal matrix fixing $W_m(U)$. Then, $W_m(U) = W_m(UV)$.*

Proof. We prove $W_{m'}(U) = W_{m'}(UV)$ for all $m' \le m$. The base case of $k = 1$ is trivial, since v_1 is fixed. We now prove the induction step for m'.

If $m' \le m$ is a triangular number, $m' = \frac{k(k+1)}{2}$, then the m'-th vector in W_m is v_k. But note that $v_k(U)$ is a deterministic function of $W_{m'-1}(U)$, and $v_k(UV)$ is the same deterministic function of $W_{m'-1}(UV)$. Hence, if the induction hypothesis holds for $m' - 1$, it also holds for m.

If $m' \le m$ is not a triangular number, then the m'-th number in $W_m(U)$ is $\Lambda^i v_j$ for some $i \ge 1$. Likewise, the m'-th number in $W_m(UV)$ is $V^\intercal \Lambda^i V v_j(UV)$. Since $i \ge 1$, we know that $v_j(U) = v_j(UV)$, by the induction hypothesis on $\frac{j(j+1)}{2} < m'$. But, we know that V fixes W_m, which means it fixes v_j and $\Lambda^i v_j$. Thus, $V^\intercal \Lambda^i V v_j(UV) = V^\intercal \Lambda^i V v_j = \Lambda^i v_j$. □

We are now ready to prove Lemma 5.13.

Proof of Lemma 5.13. We prove this by induction on m. For the base case $m = 1$, U is a random matrix and V is a random matrix that fixes v_1. Note that v_1 is chosen independently of Λ (and thus of U), so U and V are independent. Even for any fixed V, the distribution UV is a uniformly random orthogonal matrix, so overall $U \overset{\mathrm{d}}{=} UV$. Also, v_1 is deterministic, so $(v_1, U) \overset{\mathrm{d}}{=} (v_1, UV)$.

For the induction step, we split the proof into 2 cases. The proofs in both cases will be very similar, but with minor differences.

Case 1: m is a triangular number. This means that the m-th vector added is v_k, where $m = \frac{k(k+1)}{2}$. Let V_1 be a random orthogonal matrix fixing W_{m-1} and V_2 be a random orthogonal matrix fixing W_m. Our goal is then to show $(X_k, U) \overset{\mathrm{d}}{=} (X_k, UV_2)$.

To make this rigorous, we note an order of generating the random variables. First, we generate U randomly: W_m and X_k are deterministic in terms of U. Next, we define V_1 to be a random rotation fixing W_{m-1}. Finally, we define V_2 to be a random rotation fixing W_m, where V_1, V_2 are conditionally independent on U.

First, we prove that $(X_k, U) \overset{\mathrm{d}}{=} (X_k, UV_1)$. Note that $U \overset{\mathrm{d}}{=} UV_1$ by our inductive hypothesis. In addition, since V_1 fixes $W_{m-1}(U)$, $W_{m-1}(U) = W_{m-1}(UV_1)$ by Proposition 5.14. Since $m = \frac{k(k+1)}{2}$ is a triangular number, $X_k(\cdot)$ is a deterministic function of $W_{m-1}(\cdot)$, which means $X_k(U) = X_k(UV_1)$. Hence, $(X_k, U) \overset{\mathrm{d}}{=} (X_k(UV_1), UV_1) = (X_k, UV_1)$.

Next, we prove that $(X_k, UV_2) \overset{\mathrm{d}}{=} (X_k, UV_1V_2)$. It suffices to prove that

$$(X_k, U, V_2) \overset{\mathrm{d}}{=} (X_k, UV_1, V_2)\,.$$

To do so, we first show that $V_2 = f(U, R)$, where f is a deterministic function and R represents a random orthogonal matrix over $d - \dim(W_m)$ dimensions that is independent of U. (Recall that W_m is a deterministic function of U.) To define $f(U, R)$, we consider some deterministic map that sends each W_m to a set of $d - \dim(W_m)$

basis vectors in $W_m^\perp$. We then define $V_2 = f(U,R)$ to act on $W_m^\perp$ using R and the correspondence of basis vectors. Since W_m and X_k are deterministic in terms of U, this means $f(U,R)$ is well-defined. We will now show that

$$V_2 = f(U,R) = f(UV_1,R) \qquad \text{and} \qquad X_k = X_k(UV_1)\,.$$

Since $U \overset{\mathrm{d}}{=} UV_1$ by our inductive hypothesis,

$$(X_k, U, V_2) \overset{\mathrm{d}}{=} (X_k(UV_1), UV_1, f(UV_1,R)) = (X_k, UV_1, V_2)\,.$$

By Proposition 5.14, $W_{m-1}(U) = W_{m-1}(UV_1)$, and since $X_k(\cdot)$ is deterministic given $W_{m-1}(\cdot)$ for $m = \frac{k(k+1)}{2}$, $X_k(U) = X_k(UV_1)$. This implies $W_m(U) = W_m(UV_1)$, which means $f(UV_1,R) = f(U,R)$, since $f(\cdot,R)$ only depends on $W_m(\cdot)$ and R. This completes the proof.

Next, we show that $(X_k, UV_1V_2) \overset{\mathrm{d}}{=} (X_k, UV_1)$. Since we chose the order with U being defined first, we are allowed to condition on U. Since X_k is deterministic in terms of U, it suffices to show that $V_1V_2 \mid U \overset{\mathrm{d}}{=} V_1 \mid U$. Since W_{m-1}, W_m are also deterministic given U, note that V_1 is a uniformly random orthogonal matrix fixing W_{m-1}, and V_2 is a random orthogonal matrix fixing $W_m \supset W_{m-1}$. Since V_1 and V_2 are conditionally independent given U, this means $V_1V_2 \mid U$ is a uniformly random orthogonal matrix fixing W_{m-1}, so $V_1V_2 \mid U \overset{\mathrm{d}}{=} V_1 \mid U$.

In summary, we have that

$$\begin{aligned}(X_k, U) &\overset{\mathrm{d}}{=} (X_k, UV_1)\\ &\overset{\mathrm{d}}{=} (X_k, UV_1V_2)\\ &\overset{\mathrm{d}}{=} (X_k, UV_2)\,.\end{aligned}$$

Case 2: m is not a triangular number. Again, let V_1 be a random orthogonal matrix fixing W_{m-1} and V_2 be a random orthogonal matrix fixing W_m. Our goal is again to show that $(X_k, U) \overset{\mathrm{d}}{=} (X_k, UV_2)$.

First, we again have $(X_k, UV_1) \overset{\mathrm{d}}{=} (X_k, U)$ by our inductive hypothesis.

Next, we show that $(X_k, UV_2) \overset{\mathrm{d}}{=} (X_k, UV_2V_1)$. It suffices to prove that

$$(X_k, U, V_2) \overset{\mathrm{d}}{=} (X_k, UV_1, V_1^\mathsf{T} V_2 V_1)\,,$$

since $(UV_1)(V_1^\mathsf{T} V_2 V_1) = UV_2V_1$. We recall the random variable R and use the same function $V_2 = f(U,R)$. Since we have already shown that $U \overset{\mathrm{d}}{=} UV_1$, this implies that

$$(X_k, U, V_2) \overset{\mathrm{d}}{=} (X_k(UV_1), UV_1, f(UV_1,R))\,.$$

Since m is not triangular, $X_k(\cdot)$ is contained in $W_{m-1}(\cdot)$, so by Proposition 5.14, $X_k(U) = X_k(UV_1)$. So, we have

$$(X_k, U, V_2) \overset{\mathrm{d}}{=} (X_k(UV_1), UV_1, f(UV_1,R)) = (X_k, UV_1, f(UV_1,R))\,.$$

Now, if we fix U and V_1, $W_{m-1}(UV_1) = W_{m-1}(U)$ by Proposition 5.14. However, since the m-th (i,j) pair has $i \geq 1$ when m is not triangular, the final vector in $W_m(UV_1)$ will be $V_1^\mathsf{T} \Lambda^i V_1 v_j = V_1^\mathsf{T} (\Lambda^i v_j)$. For fixed U, V_1, $f(U,R)$ is a random rotation fixing W_{m-1} and $\Lambda^i v_j$, but $f(UV_1,R)$ is a random rotation fixing W_{m-1} and $V_1^\mathsf{T}(\Lambda^i v_j)$. Since V_1^T fixes W_{m-1} by how we defined V_1, this means that for fixed U, V_1, $f(U,R)$ is a random rotation fixing W_m but $f(UV_1,R)$ is a random rotation fixing $V_1^\mathsf{T} W_m$. Therefore, conditioned on U, V_1, $f(UV_1,R)$ has the same distribution as $V_1^\mathsf{T} f(U,R) V_1$. Since X_k is deterministic in terms of U, this means

$$(X_k, UV_1, f(UV_1,R)) \mid U, V_1 \overset{\mathrm{d}}{=} (X_k, UV_1, V_1^\mathsf{T} f(U,R) V_1) \mid U, V_1\,.$$

We can remove the conditioning to establish that $(X_k, UV_1, f(UV_1, R)) \overset{\mathsf{d}}{=} (X_k, UV_1, V_1^\top f(U,R)V_1) = (X_k, UV_1, V_1^\top V_2 V_1)$, which completes the proof.

Next, we show that $(X_k, UV_2V_1) \overset{\mathsf{d}}{=} (X_k, UV_1)$. The proof is essentially the same as in the case when m is triangular. We again condition on U, and we have that $V_2V_1 \mid U \overset{\mathsf{d}}{=} V_1 \mid U$ have the same distribution as uniform orthogonal matrices fixing $W_{m-1}(U)$. Since X_k is a deterministic function of U, this means $(X_k, UV_2V_1) \mid U \overset{\mathsf{d}}{=} (X_k, UV_1) \mid U$, and removing the conditioning finishes the proof.

In summary,

$$\begin{aligned}(X_k, U) &\overset{\mathsf{d}}{=} (X_k, UV_1)\\ &\overset{\mathsf{d}}{=} (X_k, UV_2V_1)\\ &\overset{\mathsf{d}}{=} (X_k, UV_2)\,. \end{aligned}$$

□

We now prove our main conditioning lemma, which will be a modification of Lemma 5.13.

Lemma 5.15 (conditioning lemma). *Let all notation be as in Lemma 5.13, and let V_0 be a fixed orthogonal matrix fixing W_m. Importantly, V_0 is a deterministic function only depending on W_m (and not directly on U). Then, $(X_k, U) \overset{\mathsf{d}}{=} (X_k, UV_0)$.*

Proof. First, note that since V_0 is a deterministic function of W_m, it is also a deterministic function of U. We can write $V_0(\cdot)$ as this function, and $V_0 = V_0(U)$.

Now, Lemma 5.13 proves that $(X_k, U) \overset{\mathsf{d}}{=} (X_k, UV)$. Note that conditioned on U, V is a random matrix fixing W_m and V_0 is a fixed matrix fixing W_m, which means that $VV_0 \mid U \overset{\mathsf{d}}{=} V \mid U$. Hence, $(X_k, UV) \overset{\mathsf{d}}{=} (X_k, UVV_0)$. But from Proposition 5.14, $X_k(UV) = X_k(U)$ and $W_m(UV) = W_m(U)$, which means that $V_0(\cdot)$, which only depends on $W_m(\cdot)$, satisfies $V_0(UV) = V_0(U)$. Hence, because $U \overset{\mathsf{d}}{=} UV$, we have $(X_k, UVV_0) = (X_k(UV), UV \cdot V_0(UV)) \overset{\mathsf{d}}{=} (X_k(U), U \cdot V_0(U)) = (X_k, UV_0)$.

In summary, we have that $(X_k, U) \overset{\mathsf{d}}{=} (X_k, UV) \overset{\mathsf{d}}{=} (X_k, UVV_0) \overset{\mathsf{d}}{=} (X_k, UV_0)$, which completes the proof. □

5.3.3 From query algorithms to block Krylov algorithms. In this section, we carry out the high-level outline from Section 2.2.2. We aim to prove the following result, which implies that any adaptive deterministic algorithm in the extended oracle model can be simulated by rotating the output of a block Krylov algorithm.

Lemma 5.16 (reduction to block Krylov). *Suppose $\Lambda = U^\top DU$, where U is a Haar-random orthogonal matrix and D is a diagonal matrix drawn from some (possibly unknown) distribution. Let $v_1, v_2(\cdot), \dots, v_K(\cdot)$ be an adaptive deterministic algorithm that makes K queries, where $K^2 < d$. Let $v_1^{\mathsf{alg}}, v_2^{\mathsf{alg}}, \dots, v_K^{\mathsf{alg}}$ be recursively defined as follows: $v_1^{\mathsf{alg}} = v_1$, and $v_k^{\mathsf{alg}} = v_k(\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_{k-1}})$ for $k \ge 2$. Let $z_1, \dots, z_K$ be i.i.d. standard Gaussian vectors. Then, from the collection $\{\Lambda^i z_j\}_{H_K}$ (without knowledge of D or Λ), we can construct a set of unit vectors $\tilde v_1, \tilde v_2, \dots, \tilde v_K$, and a set of rotation matrices $U_1^{\mathsf{sim}}, U_2^{\mathsf{sim}}, \dots, U_K^{\mathsf{sim}}$, where $\tilde v_k$ and U_k^{sim} only depend on $\{\Lambda^i z_j\}_{H_{k-1}}$ and z_k, and such that*

$$\{(U_{1:K}^{\mathsf{sim}})^\top \Lambda^i \tilde v_j\}_{H_K} \overset{\mathsf{d}}{=} \{\Lambda^i v_j^{\mathsf{alg}}\}_{H_K},$$

where $U_{1:K}^{\mathsf{sim}} := U_1^{\mathsf{sim}} \cdots U_K^{\mathsf{sim}}$, and the equivalence in distribution is over the randomness of Λ and $\{z_i\}_{i \le K}$. Moreover, $\{\Lambda^i \tilde v_j\}_{H_K}$ is deterministically determined by $\{\Lambda^i z_j\}_{H_K}$.

Lemma 5.16 says that the knowledge of $\Lambda^i z_j$ alone is sufficient to reconstruct the distribution of any adaptive algorithm's queries and responses. The proof of the lemma requires introducing a hefty amount of notation, but we emphasize that it follows along the lines of Section 2.2.2.

First, we describe how to construct $\tilde{v}_k$. Let $\tilde{v}_1 = \frac{z_1}{\|z_1\|}$, and for $k \geq 2$, let $\tilde{v}_k$ be the unit vector parallel to the component of z_k that is orthogonal to the span of $\{\Lambda^i z_j\}_{H_{k-1}}$. (With probability 1, this is well-defined.)

Because each $\tilde{v}_k$ is a linear combination of $\{\Lambda^i z_j\}_{H_{k-1}}$ and z_k, we can construct the set $\{\Lambda^i \tilde{v}_j\}_{H_K}$ from the set $\{\Lambda^i z_j\}_{H_K}$.

We now construct the rotation matrices U_k^{sim}. First, we define matrix-valued functions $U_k(\cdot)$, for $k = 1, \ldots, K$, as follows.

Definition 5.17 (rotations fixing previous queries and responses). For $1 \leq k \leq K$, the function $U_k(\cdot)$ takes arguments $\{x_{i,j}\}_{H_{k-1}}, y_k, z_k$, where the vectors y_k and z_k have unit norm and are both orthogonal to the collection $\{x_{i,j}\}_{H_{k-1}}$.

To define $U_1(\cdot)$: since H_0 is empty, the first function U_1 only takes arguments y_1, z_1, and is such that $U_1(y_1, z_1)$ is a deterministic orthogonal matrix that satisfies $U_1(y_1, z_1)^\top y_1 = z_1$. Note that $U_1(\cdot)$ exists because y_1 and z_1 both have unit norm; for example, we can complete y_1 and z_1 to orthonormal bases $(y_1, y_2, \ldots, y_d)$, $(z_1, z_2, \ldots, z_d)$ and take $U_1(y_1, z_1) = \sum_{i=1}^d y_i z_i^\top$.

To define $U_k(\cdot)$: $U_k(\{x_{i,j}\}_{H_{k-1}}, y_k, z_k)$ is a deterministic orthogonal matrix that satisfies

$$\begin{aligned} U_k^\top x_{i,j} &= x_{i,j}\,, \qquad \text{for all } (i,j) \in H_{k-1}\,, \\ U_k^\top y_k &= z_k\,. \end{aligned} \tag{15}$$

Such a choice of U_k is always possible, because $k^2 < d$, and because y_k and z_k are orthogonal to $x_{i,j}$; for example, we can start with the identity matrix on the subspace spanned by $\{x_{i,j}\}_{H_{k-1}}$ and add to it a sum of outer products formed by completing y_k and z_k to two orthonormal bases of the orthogonal complement.

Next, we describe how to construct U_k^{sim}. We will define U_k^{sim} along with an auxiliary sequence $\{v_k^{\mathsf{sim}}\}_{k=1,2,\ldots,K-1}$.

Definition 5.18 (simulated sequences). We let $v_1^{\mathsf{sim}} = v_1$, and $U_1^{\mathsf{sim}} = U_1(\tilde{v}_1, v_1^{\mathsf{sim}})$. For $k \geq 2$, v_k^{sim} and U_k^{sim} are defined recursively as follows:

$$\begin{aligned} v_k^{\mathsf{sim}} &= v_k\big(\{(U_{1:(k-1)}^{\mathsf{sim}})^\top \Lambda^i \tilde{v}_j\}_{H_{k-1}}\big) \\ U_k^{\mathsf{sim}} &= U_k\big(\{(U_{1:(k-1)}^{\mathsf{sim}})^\top \Lambda^i \tilde{v}_j\}_{H_{k-1}},\ (U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_k,\ v_k^{\mathsf{sim}}\big)\,. \end{aligned} \tag{16}$$

Intuitively, one can think of v_k^{sim} as the kth vector the simulator thinks the algorithm is querying, and U_k^{sim} as a rotation that corresponds v_k^{sim} to the random unit vector known by block Krylov.

Proposition 5.19 (existence of rotations). *Each U_k^{sim} is well-defined.*

Proof. To show that this choice of U_k^{sim} is possible, we need to check that $(U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_k$, v_k^{sim} both have unit norm and are orthogonal to the subspace S_k spanned by $(U_{1:(k-1)}^{\mathsf{sim}})^\top \Lambda^i \tilde{v}_j$ for $(i,j) \in H_{k-1}$. They both have unit norm because $\tilde{v}_k$ and v_k^{sim} are constructed to have unit norm, and inductively we can assume $U_{1:(k-1)}^{\mathsf{sim}}$ is orthogonal. Note that v_k^{sim} is orthogonal to S_k by our assumption on the function $v_k(\cdot)$, and $(U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_k$ is also orthogonal to S_k because

$$\langle (U_{1:(k-1)}^{\mathsf{sim}})^\top \Lambda^i \tilde{v}_j, (U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_k \rangle = \langle \Lambda^i \tilde{v}_j, \tilde{v}_k \rangle = 0\,,$$

where the second line follows from the definition of $\tilde{v}_k$. □

We summarize some additional properties of v_k^{sim} and U_k^{sim} in the following lemma.

Lemma 5.20 (properties of the simulated sequences). *The variables U_k^{sim} and v_k^{sim} for $k = 1, \ldots, K$ defined above satisfy the following properties:*

(P1) v_k^{sim} depends only on $\{\Lambda^i \tilde{v}_j\}_{H_{k-1}}$, and U_k^{sim} depends only on $\{\Lambda^i \tilde{v}_j\}_{i+j\le k}$.

(P2) For any $k \ge j$, we have

$$\tilde{v}_j = U_{1:k}^{\mathsf{sim}} v_j^{\mathsf{sim}} \,.$$

(P3) For $k \ge 2$, v_k^{sim} satisfies

$$v_k^{\mathsf{sim}} = v_k\bigl(\{(U_{1:(k-1)}^{\mathsf{sim}})^\top \Lambda^i U_{1:(k-1)}^{\mathsf{sim}} v_j^{\mathsf{sim}}\}_{H_{k-1}}\bigr)\,.$$

(P4) For $k \ge 2$, U_k^{sim} satisfies

$$U_k^{\mathsf{sim}} = U_k\bigl(\{(U_{1:(k-1)}^{\mathsf{sim}})^\top \Lambda^i U_{1:(k-1)}^{\mathsf{sim}} v_j^{\mathsf{sim}}\}_{H_{k-1}},\ (U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_k,\ v_k^{\mathsf{sim}}\bigr)\,.$$

Proof. (P1) is immediate from the definitions, since $\{(i,j) : i + j \le k\} = H_{k-1} \cup \{(0,k)\}$.

To show (P2), note that the second property of the function U_k from (15) implies that

$$v_j^{\mathsf{sim}} = (U_j^{\mathsf{sim}})^\top (U_{1:(j-1)}^{\mathsf{sim}})^\top \tilde{v}_j = (U_{1:j}^{\mathsf{sim}})^\top \tilde{v}_j\,. \tag{17}$$

This proves (P2) for $k = j$. To prove (P2) for $k > j$, we use induction on k. If (P2) holds for $k - 1 \ge j$, then

$$(U_{1:k}^{\mathsf{sim}})^\top \tilde{v}_j = (U_k^{\mathsf{sim}})^\top (U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_j = (U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_j = v_j^{\mathsf{sim}}. \tag{18}$$

Above, the middle equality holds by the first property of (15), since U_k^{sim} fixes $(U_{1:(k-1)}^{\mathsf{sim}})^\top \tilde{v}_j$ because $j \le k-1$. The final equality holds by our inductive hypothesis. So, (P2) holds for k.

Finally, (P3) and (P4) then follow from (P2), since $k - 1 \ge j$ if $j \in H_{k-1}$. □

We highlight the importance of (P2) for $k = K$, which roughly states that $(U_{1:K}^{\mathsf{sim}})^\top$ actually sends each block Krylov-generated vector $\tilde{v}_j$ to the simulated vector v_j^{sim}.

Before proving Lemma 5.16, we must make one more basic definition.

Definition 5.21 (queries and data). For $k \ge 2$, given the matrix Λ and a set $\{v_j\}_{1\le j\le k-1}$, define $\mathfrak{C}_k$ as the function that satisfies $\mathfrak{C}_k(\Lambda, \{v_j\}_{1\le j\le k-1}) = \{\Lambda^i v_j\}_{H_{k-1}}$. In addition, define $\mathfrak{D}_k = v_k \circ \mathfrak{C}_k$.

We are now ready to prove Lemma 5.16. Although the proof is notationally burdensome, the message is that we can show the equality of distributions inductively by repeatedly invoking the conditioning lemma (Lemma 5.15), which is designed precisely for the present situation.

Proof of Lemma 5.16. For $1 \le k \le K$, we write $\Lambda_k \coloneqq (U_{1:k}^{\mathsf{sim}})^\top \Lambda U_{1:k}^{\mathsf{sim}}$. Since we can write $(U_{1:k}^{\mathsf{sim}})^\top \Lambda^i \tilde{v}_j = (U_{1:k}^{\mathsf{sim}})^\top \Lambda^i (U_{1:k}^{\mathsf{sim}}) v_j^{\mathsf{sim}} = \Lambda_k^i v_j^{\mathsf{sim}}$ for any $k \ge j$ by (P2) of Lemma 5.20, it suffices to inductively prove that for all $1 \le k \le K$,

$$(\Lambda_k, \{v_j^{\mathsf{sim}}\}_{1\le j\le k}) \overset{\mathrm{d}}{=} (\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k})\,. \tag{19}$$

For the base case of $k = 1$, it suffices to show that $(\Lambda_1, v_1^{\mathsf{sim}}) \overset{\mathrm{d}}{=} (\Lambda, v_1^{\mathsf{alg}})$. Note, however, that $v_1^{\mathsf{sim}} = v_1^{\mathsf{alg}} = v_1$, and $\Lambda_1 = (U_1^{\mathsf{sim}})^\top \Lambda (U_1^{\mathsf{sim}}) = U_1(\tilde{v}_1, v_1)^\top \Lambda U_1(\tilde{v}_1, v_1)$. Since v_1 is a deterministic vector, $\tilde{v}_1$ is independent of Λ, and the distribution of Λ is rotationally invariant, the claim follows.

For the inductive step, assume we know $(\Lambda_k, \{v_j^{\mathsf{sim}}\}_{1\le j\le k}) \overset{\mathrm{d}}{=} (\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k})$. Then, note that $v_{k+1}^{\mathsf{alg}} = v_{k+1}(\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_k})$ and $v_{k+1}^{\mathsf{sim}} = v_{k+1}(\{\Lambda_k^i v_j^{\mathsf{sim}}\}_{H_k})$. Thus, we have $v_{k+1}^{\mathsf{alg}} = \mathfrak{D}_{k+1}(\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k})$ and $v_{k+1}^{\mathsf{sim}} = \mathfrak{D}_{k+1}(\Lambda_k, \{v_j^{\mathsf{sim}}\}_{1\le j\le k})$. In addition, because U_{k+1}^{sim} fixes $\Lambda_k^i v_j^{\mathsf{sim}}$ for all $(i,j) \in H_k$ by (P4), we also have that $\Lambda_{k+1}^i v_j^{\mathsf{sim}} = \Lambda_k^i v_j^{\mathsf{sim}}$ for all $(i,j) \in H_k$, which means $v_{k+1}^{\mathsf{sim}} = \mathfrak{D}_{k+1}(\Lambda_{k+1}, \{v_j^{\mathsf{sim}}\}_{1\le j\le k})$. Therefore, it suffices to show

$$(\Lambda_{k+1}, \{v_j^{\mathsf{sim}}\}_{1\le j\le k}) \overset{\mathrm{d}}{=} (\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k})\,, \tag{20}$$

as this implies $(\Lambda_{k+1}, \{v_j^{\mathsf{sim}}\}_{1\le j\le k+1}) \overset{\mathrm{d}}{=} (\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k+1})$, which completes the inductive step.

Next, we show that U_{k+1}^{sim} sends $\tilde{v}_{k+1}$ to a random unit vector orthogonal to the simulated queries so far. Note that $\Lambda_{k+1} = (U_{k+1}^{\mathsf{sim}})^\intercal \Lambda_k (U_{k+1}^{\mathsf{sim}})$, where, by (P4),

$$U_{k+1}^{\mathsf{sim}} = U_{k+1}(\{\Lambda_k^i v_j^{\mathsf{sim}}\}_{H_k}, (U_{1:k}^{\mathsf{sim}})^\intercal \tilde{v}_{k+1}, v_{k+1}^{\mathsf{sim}})\,. \tag{21}$$

Note that $\tilde{v}_{k+1}$ is a random unit vector orthogonal to $\{\Lambda^i z_j\}_{H_k}$, or equivalently, it is a random unit vector orthogonal to $\{\Lambda^i \tilde{v}_j\}_{H_k}$. Since $(U_{1:k}^{\mathsf{sim}})^\intercal \Lambda^i \tilde{v}_j = (U_{1:k}^{\mathsf{sim}})^\intercal \Lambda^i (U_{1:k}^{\mathsf{sim}}) v_j^{\mathsf{sim}} = \Lambda_k^i v_j^{\mathsf{sim}}$ for all $(i,j) \in H_k$ (by (P2)), this means that $(U_{1:k}^{\mathsf{sim}})^\intercal \tilde{v}_{k+1}$ is orthogonal to $\{\Lambda_k^i v_j^{\mathsf{sim}}\}_{H_k}$. The random direction of $\tilde{v}_{k+1}$ has no dependence on $\{\Lambda^i \tilde{v}_j\}_{H_k}$ apart from being orthogonal to them, which means by (P1), $(U_{1:k}^{\mathsf{sim}})^\intercal \tilde{v}_{k+1}$ is a *uniformly random* unit vector orthogonal to $\{\Lambda_k^i v_j^{\mathsf{sim}}\}_{H_k}$.

Recalling that $v_{k+1}^{\mathsf{sim}} = \mathfrak{D}_{k+1}(\Lambda_k, \{v_j^{\mathsf{sim}}\}_{1\le j\le k})$, this means that we can rewrite (21) as

$$U_{k+1}^{\mathsf{sim}} = U_{k+1}\left(\{\Lambda_k^i v_j^{\mathsf{sim}}\}_{H_k}, \hat{v}^{\mathsf{sim}}, \mathfrak{D}_{k+1}(\Lambda_k, \{v_j^{\mathsf{sim}}\}_{1\le j\le k})\right)\,, \tag{22}$$

where $\hat{v}^{\mathsf{sim}}$ is a random unit vector orthogonal to $\{\Lambda_k^i v_j^{\mathsf{sim}}\}_{H_k}$. As a result, if we define

$$U_{k+1}^{\mathsf{alg}} \coloneqq U_{k+1}\big(\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_k}, \hat{v}^{\mathsf{alg}}, \mathfrak{D}_{k+1}(\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k})\big)\,, \tag{23}$$

where $\hat{v}^{\mathsf{alg}}$ is a random unit vector orthogonal to $\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_k}$, then

$$\begin{aligned}(\Lambda_{k+1}, \{v_j^{\mathsf{sim}}\}_{1\le j\le k}) &= \big((U_{k+1}^{\mathsf{sim}})^\intercal \Lambda_k (U_{k+1}^{\mathsf{sim}}), \{v_j^{\mathsf{sim}}\}_{1\le j\le k}\big)\\ &\overset{\mathrm{d}}{=} \big((U_{k+1}^{\mathsf{alg}})^\intercal \Lambda (U_{k+1}^{\mathsf{alg}}), \{v_j^{\mathsf{alg}}\}_{1\le j\le k}\big)\,.\end{aligned}$$

Above, the first equality follows by definition, and the second follows from our inductive hypothesis that $(\Lambda_k, \{v_j^{\mathsf{sim}}\}_{1\le j\le k}) \overset{\mathrm{d}}{=} (\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k})$, along with (22) and (23).

We are now in a position to apply the conditioning lemma (Lemma 5.15). Note that U_{k+1}^{alg} only depends on $\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_k}$ (as well as some randomness in $\hat{v}^{\mathsf{alg}}$, but the randomness is independent of everything else given $\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_k}$, so we can safely condition on it). Hence, we can apply the conditioning lemma with U_{k+1}^{alg}, to obtain that

$$(\Lambda_{k+1}, \{v_j^{\mathsf{sim}}\}_{1\le j\le k}) \overset{\mathrm{d}}{=} \big((U_{k+1}^{\mathsf{alg}})^\intercal \Lambda (U_{k+1}^{\mathsf{alg}}), \{v_j^{\mathsf{alg}}\}_{1\le j\le k}\big) \overset{\mathrm{d}}{=} \big(\Lambda, \{v_j^{\mathsf{alg}}\}_{1\le j\le k}\big)\,,$$

which establishes (20) and thereby concludes the proof. □

With the block Krylov reduction in hand, we can now establish our second lower bound for sampling from Gaussians.

Theorem 5.22 (second lower bound for sampling from Gaussians). *There is a universal constant $\epsilon_0 > 0$ such that the query complexity of sampling from Gaussian distributions $\mathcal{N}(0,\Sigma)$ in $\mathbb{R}^d$, where the condition number κ of Σ satisfies $\kappa \le d^{1/5-\delta}$, with accuracy ϵ_0 in total variation distance is at least $\Omega_\delta(\sqrt{\kappa}\log d)$.*

Proof. Let U be a random orthogonal matrix, and let $\Lambda = U^\intercal D U$, $\Lambda' = U^\intercal D' U$ be as in Lemma 5.8. We first show that if $\kappa \le d^{1/5-\delta}$ and c is a sufficiently small constant, no adaptive algorithm that makes less than $c_\delta \sqrt{\kappa}\log d$ queries to the extended oracle can distinguish between Λ and Λ', with $\Omega(1)$ probability.

First we assume that the algorithm is deterministic, so its behavior is characterized by functions $v_1, v_2(\cdot), \dots, v_K(\cdot)$, as in Lemma 5.16. The algorithm then proceeds to make queries $v_1^{\mathsf{alg}}, v_2^{\mathsf{alg}}, \dots, v_K^{\mathsf{alg}}$, where $v_k^{\mathsf{alg}} = v_k(\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_{k-1}})$. Lemma 5.16 shows that the output of the algorithm $\{\Lambda^i v_j^{\mathsf{alg}}\}_{H_K}$ can be entirely simulated by a block Krylov algorithm, which receives $\{\Lambda^i z_k\}_{H_K}$, where $z_1, \ldots, z_K$ are i.i.d. standard Gaussians. Lemma 5.8 says that a block Krylov algorithm that makes $K = c_\delta \sqrt{\kappa} \log d$ queries, where c_δ is a small constant depending on δ and $\kappa \leq d^{1/5-\delta}$, cannot distinguish between Λ and Λ' with $\Omega(1)$ advantage, which then implies the same for any deterministic algorithm.

If the algorithm is randomized, then it uses a random seed ξ that is independent of Λ and Λ'. So conditional on the random seed, the algorithm will not be able to distinguish Λ and Λ' with $\Omega(1)$ advantage, so the overall probability that the randomized algorithm successfully distinguishes Λ and Λ' also cannot be $\Omega(1)$.

Finally, we note that a sample from $\mathcal{N}(0, \Lambda^{-1})$ versus $\mathcal{N}(0, \Lambda'^{-1})$ can distinguish between the two cases. This means that even if we were able to draw a sample that was $\frac{1}{3}$-far in total variation distance, we could output the correct answer with probability at least $\frac{2}{3}$. This implies that any sampling algorithm must require at least $\Omega_\delta(\sqrt{\kappa} \log d)$ queries to the extended oracle, and hence at least same number of queries to the standard oracle. □

Acknowledgments

We thank Ainesh Bakshi, Patrik R. Gerber, Piotr Indyk, Thibaut Le Gouic, Philippe Rigollet, Adil Salim, Terence Tao, and Kevin Tian for helpful conversations. SC was supported by the NSF TRIPODS program (award DMS-2022448). JD was supported by a UCLA dissertation year fellowship. CL was supported by the Eric and Wendy Schmidt Center at the Broad Institute of MIT and Harvard. SN was supported by a Google Fellowship, the NSF TRIPODS program (award DMS-2022448), and the NSF Graduate Fellowship. Part of this work was done while SC and SN were research interns at Microsoft Research.

References

Kwangjun Ahn and Sinho Chewi. 2021. Efficient constrained sampling via the mirror-Langevin algorithm. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, K. Nguyen, P. S. Liang, J. W. Vaughan, and Y. Dauphin (Eds.), Vol. 34. Curran Associates, Inc., 28405–28418.

Jason M. Altschuler and Sinho Chewi. 2024. Faster high-accuracy log-concave sampling via algorithmic warm starts. *J. ACM* (3 2024).

Jason M. Altschuler and Kunal Talwar. 2023. Resolving the mixing time of the Langevin algorithm to its stationary distribution for log-concave sampling. In *Proceedings of Thirty Sixth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 195)*, Gergely Neu and Lorenzo Rosasco (Eds.). PMLR, 2509–2510.

Zhaojun Bai, Gark Fahey, and Gene Golub. 1996. Some large-scale matrix computation problems. *J. Comput. Appl. Math.* 7 (1996), 71–89. Issue 1-2.

Ainesh Bakshi, Kenneth L. Clarkson, and David P. Woodruff. 2022. Low-rank approximation with $1/\epsilon^{1/3}$ matrix-vector products. In *54th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 1130–1143.

Krishnakumar Balasubramanian, Sinho Chewi, Murat A. Erdogdu, Adil Salim, and Matthew S. Zhang. 2022. Towards a theory of non-log-concave sampling: first-order stationarity guarantees for Langevin Monte Carlo. In *Conference on Learning Theory*. PMLR, 2896–2923.

Espen Bernton. 2018. Langevin Monte Carlo and JKO splitting. In *Conference on Learning Theory*. PMLR, 1777–1798.

Mark Braverman, Elad Hazan, Max Simchowitz, and Blake E. Woodworth. 2020. The gradient complexity of linear regression. In *Conference on Learning Theory, (COLT) (Proceedings of Machine Learning Research, Vol. 125)*. PMLR, 627–647.

Vladimir Braverman, Aditya Krishnan, and Christopher Musco. 2022. Sublinear time spectral density estimation. In *54th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 1144–1157.

Matthew Brennan, Guy Bresler, and Brice Huang. 2021. De Finetti-style results for Wishart matrices: combinatorial structure and phase transitions. *arXiv e-prints*, Article arXiv:2103.14011 (2021).

Sébastien Bubeck. 2015. Convex optimization: algorithms and complexity. *Foundations and Trends® in Machine Learning* 8, 3-4 (2015), 231–357.

Sébastien Bubeck, Jian Ding, Ronen Eldan, and Miklós Z. Rácz. 2016. Testing for high-dimensional geometry in random graphs. *Random Structures Algorithms* 49, 3 (2016), 503–532.

Sébastien Bubeck and Shirshendu Ganguly. 2018. Entropic CLT and phase transition in high-dimensional Wishart matrices. *Int. Math. Res. Not. IMRN* 2 (2018), 588–606.

Yu Cao, Jianfeng Lu, and Lihan Wang. 2021. Complexity of randomized algorithms for underdamped Langevin dynamics. *Commun. Math. Sci.* 19, 7 (2021), 1827–1853.

Niladri S. Chatterji, Peter L. Bartlett, and Philip M. Long. 2022. Oracle lower bounds for stochastic gradient sampling algorithms. *Bernoulli* 28, 2 (2022), 1074–1092.

Yongxin Chen, Sinho Chewi, Adil Salim, and Andre Wibisono. 2022. Improved analysis for a proximal algorithm for sampling. In *Proceedings of Thirty Fifth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 178)*, Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 2984–3014.

Yuansi Chen, Raaz Dwivedi, Martin J. Wainwright, and Bin Yu. 2020. Fast mixing of Metropolized Hamiltonian Monte Carlo: benefits of multi-step gradients. *J. Mach. Learn. Res.* 21 (2020), 92–1.

Yuansi Chen and Ronen Eldan. 2022. Localization schemes: a framework for proving mixing bounds for Markov chains (extended abstract). In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. 110–122.

Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. 2018. Underdamped Langevin MCMC: a non-asymptotic analysis. In *Proceedings of the 31st Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 75)*, Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet (Eds.). PMLR, 300–323.

Sinho Chewi. 2024. Log-concave sampling. (2024). Book draft available at https://chewisinho.github.io/.

Sinho Chewi, Murat A. Erdogdu, Mufan B. Li, Ruoqi Shen, and Matthew S. Zhang. 2022a. Analysis of Langevin Monte Carlo from Poincaré to log-Sobolev. In *Proceedings of Thirty Fifth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 178)*, Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 1–2.

Sinho Chewi, Patrik R. Gerber, Holden Lee, and Chen Lu. 2023. Fisher information lower bounds for sampling. In *Proceedings of the 34th International Conference on Algorithmic Learning Theory (Proceedings of Machine Learning Research, Vol. 201)*, Shipra Agrawal and Francesco Orabona (Eds.). PMLR, 375–410.

Sinho Chewi, Patrik R. Gerber, Chen Lu, Thibaut Le Gouic, and Philippe Rigollet. 2022b. The query complexity of sampling from strongly log-concave distributions in one dimension. In *Proceedings of Thirty Fifth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 178)*, Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 2041–2059.

Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu, Philippe Rigollet, and Austin J. Stromme. 2020. Exponential ergodicity of mirror-Langevin diffusions. *Advances in Neural Information Processing Systems* 33 (2020), 19573–19585.

Sinho Chewi, Chen Lu, Kwangjun Ahn, Xiang Cheng, Thibaut Le Gouic, and Philippe Rigollet. 2021. Optimal dimension dependence of the Metropolis-adjusted Langevin algorithm. In *Conference on Learning Theory*. PMLR, 1260–1300.

David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. 2018. Approximating the spectrum of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,*. ACM, 1263–1271.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of information theory* (second ed.). Wiley-Interscience [John Wiley & Sons], Hoboken, NJ. xxiv+748 pages.

Arnak S. Dalalyan. 2017. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79, 3 (2017), 651–676.

Arnak S. Dalalyan and Avetik Karagulyan. 2019. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *Stochastic Processes and their Applications* 129, 12 (2019), 5278–5311.

Arnak S. Dalalyan and Lionel Riou-Durand. 2020. On sampling from a log-concave density using kinetic Langevin diffusions. *Bernoulli* 26, 3 (2020), 1956–1988.

Arnak S. Dalalyan and Alexandre B. Tsybakov. 2012. Sparse regression learning by aggregation and Langevin Monte-Carlo. *J. Comput. System Sci.* 78, 5 (2012), 1423–1443.

Prathamesh Dharangutte and Christopher Musco. 2021. Dynamic trace estimation. In *Advances in Neural Information Processing Systems 34*. 30088–30099.

Zhiyan Ding, Qin Li, Jianfeng Lu, and Stephen J. Wright. 2021. Random coordinate Langevin Monte Carlo. In *Conference on Learning Theory*. PMLR, 1683–1710.

Alain Durmus, Szymon Majewski, and Błażej Miasojedow. 2019. Analysis of Langevin Monte Carlo via convex optimization. *J. Mach. Learn. Res.* 20 (2019), Paper No. 73, 46.

Alain Durmus and Eric Moulines. 2017. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability* 27, 3 (2017), 1551–1587.

Zeev Dvir. 2009. On the size of Kakeya sets in finite fields. *Journal of the American Mathematical Society* 22, 4 (2009), 1093–1097.

Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, and Bin Yu. 2018. Log-concave sampling: Metropolis–Hastings algorithms are fast!. In *Conference on Learning Theory*. PMLR, 793–797.

Alan Edelman. 1989. *Eigenvalues and condition numbers of random matrices*. Ph. D. Dissertation. Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA.

Jiaojiao Fan, Bo Yuan, and Yongxin Chen. 2023. Improved dimension dependence of a proximal algorithm for sampling. In *Proceedings of Thirty Sixth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 195)*, Gergely Neu and Lorenzo Rosasco (Eds.). PMLR, 1473–1521.

Khashayar Gatmiry and Santosh S. Vempala. 2022. Convergence of the Riemannian Langevin algorithm. *arXiv e-prints*, Article arXiv:2204.10818 (2022).

Rong Ge, Holden Lee, and Jianfeng Lu. 2020. Estimating normalizing constants for log-concave distributions: algorithms and lower bounds. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 579–586.

Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. 2022. Private convex optimization via exponential mechanism. In *Proceedings of Thirty Fifth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 178)*, Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 1948–1989.

Michael F. Hutchinson. 1990. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation* 19 (1990), 433–450. Issue 2.

Qijia Jiang. 2021. Mirror Langevin Monte Carlo: the case under isoperimetry. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 715–725.

Tiefeng Jiang and Danning Li. 2015. Approximation of rectangular beta-Laguerre ensembles and large deviations. *J. Theoret. Probab.* 28, 3 (2015), 804–847.

Richard Jordan, David Kinderlehrer, and Felix Otto. 1998. The variational formulation of the Fokker–Planck equation. *SIAM J. Math. Anal.* 29, 1 (1998), 1–17.

Stasys Jukna. 2011. *Extremal combinatorics: with applications in computer science*. Vol. 571. Springer.

Yin Tat Lee, Ruoqi Shen, and Kevin Tian. 2020. Logsmooth gradient concentration and tighter runtimes for Metropolized Hamiltonian Monte Carlo. In *Conference on Learning Theory*. PMLR, 2565–2597.

Yin Tat Lee, Ruoqi Shen, and Kevin Tian. 2021a. Lower bounds on Metropolized sampling methods for well-conditioned distributions. *Advances in Neural Information Processing Systems* 34 (2021), 18812–18824.

Yin Tat Lee, Ruoqi Shen, and Kevin Tian. 2021b. Structured logconcave sampling with a restricted Gaussian oracle. In *Conference on Learning Theory*. PMLR, 2993–3050.

Ruilin Li, Molei Tao, Santosh S. Vempala, and Andre Wibisono. 2022. The mirror Langevin algorithm converges with vanishing bias. In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory (Proceedings of Machine Learning Research, Vol. 167)*, Sanjoy Dasgupta and Nika Haghtalab (Eds.). PMLR, 718–742.

László Lovász and Santosh Vempala. 2006. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. System Sci.* 72, 2 (2006), 392–417.

Yi-An Ma, Niladri S. Chatterji, Xiang Cheng, Nicolas Flammarion, Peter L. Bartlett, and Michael I. Jordan. 2021. Is there an analog of Nesterov acceleration for gradient-based MCMC? *Bernoulli* 27, 3 (2021), 1942–1992.

Raphael A. Meyer, Cameron Musco, Christopher Musco, and David P. Woodruff. 2021. Hutch++: optimal stochastic trace estimation. In *4th Symposium on Simplicity in Algorithms*. SIAM, 142–155.

Dan Mikulincer. 2022. A CLT in Stein's distance for generalized Wishart matrices and higher-order tensors. *Int. Math. Res. Not. IMRN* 10 (2022), 7839–7872.

Cameron Musco and Christopher Musco. 2015. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28*. 1396–1404.

Arkadij S. Nemirovskij and David B. Yudin. 1983. Problem complexity and method efficiency in optimization. (1983).

Yurii Nesterov. 2018. *Lectures on convex optimization*. Springer Optimization and Its Applications, Vol. 137. Springer, Cham. xxiii+589 pages.

Akihiko Nishimura and Marc A. Suchard. 2022. Prior-preconditioned conjugate gradient method for accelerated Gibbs sampling in "large n, large p" Bayesian sparse regression. *J. Amer. Statist. Assoc.* 0, 0 (2022), 1–14.

Oskar Perron. 1928. Über einen Satz von Besicovitsch. *Mathematische Zeitschrift* 28, 1 (1928), 383–386.

Miklós Z. Rácz and Jacob Richey. 2019. A smooth transition from Wishart to GOE. *J. Theoret. Probab.* 32, 2 (2019), 898–906.

Luis Rademacher and Santosh Vempala. 2008. Dispersion of mass and the complexity of randomized geometric algorithms. *Adv. Math.* 219, 3 (2008), 1037–1069.

Cyrus Rashtchian, David P. Woodruff, and Hanlin Zhu. 2020. Vector-matrix-vector queries for solving linear algebra, statistics, and graph problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (LIPIcs, Vol. 176)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 26:1–26:20.

Christian P. Robert and George Casella. 2004. *Monte Carlo statistical methods* (second ed.). Springer-Verlag, New York. xxx+645 pages.

Sushant Sachdeva and Nisheeth K. Vishnoi. 2014. Faster algorithms via approximation theory. *Found. Trends Theor. Comput. Sci.* 9, 2 (2014), 125–210.

Adil Salim and Peter Richtarik. 2020. Primal dual interpretation of the proximal stochastic gradient Langevin algorithm. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 3786–3796.

Shubhangi Saraf and Madhu Sudan. 2008. An improved lower bound on the size of Kakeya sets over finite fields. *Analysis & PDE* 1, 3 (2008), 375–379.

Ruoqi Shen and Yin Tat Lee. 2019. The randomized midpoint method for log-concave sampling. *Advances in Neural Information Processing Systems* 32 (2019).

Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. 2018. Tight query complexity lower bounds for PCA via finite sample deformed Wigner law. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 1249–1259.

Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. 2019. Querying a matrix through matrix-vector products. In *46th International Colloquium on Automata, Languages, and Programming (LIPIcs, Vol. 132)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 94:1–94:16.

Stanisław J. Szarek. 1991. Condition numbers of random matrices. *J. Complexity* 7, 2 (1991), 131–149.

Kunal Talwar. 2019. Computational separations between sampling and optimization. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.

Santosh S. Vempala and Andre Wibisono. 2019. Rapid convergence of the unadjusted Langevin algorithm: isoperimetry suffices. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8094–8106.

Roman Vershynin. 2018. *High-dimensional probability*. Cambridge Series in Statistical and Probabilistic Mathematics, Vol. 47. Cambridge University Press, Cambridge. xiv+284 pages. An introduction with applications in data science, With a foreword by Sara van de Geer.

Andre Wibisono. 2018. Sampling as optimization in the space of measures: the Langevin dynamics as a composite optimization problem. In *Conference on Learning Theory*. PMLR, 2093–3027.

Andre Wibisono. 2019. Proximal Langevin algorithm: rapid convergence under isoperimetry. *arXiv preprint arXiv:1911.01469* (2019).

Karl Wimmer, Yi Wu, and Peng Zhang. 2014. Optimal query complexity for estimating the trace of a matrix. In *41st International Colloquium on Automata, Languages, and Programming (Lecture Notes in Computer Science, Vol. 8572)*. Springer, 1051–1062.

David P. Woodruff. 2014. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.* 10, 1-2 (2014), 1–157.

Blake Woodworth and Nathan Srebro. 2017. Lower bound for randomized first order convex optimization. *arXiv e-prints*, Article arXiv:1709.03594 (2017).

Keru Wu, Scott Schmidler, and Yuansi Chen. 2022. Minimax mixing time of the Metropolis-adjusted Langevin algorithm for log-concave sampling. *Journal of Machine Learning Research* 23, 270 (2022), 1–63.

Kelvin S. Zhang, Gabriel Peyré, Jalal Fadili, and Marcelo Pereyra. 2020. Wasserstein control of mirror Langevin Monte Carlo. In *Proceedings of Thirty Third Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 125)*, Jacob Abernethy and Shivani Agarwal (Eds.). PMLR, 3814–3841.

A Upper bound for log-concave sampling in constant dimension

In this section we give a simple proof that in constant dimension, one can approximately generate a sample from a log-concave distribution with condition number κ, in $O(\log \kappa)$ queries. Our query dependence also has a polylogarithmic dependence on $\frac{1}{\varepsilon}$, if we wish to generate a sample that is ε-close in TV distance to the true distribution. (We do not attempt to optimize the dependence on dimension d or the polylogarithmic dependence on $\frac{1}{\varepsilon}$.)

Let V be a convex function that is 1-strongly convex and κ-smooth, such that V is minimized at the origin and $V(0) = 0$. For any real value $y \geq 0$, define $B_V(y)$ to be the set of points x such that $V(x) \leq y$.

First, we note the following basic facts that follow immediately from our convexity assumptions.

Proposition A.1 (basic facts about log-concavity). *(1) $B_V(y)$ is a convex body for any $y > 0$, and contains 0.*
(2) $B_V(y)$ is contained in the ball of radius $\sqrt{2y}$ and contains the ball of radius $\sqrt{2y/\kappa}$.
(3) For any $0 < y < y'$, $B_V(y') \subset \frac{y'}{y} B_V(y)$.

Next, we show how to obtain a crude $d^{O(1)}$-approximation for $B_V(1)$ using $d^{O(1)} \log \kappa$ first-order queries. The proof is essentially folklore and follows from the ellipsoid method.

Proposition A.2 (ellipsoid method). *Let B be a convex body that contains $B(0, r)$ and is contained in $B(0, R)$, along with a membership and separation oracle. Using $d^{O(1)} \log \frac{R}{r}$ adaptive queries to the membership and separation oracle, we can find an ellipsoid E centered around some point z such that $E \subset B \subset E'$, where E' is E dilated by an $O(d^{3/2})$ factor about z.*

We can apply the above proposition to the convex body $B_V(1)$.

Corollary A.3 (sublevel set approximation). *Using $d^{O(1)}\log\kappa$ adaptive queries to V and ∇V, we can find an ellipsoid E centered around some point z such that $E \subset B_V(1) \subset E'$, where E' is E dilated by an $O(d^{3/2})$ factor about z.*

Proof. It suffices to show that from a single first-order query at a point x, we can generate a membership and separation oracle for $B_V(1)$. Indeed, the membership part is straightforward as we just check whether $V(x) \le 1$ (which is equivalent to $x \in B_V(1)$). The separation oracle is also simple, and can be done using the gradient. Specifically, suppose that $V(x) > 1$; then, $V(x') \ge V(x) + \langle \nabla V(x), x' - x\rangle$, which means that every x' with $\langle \nabla V(x), x'\rangle \ge \langle \nabla V(x), x\rangle$ is such that $V(x') \ge V(x) > 1$, i.e., $\nabla V(x)$ is a separation oracle for $B_V(1)$ at x. □

We are able to prove our sampling upper bound, using a rejection sampling approach.

Theorem A.4 (upper bound for log-concave sampling). *For any constant $d \ge 2$ and any 1-strongly convex and κ-smooth function V with minimum at 0, we can approximately sample from $\pi \propto \exp(-V)$ to total variation error at most ε using $O(\log\kappa + \log^{O(1)}(1/\varepsilon))$ adaptive queries to V and ∇V (here we emphasize that the asymptotic notation treats d as constant).*

Proof. Given V and any integer $t \ge 1$, let p_t be the probability that a sample from π lies in $tB_V(1)$. The normalizing constant is $Z \ge \int_{B_V(1)} \exp(-V) \ge e^{-1}\operatorname{vol}(B_V(1))$, but integral over $(t+1)B_V(1)\backslash tB_V(1)$ is

$$\int_{(t+1)B_V(1)\backslash tB_V(1)} \exp(-V) \le \exp(-t)\operatorname{vol}\big((t+1)B_V(1)\big) = \exp(-t)\,(t+1)^d\operatorname{vol}\big(B_V(1)\big)\,,$$

using Proposition A.1 which implies that $V(x) \ge t$ for any $x \notin tB_V(1)$. Therefore, the probability of $(t+1)B_V(1)\backslash tB_V(1)$ under π is at most

$$\pi\big((t+1)B_V(1)\backslash tB_V(1)\big) \le \frac{\exp(-t)\,(t+1)^d\operatorname{vol}(B_V(1))}{e^{-1}\operatorname{vol}(B_V(1))} = \exp(-(t-1))\,(t+1)^d\,.$$

By summing this quantity for all integers greater than t, the probability of the complement of $tB_V(1)$ is at most $\sum_{u\ge t}\exp(-(u-1))\,(u+1)^d = \sum_{u\ge t}\exp(-u + d\log(u+1) + 1)$. Note that for $t \ge \Omega(d\log d)$, this quantity is at most $O(\exp(-t/2))$. Taking $t = C\,(d\log d + \log(1/\varepsilon))$ for a large constant C, we obtain $\pi(\mathbb{R}^d\backslash tB_V(1)) \le \varepsilon/2$.

The algorithm now works as follows. We use Corollary A.3 to find $E \subset B_V(1) \subset E'$. We pick a uniformly random point X in tE' for $t = C\,(d\log d + \log(1/\varepsilon))$. We then accept the point X with probability $\exp(-V(X))$, and if we reject we restart the procedure. First, note that this algorithm, upon termination, samples exactly from π conditioned on tE', which is at most $\frac{\varepsilon}{2}$ away from π in total variation distance. In addition, each rejection sampling step succeeds with probability at least $\operatorname{vol}(E)/(e\operatorname{vol}(tE'))$, since with probability $\operatorname{vol}(E)/\operatorname{vol}(tE')$ we choose a point in E in which case $V(X) \le 1$ so we accept with probability at least e^{-1}. This is equal to $1/(t\,O(d^{3/2}))^d = d^{-O(d)}\,t^{-d} = d^{-O(d)}\,(\log\frac{1}{\varepsilon})^{-d}$. So, after $(d\log\frac{1}{\varepsilon})^{O(d)}$ rounds of rejection sampling, each of which only needs one query to V, we accept the sample with probability at least $1-\frac{\varepsilon}{2}$, which means that overall we have generated a sample which is ε-close in distribution to π in total variation distance.

The overall query complexity is a combination of finding E, E' and then running the rejection sampling, for a total complexity of $d^{O(1)}\log\kappa + (d\log\frac{1}{\varepsilon})^{O(d)}$. So, for any fixed dimension d and error probability ε, the query complexity for log-concave sampling is $O(\log\kappa)$. In addition, the dependence on the error probability is polylogarithmic for any fixed d. □

Remark A.5. We briefly note that the exponential dependence on d is not necessary: using more sophisticated tools developed for sampling from convex bodies one should be able to obtain a complexity of $\log(\kappa)\,(d\log\frac{1}{\varepsilon})^{O(1)}$. However, we choose to not optimize the dimension dependence in this result for the sake of simplicity, and since we are focused on the setting of $d = O(1)$.

B Upper bound for sampling from Gaussians

Finally, we show a simple proof that, using only $O(\min(\sqrt{\kappa}\log d, d))$ gradient queries, one can generate an approximate sample from a Gaussian $\mathcal{N}(0,\Sigma)$ in d dimensions. Note that the density evaluated at x, up to an additive constant, equals $-\frac{1}{2}\,x^\top \Lambda x$ for $\Lambda = \Sigma^{-1}$, which means that a gradient query at x amounts to receiving the matrix-vector product Λx.

First, we require a well-known proposition from approximation theory.

Proposition B.1 ([Sachdeva and Vishnoi 2014, Theorem 3.3]). *For any positive integer s and $0 < \delta < 1$, there exists a polynomial $p_{s,\delta}$ of degree $\lceil \sqrt{2s\ln(2/\delta)}\rceil$ such that $|p_{s,\delta}(x) - x^s| \le \delta$ for all $x \in [-1,1]$.*

As a corollary, we have the following result.

Proposition B.2 (polynomial approximation of inverse square root). *For any $\kappa \ge 2$ and $\delta < \frac{1}{2}$, there exists a polynomial $q_{\kappa,\delta}$ of degree $O(\sqrt{\kappa}\log\frac{\kappa}{\delta})$ such that $|q_{\kappa,\delta}(x) - x^{-1/2}| \le \delta/\sqrt{\kappa}$ for all $1 \le x \le \kappa$.*

Proof. First, consider the function $(1+x)^{-1/2}$. For $|x| \le 1 - \frac{1}{\kappa} < 1$, we can use the Taylor series to write

$$(1+x)^{-1/2} = 1 + \sum_{t=1}^{\infty} \frac{(\frac{1}{2}-1)\,(\frac{1}{2}-2)\,(\frac{1}{2}-3)\cdots(\frac{1}{2}-t)}{t!}\,x^t = 1 + \sum_{i=1}^{\infty} c_t x^t\,,$$

where $|c_t| \le 1$ for all $t \ge 1$.

Note that for $|x| \le 1 - \frac{1}{\kappa}$, $\left|\sum_{t>T} c_t x^t\right| \le \sum_{t>T} |x|^t \le \frac{|x|^T}{1-|x|}$. For $T = O(\kappa \log\frac{\kappa}{\delta})$, we can bound this by $\frac{(1-1/\kappa)^T}{1/\kappa} \le \frac{\delta}{2}$. Therefore, for all such x,

$$\Bigl|(1+x)^{-1/2} - \sum_{t=0}^{T} c_t x^t\Bigr| \le \frac{\delta}{2}\,,$$

where we have set $c_0 \coloneqq 1$.

Next, using Proposition B.1, we can replace each x^t with $p_{t,\delta}(x)$ where $p_{t,\delta}$ is a polynomial of degree $O(\sqrt{t\log(t/\delta)})$ such that $|p_{t,\delta}(x) - x^t| \le \delta/(4t^2)$ for all $|x| \le 1$. (We also let $p_{0,\delta}$ simply be the constant function 1.) Therefore,

$$\Bigl|(1+x)^{-1/2} - \sum_{t=0}^{T} c_t p_{t,\delta}(x)\Bigr| \le \frac{\delta}{2} + \sum_{t=1}^{T} |c_t|\,\frac{\delta}{4t^2} \le \delta\,.$$

In addition, the polynomial $\hat{p} \coloneqq \sum_{t=0}^{T} c_t p_{t,\delta}$ has degree at most $O(\sqrt{T\log(T/\delta)}) = O(\sqrt{\kappa}\log\frac{\kappa}{\delta})$.

To finish, $|\hat{p}(x-1) - x^{-1/2}| \le \frac{\delta}{\kappa}$ for all $\frac{1}{\kappa} \le x \le 1$, which means that

$$\Bigl|\hat{p}\bigl(\frac{x}{\kappa} - 1\bigr)\,\frac{1}{\sqrt{\kappa}} - x^{-1/2}\Bigr| \le \frac{\delta}{\sqrt{\kappa}} \qquad \text{for all } 1 \le x \le \kappa\,.$$

So, there exists a polynomial $q_{\kappa,\delta}$ with $q_{k,\delta}(x) = \hat{p}(\frac{x}{\kappa} - 1)\,\frac{1}{\sqrt{\kappa}}$, such that $q_{\kappa,\delta}$ has degree $O(\sqrt{\kappa}\log\frac{\kappa}{\delta})$ and $|q_{\kappa,\delta}(x) - x^{-1/2}| \le \delta/\sqrt{\kappa}$ for all $1 \le x \le \kappa$. □

We are now ready to prove our query complexity upper bound.

Theorem B.3 (optimal algorithm for sampling from Gaussians). *Let $\Lambda = \Sigma^{-1}$ be an unknown positive definite matrix with all eigenvalues between 1 and κ. Then, using $O(\min(\sqrt{\kappa}\log\frac{d}{\varepsilon}, d))$ adaptive matrix-vector queries to Λ, we can produce a sample from a distribution $\hat{\pi}$ such that $\mathsf{KL}(\hat{\pi} \,\|\, \mathcal{N}(0,\Sigma)) \le \varepsilon^2$.*

Proof. Choose $X \sim \mathcal{N}(0, I_d)$, define $R = O(\sqrt{\kappa} \log \frac{\kappa}{\delta})$ be the degree of $q_{\kappa,\delta}$, and for simplicity write $q(x) \coloneqq q_{\kappa,\delta}(x) \coloneqq \sum_{i=0}^R a_i x^i$. The algorithm works as follows. Using the power method, we compute $X, \Lambda X, \Lambda^2 X, \dots, \Lambda^R X$. We output $Y = \sum_{i=0}^R a_i \, \Lambda^i X$. Note that $Y \sim \mathcal{N}(0, \hat{\Sigma})$, where we set $\hat{\Sigma} \coloneqq (\sum_{i=0}^R a_i \Lambda^i)^2$. If $\lambda_1, \dots, \lambda_d$ denote the eigenvalues of Λ, then the eigenvalues of $\hat{\Sigma}$ are $q(\lambda_1), \dots, q(\lambda_d)$. The KL divergence is given by

$$\begin{aligned}\mathsf{KL}\big(\mathcal{N}(0,\hat{\Sigma}) \bigm\| \mathcal{N}(0,\Sigma)\big) &\lesssim \sum_{k=1}^d |q(\lambda_k)^2 \, \lambda_k - 1|^2 \lesssim \sum_{k=1}^d |q(\lambda_k) \, \lambda_k^{1/2} - 1|^2 \lesssim \sum_{k=1}^d \lambda_k \, |q(\lambda_k) - \lambda_k^{-1/2}|^2 \\ &\lesssim d\kappa \, \frac{\delta^2}{\kappa} \, .\end{aligned}$$

If we set $\delta \asymp \log(\varepsilon/d)$, then we obtain a KL divergence of at most ε^2.

Finally, we can also learn Λ by querying Λe_i for each unit basis vector $e_1, \dots, e_d$. So, we can thus learn Σ, and then generate a perfect random sample from $\mathcal{N}(0, \Sigma)$. Hence, the query complexity of generating a sample from $\mathcal{N}(0, \Sigma)$ is at most $O(\min(\sqrt{\kappa} \log \frac{\kappa d}{\varepsilon}, d)) = O(\min(\sqrt{\kappa} \log \frac{d}{\varepsilon}, d))$. □

Remark B.4. If π is an α-strongly log-concave distribution, then from Pinsker's inequality and Talagrand's transport inequality,

$$\max\{\|\mu - \pi\|_{\mathrm{TV}}^2, \ \alpha \, W_2^2(\mu, \pi)\} \lesssim \mathsf{KL}(\mu \parallel \pi) \, .$$

Hence, this algorithmic result for Gaussians complements the two lower bounds in Corollary 4.4.

Received 23 December 2023; revised 25 May 2024; accepted 11 June 2024