Steering diffusion models with quadratic rewards: a fine-grained analysis

Ankur Moitra MOITRA@MIT.EDU
Massachusetts Institute of Technology

Andrej Risteski ARISTESK@ANDREW.CMU.EDU
Carnegie Mellon University

Dhruv Rohatgi DROHATGI@MIT.EDU
Massachusetts Institute of Technology

Editors: Steve Hanneke and Tor Lattimore

Abstract

Inference-time algorithms are an emerging paradigm in which pre-trained models are used as subroutines to solve downstream tasks. Such algorithms have been proposed for tasks ranging from inverse problems and guided image generation to reasoning. However, the methods currently deployed in practice are heuristics with a variety of failure modes—and we have very little understanding of when these heuristics can be efficiently improved.

In this paper, we consider the task of sampling from a reward-tilted diffusion model—that is, sampling from $p^\star(x) \propto p(x)\exp(\mathbf{r}(x))$—given a reward function $\mathbf{r}$ and pre-trained diffusion oracle for p. We provide a fine-grained analysis of the computational tractability of this task for quadratic rewards $\mathbf{r}(x) = x^\top Ax + b^\top x$. We show that linear-reward tilts are always efficiently sampleable—a simple result that seems to have gone unnoticed in the literature. We use this as a building block, along with a conceptually new ingredient—the Hubbard-Stratonovich transform—to provide an efficient algorithm for sampling from low-rank positive-definite quadratic tilts, i.e. $\mathbf{r}(x) = x^\top Ax$ where A is positive-definite and of rank $O(1)$. For negative-definite tilts, i.e. $\mathbf{r}(x) = -x^\top Ax$ where A is positive-definite, we prove that the problem is intractable even if A is of rank 1 (albeit with exponentially-large entries).

Keywords: diffusion models; inference-time algorithms; reward-tilted sampling;

1. Introduction

It is increasingly common to use pre-trained generative models as components within algorithms for more complex downstream tasks. Broadly, such algorithms are termed as *inference-time* or *meta-generation* algorithms (Welleck et al., 2024). Examples of tasks that can be framed in this paradigm include *inverse problems* in the sciences (Bruna and Han, 2024), guidance to perform conditional sampling in image generation (Dhariwal and Nichol, 2021), and tilting a distribution by a (trained or pre-specified) reward in reasoning tasks (Korbak et al., 2022; Geuter et al., 2025) and even protein design (Lisanza et al., 2025; Hartman et al., 2025).

In this paper, we focus on the algorithmic task of steering a pre-trained diffusion model according to a given reward function. Specifically, given a diffusion model for a *base distribution* $p(x)$ (which provides access to the scores of convolutions of p with Gaussian noise), and a reward function $\mathbf{r}(x)$, our task is to sample from the *tilted distribution*

$$p^\star(x) \propto p(x)\exp(\mathbf{r}(x)).$$

© 2026 A. Moitra, A. Risteski & D. Rohatgi.

For any reward function $\mathbf{r}$, $p^\star$ has a natural variational interpretation as the solution to the KL-regularized optimization problem $\arg\max_q \mathbb{E}_q[\mathbf{r}] - D_{\mathsf{KL}}(q\|p)$ (Korbak et al., 2022). Moreover, with appropriate choices of the reward function—ranging from simple quadratics to complex pre-trained reward models—the task of sampling from $p^\star$ formalizes many concrete practical problems, including inpainting (Karan et al., 2025), posterior inference under noisy measurements (Bruna and Han, 2024), and human preference alignment (Singhal et al., 2025). Empirically, the algorithms used are often heuristics, and have known failure modes (Chidambaram et al., 2024). Theoretically, the algorithmic landscape for this task, and the fundamental computational barriers, remain largely unexplored—even when the reward function is very simple.

In this paper, we focus on the family of quadratic reward functions $\mathbf{r}(x) = x^\top Ax + b^\top x$. With this family of rewards, the steering task already encapsulates several of the preceding applications (Karan et al., 2025; Bruna and Han, 2024). For example, posterior inference with linear measurements and Gaussian noise corresponds to steering with the log-density of an appropriate Gaussian: $\mathbf{r}(x) = -\frac{1}{2\sigma^2}\|Mx - y\|_2^2$. Moreover, quadratic bonuses of the form $\mathbf{r}(x) = x^\top \Sigma^{-1} x$, for positive-definite Σ, are commonly used in applications such as rare-event sampling (Asmussen et al., 2011) and reinforcement learning (Tuyls et al., 2025), to steer towards (or optimize for) rare or novel generations.

From a theoretical perspective, steering with quadratic rewards is computationally intractable with no further assumptions (Gupta et al., 2024; Bruna and Han, 2024). Indeed, if the base distribution $p(x)$ is uniform over the *discrete hypercube*, then $p^\star(x) \propto \exp(x^\top Ax + b^\top x)$ is precisely a classical *Ising model* (Ising, 1925), and a seminal line of work has shown that exact and approximate sampling from an Ising model can be intractable (Jerrum and Sinclair, 1993; Sly and Sun, 2012; Galanis et al., 2016). However, there is also a rich literature on efficient algorithms for sampling from Ising models with special structure (Jerrum and Sinclair, 1993; Eldan et al., 2022; Koehler et al., 2022; Chen and Eldan, 2022). Can these sorts of structural assumptions help explain when steering a general base measure p accessible via a diffusion model is indeed tractable?

1.1. Contributions

In this paper, we provide a fine-grained understanding of the computational landscape of steering diffusion models with quadratic rewards $\mathbf{r}(x) := x^\top Ax + b^\top x$. Through the lens of the *rank* of the quadratic form (i.e. $\mathrm{rank}(A)$)—a fundamental quantity in the special case of sampling from Ising models (Koehler et al., 2022)—we delineate regimes in which the task is computationally tractable, and regimes in which it is not. Precisely, we show the following:

Linear rewards admit an efficient sampler (Section 3). If the reward is a linear function $\mathbf{r}(x) = b^\top x$ for some vector $b \in \mathbb{R}^d$ (in other words, $\mathrm{rank}(A) = 0$), we can efficiently sample from the tilt $p^\star$. This relatively simple result seemingly has not been explicitly observed in prior literature; it is a consequence of the fact that the scores of $p^\star$ have a simple closed-form expression in terms of the reward function and scores of the base distribution p.

Negative-definite quadratic rewards induce intractability, even for rank-1 matrices (Section 4). If $\mathbf{r}(x) = x^\top Ax$, where A is a rank-1 negative semi-definite matrix, then the task of sampling from $p^\star$ is computationally intractable, assuming $\mathrm{NP} \not\subseteq \mathrm{BPP}$. In fact the result holds even when A is further restricted to be a diagonal matrix.

Positive-definite, low-rank quadratic rewards admit an efficient sampler (Section 5). If $\mathbf{r}(x) = x^\top Ax$, where A is a rank-$O(1)$ positive semi-definite matrix, then there is an algorithm that samples

from p^* (approximately, in Wasserstein distance) in polynomial time.[1] For completeness, we also show that without the low-rank assumption, the problem becomes computationally intractable (Appendix A). The algorithmic result is based on the idea of using the *Hubbard-Stratonovich* transform (Hubbard, 1959) to construct a lifting of the target distribution (i.e. introduce a new variable). We show how to sample from this lifted distribution using *sampling from linear-reward tilts* as a subroutine. We believe this result is of additional conceptual interest as it shows that sampling from linear-reward tilts can be a useful building block for designing inference-time algorithms even for more complex reward models.

1.2. Related work

Several prior works have studied the problem of provably steering diffusion models with a quadratic reward function $r(x) = -\frac{1}{2\sigma^2}\|Ax - b\|^2$ (Gupta et al., 2024; Bruna and Han, 2024; Xun et al., 2025; Parulekar et al., 2025), motivated by the task of linear inverse problems, i.e. posterior inference with noisy linear measurements. Gupta et al. (2024) show that steering with general (i.e. potentially high-rank) negative-definite quadratic rewards is cryptographically hard, and Bruna and Han (2024) show that the same problem is intractable via reduction from the problem of sampling Ising models. Our result in Section 4 strengthens these by showing that intractability holds even when the quadratic form is rank-1.

Bruna and Han (2024) show that the hardness can be circumvented when σ is sufficiently small. In the context of inverse problems, A corresponds to a measurement operator, and σ to a signal-to-noise ratio; the small-σ regime is easier since it makes $p^\star$ more log-concave. The main tool they use is the Polchinsky flow (Bauerschmidt et al., 2024). Xun et al. (2025) develop an efficient algorithm for the same problem, without assumptions on A and σ, but require that the base distribution p satisfies a condition called "local log-concavity". Parulekar et al. (2025) develop an efficient algorithm with no assumptions, but it (necessarily) has no guarantee of closeness in total variation or Wasserstein distance.

Zooming out, Chidambaram et al. (2024) analyze a popular heuristic for guidance—i.e. sampling from a class-conditioned diffusion model—and show some natural examples in which it has the intended behavior, and some examples of failure modes. Karan et al. (2025) provide a wrapper for existing heuristics, which admits a theoretical guarantee that is weaker than approximate sampling. Rohatgi et al. (2025) consider sampling from tilted distributions when p is the law of an autoregressive language model, and reward access is augmented by a "process reward" that estimates the quality of partial generations.

At a technical level, our application of Hubbard-Stratonovich in the PSD setting is inspired by that of Koehler et al. (2022) for sampling from (approximately) low-rank Ising models, given access to their unnormalized density. Their algorithm uses the transform in conjunction with techniques from Markov Chains and variational inference, whereas ours requires building on our algorithm for linear rewards.

Finally, we remark that the identity in Lemma 3.3 (which we use to show that steering with linear rewards is tractable) is also implicit in the recent work of Holderrieth et al. (2025), albeit in different language and for a different purpose.

1. We remark that the runtime also scales polynomially in $\|A\|_2$. This leaves a conceptual gap, since our hardness result for negative-definite rewards does not rule out an algorithm with similar scaling. We believe such an algorithm is unlikely to exist, but we defer resolution of this question to future work.

2. Preliminaries and notation

Notation. For a set S, let $\Delta(S)$ denote the space of distributions over S. For $R > 0$, let $\mathscr{B}_{d,2}(R) := \{x \in \mathbb{R}^d : \|x\|_2 \le R\}$. For a distribution p, $\mathrm{supp}(p)$ denotes its support. For distributions p, q, $\mathsf{TV}(p, q)$ denotes total variation distance and $\mathsf{W}_2(p, q)$ denotes Wasserstein-2 distance. We let $\mathcal{N}(\mu, \Sigma)$ denote the Gaussian distribution with mean μ and covariance Σ.

Definition 2.1 (Noised distribution) *For any distribution $q \in \Delta(\mathbb{R}^d)$, for any $\sigma \in [0, 1]$, we define $q_\sigma \in \Delta(\mathbb{R}^d)$ as the law of $\sqrt{1-\sigma^2}X + \sigma Z$ where $X \sim q$ and $Z \sim \mathcal{N}(0, I_d)$.*

Formal setting. Let $p \in \Delta(\mathbb{R}^d)$ be a *base distribution* and let $\mathbf{r}(x) : \mathbb{R}^d \to \mathbb{R}$ be a reward function. Our goal is to (approximately) sample from the *tilted distribution* $p^\star \in \Delta(\mathbb{R}^d)$ defined by:

$$p^\star(x) \propto p(x) \exp(\mathbf{r}(x)),$$

which is well-defined whenever $\mathbb{E}_{x\sim p}[\exp(\mathbf{r}(x))] < \infty$. We make the following standard boundedness assumption (De Bortoli, 2022; Chen et al., 2023), which ensures that $p^\star$ is well-defined for any continuous reward; our algorithms will have runtime polynomial in the bound C_{norm}:

Assumption 2.2 (Boundedness) *Let $C_{\mathsf{norm}} \ge 1$. We assume that $\sup_{x\in\mathrm{supp}(p)} \|x\|_2 \le C_{\mathsf{norm}}$.*

We access p via the following oracle, which is exactly the object one would have access to if a diffusion model was (pre-)trained on the distribution p.

Assumption 2.3 (Score oracle) *For any $\sigma \in (0, 1)$ and $x \in \mathbb{R}^d$, we can query $s_\sigma(x) := \nabla \log p_\sigma(x)$.*

Note that we assume exact access to the scores, following prior work (Bruna and Han, 2024; Parulekar et al., 2025); understanding the effect of errors is an interesting open question.

As shown by Chen et al. (2023), under Assumption 2.2, the score oracle enables efficient approximate sampling from the *base distribution* p with small Wasserstein-2 error:

Theorem 2.4 (Chen et al. (2023)) *Let $d, C_{\mathsf{norm}} \in \mathbb{N}$ and $\epsilon > 0$. Fix $q \in \Delta(\mathbb{R}^d)$. Suppose that $\mathrm{supp}(q) \subseteq \mathscr{B}_{d,2}(C_{\mathsf{norm}})$. There is a $\mathrm{poly}(d, \epsilon^{-1}, C_{\mathsf{norm}})$-time algorithm* UnadjustedSampler *that, given parameters $\epsilon, C_{\mathsf{norm}}$ as well as query access to $\nabla \log q_\sigma(x)$ for any $x \in \mathbb{R}^d$ and $\sigma \in (0, 1)$, produces a sample from distribution $\widetilde{q}$ with $\mathsf{W}_2(q, \widetilde{q}) \le \epsilon$ and $\mathrm{supp}(\widetilde{q}) \subseteq \mathscr{B}_{d,2}(C_{\mathsf{norm}})$.*[2]

Under stronger assumptions (e.g. Lipschitzness of the scores), the approximation in Wasserstein distance can be upgraded to approximation in total variation (Chen et al., 2023). However, in this work we focus on the minimal assumptions described above, and seek to approximately sample from $p^\star$ in Wasserstein.

2. The second property is not explicitly stated by Chen et al. (2023), but it is immediate since projection onto $\mathscr{B}_{d,2}(C_{\mathsf{norm}})$ is contractive in ℓ_2.

Algorithm 1 LinTiltSampler: Steering diffusion model with linear reward

1: **input:** Score functions $(s_\sigma)_{\sigma\in(0,1)}$, tilt vector $v \in \mathbb{R}^d$, error tolerance $\epsilon > 0$, norm bound $C_{\mathsf{norm}} \geq 1$.
2: For each $\sigma \in (0,1)$, define $s^\star_\sigma : \mathbb{R}^d \to \mathbb{R}^d$ by

$$s^\star_\sigma(x) := \frac{v}{\sqrt{1-\sigma^2}} + s_\sigma\left(x + \frac{\sigma^2}{\sqrt{1-\sigma^2}}v\right).$$

3: Compute $\widetilde{x} \leftarrow \mathsf{UnadjustedSampler}((s^\star_\sigma)_\sigma, \epsilon, C_{\mathsf{norm}})$. ▷ Theorem 2.4
4: **return:** $\widetilde{x}$.

3. Steering with linear rewards is tractable

In this section we prove Theorem 3.2, which states that if $\mathbf{r}$ is a linear function, there is an efficient approximate sampler LinTiltSampler (Algorithm 1) for $p^\star$. The following definition will be convenient:

Definition 3.1 *Fix $v \in \mathbb{R}^d$. We define $p(\cdot; v) \in \Delta(\mathbb{R}^d)$ by $p(x; v) \propto p(x)e^{\langle x, v\rangle}$.*

Theorem 3.2 *Suppose that Assumption 2.2 holds. Let $v \in \mathbb{R}^d$ and $\epsilon > 0$. The output $\widetilde{x} \leftarrow \mathsf{LinTiltSampler}((s_\sigma)_\sigma, v, \epsilon, C_{\mathsf{norm}})$ has law $\widehat{p}(\cdot; v)$ satisfying $\mathsf{W}_2(\widehat{p}(\cdot; v), p(\cdot; v)) \leq \epsilon$ and $\mathrm{supp}(\widehat{p}(\cdot; v)) \subseteq \mathscr{B}_{d,2}(C_{\mathsf{norm}})$. Moreover, the time complexity of the algorithm is at most $\mathrm{poly}(d, \epsilon^{-1}, C_{\mathsf{norm}})$.*

The key lemma facilitating this result is the following:

Lemma 3.3 *For any $x \in \mathbb{R}^d$ and $\sigma \in [0, 1)$, it holds that*

$$\nabla \log p_\sigma(x; v) = \frac{v}{\sqrt{1-\sigma^2}} + \nabla \log p_\sigma\left(x + \frac{\sigma^2}{\sqrt{1-\sigma^2}}v\right)$$

Proof We explicitly compute the density $p_\sigma(x; v)$. Set $t := \sqrt{1-\sigma^2}$. Then:

$$\begin{aligned}
p_\sigma(x; v) &= \frac{t^{-d}}{(2\pi\sigma^2)^{d/2}} \int_{\mathbb{R}^d} p(t^{-1}y; v) e^{-\|y-x\|_2^2/(2\sigma^2)}\,\mathrm{d}y \\
&\propto \int_{\mathbb{R}^d} p(t^{-1}y) e^{\langle t^{-1}y, v\rangle - \|y-x\|_2^2/(2\sigma^2)}\,\mathrm{d}y \\
&= e^{\langle t^{-1}x, v\rangle} \int_{\mathbb{R}^d} p(t^{-1}y) e^{\langle t^{-1}(y-x), v\rangle - \|y-x\|_2^2/(2\sigma^2)}\,\mathrm{d}y \\
&\propto e^{\langle t^{-1}x, v\rangle} \int_{\mathbb{R}^d} p(t^{-1}y) e^{-\left\|y-x-t^{-1}\sigma^2 v\right\|_2^2/(2\sigma^2)}\,\mathrm{d}y \\
&\propto e^{\langle t^{-1}x, v\rangle} p_\sigma(x + t^{-1}\sigma^2 v)
\end{aligned}$$

where the fourth equality is by completing the square. It follows that

$$\nabla \log p_\sigma(x; v) = \frac{v}{\sqrt{1-\sigma^2}} + \nabla \log p_\sigma\left(x + \frac{\sigma^2}{\sqrt{1-\sigma^2}}v\right)$$

as claimed. ■

The implication of this lemma is that using the score oracle for p, we can efficiently simulate a score oracle for $p^\star := p(\cdot; v)$. We can plug this result into Theorem 2.4 (due to Chen et al. (2023)), which states that for any distribution q with bounded support, given query access to $\nabla \log q_\sigma$ for any $\sigma \in (0,1)$, there is an efficient algorithm that approximately samples from q. The proof of Theorem 3.2 is essentially immediate:

Proof of Theorem 3.2. By Lemma 3.3, the functions $s^\star_\sigma$ defined in Algorithm 1 satisfy $s^\star_\sigma(x) = \nabla \log p_\sigma(x; v)$ for all $x \in \mathbb{R}^d$ and $\sigma \in (0,1)$. Since $\mathrm{supp}(p) \subseteq \mathscr{B}_{d,2}(C_{\mathsf{norm}})$, it is immediate that $\mathrm{supp}(p(\cdot; v)) \subseteq \mathscr{B}_{d,2}(C_{\mathsf{norm}})$. The claim then follows from Theorem 2.4. ■

4. Steering with low-rank negative-definite rewards is hard

In this section, we show that the problem of steering a diffusion model with a low-rank negative-definite reward—specifically, $\mathbf{r}(x) = x^\top A x$, where A is a negative semi-definite matrix of rank 1—is computationally intractable, assuming a standard conjecture (NP $\not\subseteq$ BPP) from computational complexity. This refines a result of Bruna and Han (2024) (who showed this claim for general negative semi-definite matrices A) and combines elements of their analysis with a result of Koehler et al. (2022) (who showed that sampling from rank-1 Ising models is computationally intractable).

Theorem 4.1 *Suppose that there is a randomized algorithm $\mathcal{A}$ with the following property. For any integer $d \in \mathbb{N}$, any distribution $p \in \Delta(\mathbb{R}^d)$ satisfying Assumption 2.2 with parameter $C_{\mathsf{norm}} \geq 1$, and any rank-1, negative semi-definite $A \in \mathbb{R}^{d \times d}$, the output $\widetilde{x} \in \mathbb{R}^d$ of $\mathcal{A}((\nabla \log p_\sigma)_{\sigma \in (0,1)}, A, C_{\mathsf{norm}})$ has law ν satisfying $\mathsf{W}_2(\nu, p^\star) \leq 1/4$, where $p^\star \in \Delta(\mathbb{R}^d)$ is the tilted distribution*

$$p^\star(x) \propto p(x) \exp(x^\top A x).$$

Moreover, the time complexity of $\mathcal{A}((\nabla \log p_\sigma)_{\sigma \in (0,1)}, A, C_{\mathsf{norm}})$ is $\mathrm{poly}(d, C_{\mathsf{norm}})$.

Then, NP $\subseteq$ BPP.

In fact, the result straightforwardly extends to the special case where A is diagonal; see Corollary B.1. As a caveat, Theorem 4.1 does not rule out an algorithm with time complexity that also scales polynomially in $\|A\|_2$. We will prove Theorem 4.1 by reducing from the NP-hard PARTITION problem (Karp, 1975), defined as follows:

Definition 4.2 (PARTITION) *Given integers $w := (a_1, \ldots, a_d) \in \mathbb{Z}^d$, the PARTITION problem is to decide whether there exists $x \in \{\pm 1\}^d$ such that $w^\top x = 0$.*

Given an instance $w \in \mathbb{Z}^d$ of the partition problem, we will define a base distribution $p \in \Delta(\mathbb{R}^d)$ and matrix $A_w \in \mathbb{R}^{d \times d}$ as follows:

$$p := \mathrm{Unif}(\{-1,1\}^d) = 2^{-d} \sum_{x \in \{\pm 1\}^d} \delta_x \in \Delta(\mathbb{R}^d), \qquad A_w := -(d+5)\, w w^\top \preceq 0.$$

For notational convenience, we then define $q_w \in \Delta(\mathbb{R}^d)$ to be the tilted distribution

$$q_w(x) \propto p(x) \exp(x^\top A_w x).$$

Note that since p_σ is a product distribution for each $\sigma \in (0,1)$, and each marginal is a mixture of two Gaussians, the score oracle for p can be simulated efficiently. Moreover, intuitively, the tilted distribution q_w will be concentrated on $x \in \{-1,1\}^d$ with $\langle x, w\rangle \approx 0$. Thus, if ν is close to q_w in Wasserstein distance, then ν will be concentrated near such x.

The following lemma helps formalize this intuition by lower bounding the mass of the tilted distribution on solutions to the PARTITION problem, in the event that the tilt corresponds to a YES instance of the PARTITION problem. We show:

Lemma 4.3 *Given a PARTITION instance $w \in \mathbb{Z}^d$, define the set $S_w := \{x \in \{\pm 1\}^d : w^\top x = 0\}$. Assume $S_w \neq \emptyset$ (i.e. the PARTITION instance is a YES instance). Then $q_w(S_w) \geq \frac{200}{201}$.*

See Section C.1 for the proof. With this lemma in hand, we can prove Theorem 4.1:

Proof of Theorem 4.1. Given a PARTITION instance $w \in \mathbb{Z}^d$, construct p and A_w as above, and consider the corresponding tilted distribution $p^\star = q_w$ supported on $\{\pm 1\}^d$. Set $C_{\mathsf{norm}} := \sqrt{d}$ and note that Assumption 2.2 is satisfied. Define the set $S_w := \{x \in \{\pm 1\}^d : w^\top x = 0\}$ and define

$$R_{S_w} := \{y \in \mathbb{R}^d : \mathrm{sgn}(y) \in S_w\}.$$

For $\sigma \in (0,1)$, let $s_\sigma := \nabla \log p_\sigma$, and note that we can efficiently simulate queries to s_σ for any $x \in \mathbb{R}^d$. To solve the PARTITION problem, we will run the sampler $\mathcal{A}((s_\sigma)_{\sigma\in(0,1)}, A_w, C_{\mathsf{norm}})$ to obtain $Y \in \mathbb{R}^d$. We will then round the output to compute $\widehat{x} := \mathrm{sgn}(Y) \in \{\pm 1\}^d$. We will output YES iff $w^\top \widehat{x} = 0$.

We analyze what happens in the YES and NO cases.

If the PARTITION instance is NO, then $S_w = \emptyset$. In that case, for every $\widehat{x} \in \{\pm 1\}^d$, we have $w^\top \widehat{x} \neq 0$, so the above algorithm always outputs NO.

In the YES case, we have $S_w \neq \emptyset$. Let ν_w be the output distribution of Y. By assumption, $\mathsf{W}_2(\nu_w, q_w) \leq 1/4$. Applying Lemma C.1 with $\mu := q_w$, $\nu := \nu_w$, and $S := S_w$, we obtain

$$\nu_w(R_{S_w}) \;\geq\; q_w(S_w) - (1/4)^2.$$

By Lemma 4.3, $q_w(S_w) \geq \frac{200}{201}$. Plugging in these numbers, we get $\nu_w(R_{S_w}) > 0.9$. Thus, this algorithm decides PARTITION with one-sided error: it always outputs NO on NO instances, and outputs YES on YES instances with probability at least 0.9. Moreover, by assumption on $\mathcal{A}$, the time complexity of this algorithm is $\mathrm{poly}(d)$. It follows that $\mathrm{NP} \subseteq \mathrm{BPP}$. ■

5. Steering with low-rank positive-definite rewards is tractable

In this section, we show that the problem of steering a diffusion model with low-rank *positive-definite* rewards, i.e. with $\mathbf{r}(x) = x^\top A x$ for positive semi-definite low-rank A, is computationally tractable. Without loss of generality, we may write $A = \frac{1}{2} L^\top L$ where L is an $r \times d$ matrix. Then the tilted distribution $p^\star$ is as defined below:

Definition 5.1 *Fix $p \in \Delta(\mathbb{R}^d)$ satisfying Assumption 2.2 with parameter C_{norm}. Fix a matrix $L \in \mathbb{R}^{r \times d}$. We define $p^\star \in \Delta(\mathbb{R}^d)$ by*

$$p^\star(x) := \frac{p(x) e^{\frac{1}{2}\|Lx\|_2^2}}{Z}$$

where $Z := \mathbb{E}_{x \sim p}[e^{\frac{1}{2}\|Lx\|_2^2}]$ is the normalization constant.

Note that Z is finite by Assumption 2.2, and hence $p^\star$ is well-defined. The main result of this section is the following theorem, which shows that PSDTiltSampler (Algorithm 2) samples from $p^\star$ in polynomial time whenever $r = O(1)$ (and C_{norm} and $\|L\|_2$ are polynomially bounded):

Theorem 5.2 *Suppose that Assumption 2.2 holds. Let $D \geq 1$ and $\epsilon_{\mathsf{final}} \in (0, 1/2)$. Suppose that $D \geq \sup_{x \in \mathrm{supp}(p)} \|Lx\|_2$. Then the output $\widetilde{x}$ of $\mathsf{PSDTiltSampler}((s_\sigma)_\sigma, L, D, C_{\mathsf{norm}}, \epsilon_{\mathsf{final}})$ has law μ satisfying $\mathsf{W}_2(\mu, p^\star) \leq \epsilon_{\mathsf{final}}$. Moreover, the time complexity of the algorithm is at most*

$$\mathrm{poly}(d, \|L\|_2, C_{\mathsf{norm}}^r, D^r, r^r, \epsilon_{\mathsf{final}}^{-r}).$$

The key insight is the following decomposition, leveraging the Hubbard-Stratonovich (Hubbard, 1959) transform:

Definition 5.3 *For each $z \in \mathbb{R}^r$, define $Z(z) := \int_{\mathbb{R}^d} p(x) e^{\langle Lx, z \rangle} \, \mathrm{d}x$.*

As before, $Z(z)$ is finite by Assumption 2.2.

Lemma 5.4 *It holds for all $x \in \mathbb{R}^d$ that*

$$p^\star(x) = \frac{(2\pi)^{-r/2}}{Z} \int_{\mathbb{R}^r} Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z) \, \mathrm{d}z.$$

Proof The Hubbard-Stratonovich transform gives

$$e^{\frac{1}{2}\|Lx\|_2^2} = (2\pi)^{-d/2} \int_z e^{-\frac{1}{2}\|z\|_2^2 + \langle Lx, z \rangle} \, \mathrm{d}z.$$

It follows that

$$\begin{aligned} p^\star(x) &= \frac{(2\pi)^{-d/2} \int_{\mathbb{R}^r} e^{-\frac{1}{2}\|z\|_2^2 + \langle Lx, z \rangle} p(x) \, \mathrm{d}z}{Z} \\ &= \frac{(2\pi)^{-d/2} \int_{\mathbb{R}^r} Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z) \, \mathrm{d}z}{Z} \end{aligned}$$

as claimed. ∎

The above decomposition implies that $p^\star$ is the marginal distribution of x under the following lifted distribution in $\mathbb{R}^{d+r}$:

$$q(x, z) \propto Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z).$$

For any fixed z, the conditional distribution $q(x \mid z)$ is precisely $p(x; L^\top z)$, which we can efficiently sample from using Algorithm 1 from Section 3. Thus, it suffices to (approximately) sample from the

Algorithm 2 PSDTiltSampler: Sampling from diffusion model with PSD quadratic tilt

input: Score functions $(s_\sigma)_{\sigma\in(0,1)}$, reward matrix $L \in \mathbb{R}^{r\times d}$, norm bounds $D \geq 1$ and $C_{\mathsf{norm}} \geq 1$, final error tolerance $\epsilon_{\mathsf{final}} \in (0,1)$.
Set $R := D + 2\sqrt{r} + 2\sqrt{\log(180C_{\mathsf{norm}}/\epsilon_{\mathsf{final}}^2)}$, $\gamma := \epsilon_{\mathsf{final}}^2/(180C_{\mathsf{norm}}D)$.
Define $\mathcal{S} := \gamma\mathbb{Z}^r \cap \mathscr{B}_{r,2}(R) \subseteq \mathbb{R}^r$.
Set $N := C_{\mathsf{norm}} R \|L\|_2$, $\epsilon_1 := \epsilon_{\mathsf{final}}^2/(720C_{\mathsf{norm}})$, $\epsilon_2 := \epsilon_{\mathsf{final}}/30$, $\delta_1 := \epsilon_{\mathsf{final}}^2/(720C_{\mathsf{norm}}|\mathcal{S}|)$.
for $z \in \mathcal{S}$ **do**
    $\hat{Z}(z) \leftarrow \mathsf{EstimateNormalization}((s_\sigma)_\sigma, L^\top z, \epsilon_1, \delta_1, C_{\mathsf{norm}})$. ▷ Algorithm 3
Set $\hat{Z} := \sum_{z\in\mathcal{S}} \hat{Z}(z)\exp(-\frac{1}{2}\|z\|_2^2)$ and define $\widehat{p}_z(z) := \hat{Z}(z)\exp(-\frac{1}{2}\|z\|_2^2)/\hat{Z}$.
Sample $\widetilde{z} \sim \widehat{p}_z$.
Sample $\widetilde{x} \leftarrow \mathsf{LinTiltSampler}((s_\sigma)_\sigma, L^\top\widetilde{z}, \epsilon_2, C_{\mathsf{norm}})$. ▷ Algorithm 1
return: $\widetilde{x}$.

Algorithm 3 EstimateNormalization: Estimate normalization for linear tilt

input: Score functions $(s_\sigma)_{\sigma\in(0,1)}$, tilt vector $v \in \mathbb{R}^d$, error tolerance $\epsilon > 0$, failure probability $\delta \in (0,1/2)$, norm bound $C_{\mathsf{norm}} \geq 1$.
Set $N := C_{\mathsf{norm}}\|v\|_2$, $\epsilon' := \frac{\epsilon}{2(1+e)eN}$, and $M := e^{-2}\log(2N/\delta)/(\epsilon')^2$.
for $1 \leq n \leq N$ **do**
    for $1 \leq m \leq M$ **do**
        $x^{(m)} \leftarrow \mathsf{LinTiltSampler}((s_\sigma)_\sigma, \frac{(n-1)}{N}v, \epsilon', C_{\mathsf{norm}})$. ▷ Algorithm 1
    Set $\hat{\kappa}(n) := \frac{1}{M}\sum_{m=1}^M \exp(\langle x^{(m)}, \frac{1}{N}v\rangle)$.
return: $\hat{\kappa} := \prod_{n=1}^N \hat{\kappa}(n)$.

marginal distribution over z, which is precisely $q(z) \propto Z(z)e^{-\frac{1}{2}\|z\|_2^2}$. This is where we exploit the low-rank assumption: since z is r-dimensional, and $p(x; L^\top z)$ satisfies appropriate smoothness in z, it suffices to explicitly estimate the densities $q(z)$ for z on a grid of cardinality (roughly) $\exp(r)$. We accomplish this using EstimateNormalization (Algorithm 3), which approximates $Z(z)$ (given the vector $v = L^\top z$) by telescoping Monte Carlo approximations. See Algorithm 2 for the pseudocode.

Remark 5.5 *A natural alternative approach to sample from $q(x,z)$ would be Gibbs sampling: i.e. alternately sample from $q(x \mid z)$ and $q(z \mid x)$, using that both conditional distributions are tractable. However, examples can be constructed in which the Markov chain does not mix rapidly. In particular, if p is a mixture of two Gaussians defined as $p = \frac{1}{2}\mathcal{N}(-u,\sigma^2 I) + \frac{1}{2}\mathcal{N}(u,\sigma^2 I)$, and we tilt with a quadratic reward $\mathbf{r}(x) = \lambda uu^\top$, it can be seen that $q(z|x) = \mathcal{N}(\sqrt{2\lambda}u^\top x, 1)(z)$ and $q(x|z) \propto p(x)\exp(\sqrt{2\lambda}zu^\top x)$. So, if z is positive, then $q(x|z)$ is biased towards the $+u$ mode, and $q(z|x)$ has positive mean $\approx \sqrt{2\lambda}\|u\|^2$ so it will remain positive. Thus, the tilt induces a metastability of the Gibbs sampler.*

In Section 5.1, we analyze the subroutine EstimateNormalization. In Section 5.2 we complete the proof of Theorem 5.2.

5.1. Analysis of EstimateNormalization

Given $z \in \mathbb{R}^r$, a first attempt at estimating $Z(z)$ would be Monte Carlo estimation using samples from p. However, this could take exponential time in the problem parameters. Instead, we observe that for any sufficiently close $z, z' \in \mathbb{R}^r$, it is possible to efficiently estimate $Z(z)/Z(z')$ using samples from $p(\cdot; L^\top z')$—which we can (approximately) obtain using LinTiltSampler. Thus, we can estimate $Z(z)$ by telescoping along a path from $Z(0) = 1$. This idea is formalized in Algorithm 3 and analyzed in Lemma 5.6 below.

Lemma 5.6 *Let $v \in \mathbb{R}^d$, $\epsilon, \delta \in (0, 1/2)$, and $N \in \mathbb{N}$. Suppose that $N \geq C_{\mathsf{norm}} \|v\|_2$. Then the output $\hat{\kappa} \leftarrow \mathsf{EstimateNormalization}((s_\sigma)_\sigma, v, \epsilon, \delta, C_{\mathsf{norm}})$ satisfies*

$$\frac{\hat{\kappa}}{\mathbb{E}_{x\sim p}[\exp(\langle x, v\rangle)]} \in [1-\epsilon, 1+\epsilon] \tag{1}$$

with probability at least $1-\delta$. Moreover, the time complexity of the algorithm is at most

$$\mathrm{poly}(d, \epsilon^{-1}, C_{\mathsf{norm}}, \|v\|_2, \log(1/\delta)).$$

Proof Fix $1 \leq n \leq N$. Set $u := \frac{(n-1)}{N} v$ and $w := \frac{n}{N} v$. Let $\widehat{p}(\cdot; u)$ denote the law of

$$\mathsf{LinTiltSampler}((s_\sigma)_\sigma, u, \epsilon', C_{\mathsf{norm}}).$$

By Theorem 3.2, we have $\mathsf{W}_2(\widehat{p}(\cdot; u), p(\cdot; u)) \leq \epsilon'$ and $\mathrm{supp}(\widehat{p}(\cdot; u)) \subseteq \mathscr{B}_{d,2}(C_{\mathsf{norm}})$.

Define $\xi^{(m)} := \exp(\langle x^{(m)}, \frac{1}{N} v\rangle)$ for $1 \leq m \leq M$. Observe that $\xi^{(1)}, \ldots, \xi^{(M)}$ are i.i.d. random variables with $\xi^{(m)} \in [0, e]$ almost surely (by assumption on N, and the fact that $\widehat{p}(\cdot; u)$ is supported on $\mathscr{B}_{d,2}(C_{\mathsf{norm}})$), and $\mathbb{E}[\xi^{(m)}] = \mathbb{E}_{x\sim\widehat{p}(\cdot;u)}[\exp(\langle x, \frac{1}{N} v\rangle)]$. It follows from Hoeffding's inequality and choice of M that with probability at least $1 - \delta/N$,

$$\left| \hat{\kappa}(n) - \mathop{\mathbb{E}}_{x\sim\widehat{p}(\cdot;u)}[\exp(\langle x, \frac{1}{N} v\rangle)] \right| \leq \epsilon'.$$

Moreover, since $x \mapsto e^x$ is e-Lipschitz in $(-\infty, 1]$, we can bound

$$\begin{aligned} \left| \mathop{\mathbb{E}}_{x\sim\widehat{p}(\cdot;u)}[\exp(\langle x, \frac{1}{N} v\rangle)] - \mathop{\mathbb{E}}_{x\sim p(\cdot;u)}[\exp(\langle x, \frac{1}{N} v\rangle)] \right| &\leq \frac{e\mathsf{W}_2(\widehat{p}(\cdot; u), p(\cdot; u))\, \|v\|_2}{N} \\ &\leq e\epsilon'. \end{aligned}$$

Combining the preceding bounds gives

$$\begin{aligned} \left| \hat{\kappa}(n) - \frac{\mathbb{E}_{x\sim p}[\exp(\langle x, \frac{n}{N} v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x, \frac{n-1}{N} v\rangle)]} \right| &= \left| \hat{\kappa}(n) - \mathop{\mathbb{E}}_{x\sim p(\cdot;u)}[\exp(\langle x, \frac{1}{N} v\rangle)] \right| \\ &\leq (1+e)\epsilon'. \end{aligned}$$

Since

$$\frac{\mathbb{E}_{x\sim p}[\exp(\langle x, \frac{n}{N} v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x, \frac{n-1}{N} v\rangle)]} = \mathop{\mathbb{E}}_{x\sim p(\cdot;u)}[\exp(\langle x, \frac{1}{N} v\rangle)] \geq 1/e,$$

in the above event we have

$$\left|\frac{\hat{\kappa}(n)\cdot\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n-1}{N}v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n}{N}v\rangle)]}-1\right|\le(1+e)\epsilon'\cdot\frac{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n-1}{N}v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n}{N}v\rangle)]}\le(1+e)e\epsilon'.$$

By the union bound, we have with probability at least $1-\delta$ that

$$\begin{aligned}\hat{\kappa}&=\prod_{n=1}^{N}\hat{\kappa}(n)\\&\le\prod_{n=1}^{N}(1+(1+e)e\epsilon')\frac{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n}{N}v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n-1}{N}v\rangle)]}\\&\le\exp((1+e)e\epsilon'N)\prod_{n=1}^{N}\frac{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n}{N}v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n-1}{N}v\rangle)]}\\&\le(1+\epsilon)\mathop{\mathbb{E}}_{x\sim p}[\exp(\langle x,v\rangle)]\end{aligned}$$

so long as $\epsilon'\le\frac{\epsilon}{2(1+e)eN}$, and similarly

$$\begin{aligned}\hat{\kappa}&\ge\prod_{n=1}^{N}(1-(1+e)e\epsilon')\frac{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n}{N}v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n-1}{N}v\rangle)]}\\&\ge\exp(-2(1+e)e\epsilon'N)\prod_{n=1}^{N}\frac{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n}{N}v\rangle)]}{\mathbb{E}_{x\sim p}[\exp(\langle x,\frac{n-1}{N}v\rangle)]}\\&\ge(1-\epsilon)\mathop{\mathbb{E}}_{x\sim p}[\exp(\langle x,v\rangle)],\end{aligned}$$

which completes the proof of Eq. (1). The time complexity of EstimateNormalization is dominated by MN calls to LinTiltSampler. Thus, the claimed time complexity bound follows from Theorem 3.2 and choice of N, M. ■

5.2. Analysis of PSDTiltSampler

With the analysis of EstimateNormalization in hand, the proof of Theorem 5.2 is straightforward. There are three types of errors in PSDTiltSampler to handle: (1) error from estimation of the normalization constants $Z(z)$, which is bounded using Lemma 5.6; (2) error in sampling from $p(\cdot;\widetilde{z})$, which is bounded using Theorem 3.2, and (3) error due to discretization of the Hubbard-Stratonovich transform, for which the bound is deferred to Lemma D.1. Below, we accumulate these errors to complete the analysis of PSDTiltSampler and prove Theorem 5.2.

Proof of Theorem 5.2. For each $y\in\mathbb{R}^d$ let $\widehat{p}(\cdot;y)$ denote the law of $\mathsf{LinTiltSampler}((s_\sigma)_\sigma,y,\epsilon_2,C_{\mathsf{norm}})$. For the purposes of the analysis, we define distributions $q_1,q_2\in\Delta(\mathbb{R}^d)$ by

$$q_1(x)=\frac{1}{\sum_{z\in\mathcal{S}}Z(z)e^{-\frac{1}{2}\|z\|_2^2}}\sum_{z\in\mathcal{S}}Z(z)e^{-\frac{1}{2}\|z\|_2^2}\widehat{p}(x;L^\top z)$$

and

$$q_2(x) = \frac{1}{\sum_{z\in\mathcal{S}} Z(z)e^{-\frac{1}{2}\|z\|_2^2}} \sum_{z\in\mathcal{S}} Z(z)e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z)$$

We will decompose

$$\begin{aligned}\mathsf{W}_2(\mu, p^\star) &\le \mathsf{W}_2(\mu, q_1) + \mathsf{W}_2(q_1, q_2) + \mathsf{W}_2(q_2, p^\star) \\ &\le 2C_{\mathsf{norm}}\sqrt{\mathsf{TV}(\mu, q_1)} + \mathsf{W}_2(q_1, q_2) + 2C_{\mathsf{norm}}\sqrt{\mathsf{TV}(q_2, p^\star)} \end{aligned} \tag{2}$$

where the second inequality uses the fact that all of the above distributions are supported on $\mathscr{B}_{d,2}(C_{\mathsf{norm}})$. We start by bounding $\mathsf{TV}(\mu, q_1)$. Applying Lemma 5.6 and a union bound over $z \in \mathcal{S}$, we get that in an event $\mathcal{E}$ with probability at least $1 - \delta_1|\mathcal{S}|$, for all $z \in \mathcal{S}$,

$$\frac{\hat{Z}(z)}{Z(z)} = \frac{\hat{Z}(z)}{\mathbb{E}_{x\sim p}[\exp(\langle x, L^\top z\rangle)]} \in [1 - \epsilon_1, 1 + \epsilon_1].$$

Condition on $(\hat{Z}(z) : z \in \mathcal{S})$ and suppose that event $\mathcal{E}$ holds. Let $\nu(\cdot \mid \hat{Z}) \in \Delta(\mathbb{R}^d)$ denote the conditional law of the output $\widetilde{x}$. Then $\nu(\cdot \mid \hat{Z})$ has density

$$\nu(x \mid \hat{Z}) = \sum_{z\in\mathcal{S}} \widehat{p}_z(z)\widehat{p}(x; L^\top z) = \frac{1}{\hat{Z}} \sum_{z\in\mathcal{S}} \hat{Z}(z)e^{-\frac{1}{2}\|z\|_2^2}\widehat{p}(x; L^\top z).$$

Define $f(x) = \sum_{z\in\mathcal{S}} Z(z)e^{-\frac{1}{2}\|z\|_2^2}\widehat{p}(x; L^\top z)$ and $g(x) = \sum_{z\in\mathcal{S}} \hat{Z}(z)e^{-\frac{1}{2}\|z\|_2^2}\widehat{p}(x; L^\top z)$. Then

$$\begin{aligned}\int_x |f(x) - g(x)|\,\mathrm{d}x &= \int_x \sum_{z\in\mathcal{S}} |Z(z) - \hat{Z}(z)|e^{-\frac{1}{2}\|z\|_2^2}\widehat{p}(x; L^\top z)\,\mathrm{d}x \\ &\le \epsilon_1 \int_x \sum_{z\in\mathcal{S}} Z(z)e^{-\frac{1}{2}\|z\|_2^2}\widehat{p}(x; L^\top z)\,\mathrm{d}x \\ &= \epsilon_1 \int_x f(x)\,\mathrm{d}x.\end{aligned}$$

Thus, since $\nu(x \mid \hat{Z}) \propto g(x)$ and $q_1(x) \propto f(x)$, Lemma D.3 implies that $\mathsf{TV}(\nu(\cdot \mid \hat{Z}), q_1) \le 2\epsilon_1$. Since this bound holds for all $\hat{Z} \in \mathcal{E}$, and since $\Pr[\mathcal{E}] \ge 1 - \delta_1|\mathcal{S}|$, we get

$$\mathsf{TV}(\mu, q_1) = \mathsf{TV}(\mathbb{E}[\nu(\cdot \mid \hat{Z})], q_1) \le 2\epsilon_1 + \delta_1|\mathcal{S}|.$$

Next, we bound $\mathsf{W}_2(q_1, q_2)$. By Theorem 3.2 it holds that $\mathsf{W}_2(\widehat{p}(\cdot; L^\top z), p(\cdot; L^\top z)) \le \epsilon_2$ for all $z \in \mathbb{R}^r$. Let $p_z \in \Delta(\mathbb{R}^r)$ be defined by $p_z(z) \propto Z(z)e^{-\frac{1}{2}\|z\|_2^2}$. Then q_1 is the density of the random variable X obtained by sampling $z \sim p_z$ and $X \sim \widehat{p}(\cdot; L^\top z)$. Moreover, q_2 is the density of the random variable Y obtained by sampling $z \sim p_z$ and $Y \sim p(\cdot; L^\top z)$. Since for any fixed z, there is a coupling of X and Y (in the event that z is realized) with $\mathbb{E}[\|X - Y\|_2^2 \mid z] \le \epsilon_2^2$, it follows that there is a coupling of X and Y with $\mathbb{E}[\|X - Y\|_2^2] \le \epsilon_2^2$. Thus, we have $\mathsf{W}_2(q_1, q_2) \le \epsilon_2$.

Finally, we apply Lemma D.1 with parameter $\epsilon := \epsilon_{\mathsf{final}}^2/(180C_{\mathsf{norm}})$. By choice of parameters R and γ, we get that $\mathsf{TV}(q_2, p^\star) \le 20\epsilon = \epsilon_{\mathsf{final}}^2/(9C_{\mathsf{norm}})$. We conclude from Eq. (2) that

$$\mathsf{TV}(\mu, p^\star) \le 2C_{\mathsf{norm}}\sqrt{2\epsilon_1 + \delta_1|\mathcal{S}|} + \epsilon_2 + 2C_{\mathsf{norm}}\sqrt{\epsilon_{\mathsf{final}}^2/(9C_{\mathsf{norm}})} \le \epsilon_{\mathsf{final}}$$

by choice of ϵ_1, ϵ_2, and δ_1. Finally, the time complexity of PSDTiltSampler is dominated by $|\mathcal{S}|$ calls to EstimateNormalization and one call to LinTiltSampler. Note that $|\mathcal{S}| \leq (2R/\gamma)^r \leq \mathrm{poly}(C_{\mathsf{norm}}^r, D^r, r^r, \epsilon_{\mathsf{final}}^{-r})$. The claimed time complexity bound therefore follows from Theorem 3.2 and Lemma 5.6. ■

6. Conclusion

In this paper, we considered the task of sampling from a diffusion model, tilted by a quadratic reward. We provide a fine-grained analysis of the computational tractability of this task through the lens of the rank of the quadratic form. In particular, this task is computationally intractable even for rank-1 negative-definite tilts. For low-rank positive-definite tilts, we give an efficient algorithm based on two conceptually new algorithmic ingredients: sampling from linearly tilted diffusion models, and the Hubbard-Stratonovich transform.

There are many natural directions for further work. Our results, like many other results in the literature, assume that the score oracle is exact—whereas in practice, the oracle we have access to is trained, and thus would have errors. Handling errors is non-trivial because they will be small on average under the base distribution—but algorithms which sample from the tilted distribution may deviate substantially from the trajectories that would arise when sampling from the base distribution.

Towards handling more complex rewards, it would also be interesting to understand what interaction between the structure of the reward and the structure of the base distribution allow for efficient algorithms.

Acknowledgments

We thank a bunch of people and funding agency.

References

Søren Asmussen, José Blanchet, Sandeep Juneja, and Leonardo Rojas-Nandayapa. Efficient simulation of tail probabilities of sums of correlated lognormals. *Annals of Operations Research*, 189(1): 5–23, 2011.

Roland Bauerschmidt, Thierry Bodineau, and Benoit Dagallier. Stochastic dynamics and the polchinski equation: an introduction. *Probability Surveys*, 21:200–290, 2024.

Joan Bruna and Jiequn Han. Provable posterior sampling with denoising oracles via tilted transport. *Advances in Neural Information Processing Systems*, 37:82863–82894, 2024.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zyLVMgsZ0U_.

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

Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? a fine-grained analysis in a simple setting. *Advances in Neural Information Processing Systems*, 37:84968–85005, 2024.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Ronen Eldan, Frederic Koehler, and Ofer Zeitouni. A spectral condition for spectral gap: fast mixing in high-temperature ising models. *Probability theory and related fields*, 182(3):1035–1051, 2022.

Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic ising and hard-core models. *Combinatorics, Probability and Computing*, 25(4):500–559, 2016.

Jonathan Geuter, Youssef Mroueh, and David Alvarez-Melis. Guided speculative inference for efficient test-time alignment of llms. *arXiv preprint arXiv:2506.04118*, 2025.

Shivam Gupta, Ajil Jalal, Aditya Parulekar, Eric Price, and Zhiyang Xun. Diffusion posterior sampling is computationally intractable. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=tp6ruPIfIV.

Erik Hartman, Jonas Wallin, Johan Malmström, and Jimmy Olsson. Controllable protein design through feynman-kac steering. *arXiv preprint arXiv:2511.09216*, 2025.

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky TQ Chen, Yaron Lipman, and Brian Karrer. Glass flows: Transition sampling for alignment of flow and diffusion models. *arXiv preprint arXiv:2509.25170*, 2025.

John Hubbard. Calculation of partition functions. *Physical Review Letters*, 3(2):77, 1959.

Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on computing*, 22(5):1087–1116, 1993.

Aayush Karan, Kulin Shah, and Sitan Chen. Reguidance: A simple diffusion wrapper for boosting sample quality on hard inverse problems. *arXiv preprint arXiv:2506.10955*, 2025.

Richard M Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1): 45–68, 1975.

Frederic Koehler, Holden Lee, and Andrej Risteski. Sampling approximately low-rank ising models: Mcmc meets variational methods. In *Conference on Learning Theory*, pages 4945–4988. PMLR, 2022.

Tomasz Korbak, Ethan Perez, and Christopher L Buckley. Rl with kl penalties is better viewed as bayesian inference, 2022. *URL https://arxiv. org/abs/2205.11275*, 2022.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel WK Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J Hendel, Miriam K Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, 43(8):1288–1298, 2025.

Advait Parulekar, Litu Rout, Karthikeyan Shanmugam, and Sanjay Shakkottai. Efficient approximate posterior sampling with annealed langevin monte carlo. *arXiv preprint arXiv:2508.07631*, 2025.

Dhruv Rohatgi, Abhishek Shetty, Donya Saless, Yuchen Li, Ankur Moitra, Andrej Risteski, and Dylan J Foster. Taming imperfect process verifiers: A sampling perspective on backtracking. *arXiv preprint arXiv:2510.03149*, 2025.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Jp988ELppQ.

Allan Sly and Nike Sun. The computational hardness of counting in two-spin models on d-regular graphs. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 361–369. IEEE, 2012.

Jens Tuyls, Dylan J Foster, Akshay Krishnamurthy, and Jordan T Ash. Representation-based exploration for language models: From test-time to post-training. *arXiv preprint arXiv:2510.11686*, 2025.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*, 2024.

Zhiyang Xun, Shivam Gupta, and Eric Price. Posterior sampling by combining diffusion models with annealed langevin dynamics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ARZiMmb619.

Appendix A. Steering with general positive-definite rewards is hard

In this section we prove the following hardness result, which shows that the low-rank assumption in Theorem 5.2 cannot be removed.

Theorem A.1 *Suppose that there is a randomized algorithm $\mathcal{A}$ with the following property. For any integer $d \in \mathbb{N}$, any distribution $p \in \Delta(\mathbb{R}^d)$ satisfying Assumption 2.2 with parameter $C_{\mathsf{norm}} \geq 1$, and any positive semi-definite $A \in \mathbb{R}^{d\times d}$, the output $\widetilde{x} \in \mathbb{R}^d$ of $\mathcal{A}((\nabla \log p_\sigma)_{\sigma\in(0,1)}, A, C_{\mathsf{norm}})$ has law ν satisfying $\mathsf{W}_2(\nu, p^\star) \leq 1/4$, where $p^\star \in \Delta(\mathbb{R}^d)$ is the tilted distribution*

$$p^\star(x) \propto p(x)\exp(x^\top Ax).$$

Moreover, the time complexity of $\mathcal{A}((\nabla \log p_\sigma)_{\sigma\in(0,1)}, w, C_{\mathsf{norm}})$ is $\mathrm{poly}(d, C_{\mathsf{norm}}, \|A\|_2)$.

Then, NP $\subseteq$ BPP.

We prove Theorem A.1 by reduction from the NP-hard problem MAX-CUT:

Definition A.2 (MAX-CUT) *Given a graph $G = ([d], E)$ and an integer k, the MAX-CUT problem is to determine whether there is a set $S \subseteq [d]$ such that $c_G(S) \geq k$, where*

$$c_G(S) := |\{(u, v) \in E : (u \in S \text{ and } v \notin S) \text{ or } (u \notin S \text{ and } v \in S)\}|.$$

The proof is analogous to that of Theorem 4.1: given an instance of MAX-CUT, we construct a tilting problem so that $p^\star$ puts most of its mass on solutions to the instance.

Proof of Theorem A.1. We give an algorithm for MAX-CUT using $\mathcal{A}$ as a subroutine. Fix an instance of MAX-CUT, which is described by a graph $G = ([d], E)$ and integer k. Define $p := \mathrm{Unif}(\{0, 1\}^d)$ and $\beta := d + 100$ and $C_{\mathsf{norm}} = \sqrt{d}$ and

$$A := \beta \left(\sum_{(u,v) \in E} (e_{uu} + e_{vv} - e_{uv} - e_{vu}) \right) \in \mathbb{R}^{d \times d}$$

where e_{ij} is the $d \times d$ matrix with a 1 in entry (i, j) and 0 everywhere else. We sample $\widetilde{x} \in \mathbb{R}^d$ from $\mathcal{A}((s_\sigma)_{\sigma \in (0,1)}, A, C_{\mathsf{norm}})$, where $s_\sigma = \nabla \log p_\sigma$, using the fact that p_σ is a product distribution where each marginal is a mixture of two Gaussians, and hence the score $\nabla \log p_\sigma(x)$ can be explicitly evaluated for any x. We then compute $\hat{x} \in \{0, 1\}^d$ defined by $\hat{x}_i := \mathbb{1}[\widetilde{x}_i \geq 1/2]$ for each $i \in [d]$. We return YES if $\beta^{-1} \hat{x}^\top A \hat{x} \geq k$ and NO otherwise.

Analysis. Observe that for any $x \in \mathbb{R}^d$,

$$\beta^{-1} x^\top A x = \sum_{(u,v) \in E} (x_u^2 + x_v^2 - 2x_u x_v) = \sum_{(u,v) \in E} (x_u - x_v)^2 \geq 0$$

and thus A is positive semi-definite. Moreover, if $x = \mathbb{1}[S]$ for some $S \subset [d]$, then we have

$$\beta^{-1} x^\top A x = c_G(S).$$

Define $p^\star \in \Delta(\mathbb{R}^d)$ by

$$p^\star(x) \propto p(x) \exp(x^\top A x).$$

Let $U \subset \{-1, 1\}^d$ be the set of x such that $\beta^{-1} x^\top A x = \max_{y \in \{-1,1\}^d} \beta^{-1} y^\top A y =: c_G$. Then

$$\sum_{x \in U} \exp(x^\top A x) \geq \exp(\beta c_G).$$

Moreover, for any $x \in \{-1, 1\}^d \setminus U$, we have $\beta^{-1} x^\top A x \leq c_G - 1$, so

$$\sum_{x \in \{-1,1\}^d \setminus U} \exp(x^\top A x) \leq 2^d \exp(\beta(c_G - 1)).$$

It follows that

$$p^\star(\{-1, 1\}^d \setminus U) = \frac{\sum_{x \in \{-1,1\}^d \setminus U} \exp(x^\top A x)}{\sum_{x \in \{-1,1\}^d} \exp(x^\top A x)}$$

$$\begin{aligned}
&\leq \frac{2^d \exp(\beta(c_G - 1))}{\exp(\beta c_G) + 2^d \exp(\beta(c_G - 1))} \\
&= \frac{2^d}{e^\beta + 2^d} \\
&\leq 1/100
\end{aligned}$$

by choice of β. Let ν be the law of the sample $\widetilde{x}$. By assumption, we have $\mathsf{W}_2(\nu, p^\star) \leq 1/4$. Let $V := \{y \in \mathbb{R}^d : \mathbb{1}[y \geq 1/2] \in U\}$ where $\mathbb{1}[y \geq 1/2]$ refers to the coordinate-wise thresholding of y. By Lemma C.1 and the preceding bounds, we have $\nu(V) \geq p^\star(U) - (1/4)^2 \geq 0.9$. Thus, the rounded vector $\hat{x}$ satisfies $\beta^{-1}\hat{x}^\top A\hat{x} = c_G$ with probability at least 0.9. If the answer to the MAX-CUT instance is NO, then the output is always NO, since it holds almost surely that $\beta^{-1}\hat{x}^\top A\hat{x} = c_G(S) \leq c_G < k$ where $\hat{x} = \mathbb{1}[S]$. If the answer is YES, then with probability at least 0.9 we have $\beta^{-1}\hat{x}^\top A\hat{x} = c_G \geq k$ and hence the output is YES. Moreover, the time complexity of the reduction is $\mathrm{poly}(d)$ by assumption. Thus, NP $\subseteq$ BPP. ■

Appendix B. Steering with diagonal rewards is hard

The following result is a straightforward corollary of Theorem 4.1. It asserts that even if the reward is $\mathbf{r}(x) = x^\top Ax$ where A is negative-definite, rank-one, and diagonal (and hence has only a single nonzero entry), the steering problem is intractable.

Corollary B.1 *Suppose that there is a randomized algorithm $\mathcal{A}$ with the following property. For any integer $d \in \mathbb{N}$, any distribution $p \in \Delta(\mathbb{R}^d)$ satisfying Assumption 2.2 with parameter $C_{\mathsf{norm}} \geq 1$, and any rank-1, negative semi-definite, diagonal $A \in \mathbb{R}^{d\times d}$, the output $\widetilde{x} \in \mathbb{R}^d$ of $\mathcal{A}((\nabla \log p_\sigma)_{\sigma\in(0,1)}, A, C_{\mathsf{norm}})$ has law ν satisfying $\mathsf{W}_2(\nu, p^\star) \leq 1/4$, where $p^\star \in \Delta(\mathbb{R}^d)$ is the tilted distribution*

$$p^\star(x) \propto p(x)\exp(x^\top Ax).$$

Moreover, the time complexity of $\mathcal{A}((\nabla \log p_\sigma)_{\sigma\in(0,1)}, A, C_{\mathsf{norm}})$ is $\mathrm{poly}(d, C_{\mathsf{norm}})$.

Then, NP $\subseteq$ BPP.

Proof Any negative semi-definite A is diagonalizable, i.e. we can write $A = U^\top DU$ where U is orthogonal and D is diagonal. Let q be the law of Ux where $x \sim p$. Define $q^\star$ by $q^\star(x) \propto q(x)\exp(x^\top Dx)$. Observe that $p^\star$ is the law of $U^\top x$ where $x \sim q^\star$. Moreover, given query access to $\nabla \log p_\sigma$, we can efficiently simulate query access to $\nabla \log q_\sigma$, since $q_\sigma(x) = p_\sigma(U^\top x)$ and hence $\nabla \log q_\sigma(x) = U^\top \nabla \log p_\sigma(U^\top x)$.

Thus, if an algorithm $\mathcal{A}$ exists as posited in the corollary statement, then an algorithm $\mathcal{A}'$ exists as posited in the statement of Theorem 4.1, and hence NP $\subseteq$ BPP. ■

Appendix C. Omitted details from Section 4

The following lemma lower bounds the Wasserstein distance between a distribution on the hypercube μ and any other distribution ν in terms of the measure that ν puts on the "rounding" of the measure onto the hypercube:

Lemma C.1 *Let μ be any probability measure supported on $\{\pm 1\}^d$, let $S \subseteq \{\pm 1\}^d$, and let ν be any probability measure on $\mathbb{R}^d$ with finite second moment. Let $R_S := \{y \in \mathbb{R}^d : \mathrm{sgn}(y) \in S\}$. Then:*

$$W_2(\mu,\nu)^2 \;\geq\; \mu(S) - \nu(R_S).$$

Thus, if $W_2(\mu,\nu) \leq \varepsilon$, then

$$\nu(R_S) \;\geq\; \mu(S) - \varepsilon^2.$$

Proof Fix an arbitrary coupling (X,Y) of μ and ν, i.e. $X \sim \mu$ and $Y \sim \nu$. Define the event

$$E := \{X \in S \text{ and } Y \notin R_S\}.$$

On E, we have $X \in S \subseteq \{\pm 1\}^n$, while $Y \notin R_S$ means $\mathrm{sgn}(Y) \notin S$. In particular, since $X \in S$ but $\mathrm{sgn}(Y) \notin S$, we must have $\mathrm{sgn}(Y) \neq X$. Therefore, on the event E we have $\|X - Y\|_2^2 \geq 1$. Consequently,

$$\mathbb{E}\|X-Y\|_2^2 \;\geq\; \mathbb{E}\big[\|X-Y\|_2^2 \cdot \mathbf{1}_E\big] \;\geq\; \mathbb{E}[\mathbf{1}_E] \;=\; \mathbb{P}(E).$$

Next we lower bound $\mathbb{P}(E)$. Since $\mathbb{P}(E) = \mathbb{P}(X \in S) - \mathbb{P}(X \in S,\ Y \in R_S)$, and since $\mathbb{P}(X \in S) = \mu(S)$ and $\mathbb{P}(X \in S,\ Y \in R_S) \leq \mathbb{P}(Y \in R_S) = \nu(R_S)$, we get:

$$\mathbb{P}(E) \;\geq\; \mu(S) - \nu(R_S).$$

Combining the two inequalities, we get that for an arbitrary coupling (X,Y),

$$\mathbb{E}\|X-Y\|_2^2 \;\geq\; \mu(S) - \nu(R_S).$$

Taking the infimum over all couplings gives

$$W_2(\mu,\nu)^2 = \inf \mathbb{E}\|X-Y\|_2^2 \;\geq\; \mu(S) - \nu(R_S),$$

which is what we wanted. ■

C.1. Proof of Lemma 4.3

Lemma C.2 (Restatement of Lemma 4.3) *Given a PARTITION instance $w \in \mathbb{Z}^d$, define the set $S_w := \{x \in \{\pm 1\}^d : w^\top x = 0\}$. Assume $S_w \neq \emptyset$ (i.e. the PARTITION instance is a YES instance). Then $q_w(S_w) \;\geq\; \frac{200}{201}$, where $q_w \in \Delta(\mathbb{R}^d)$ is defined by $q_w(x) \propto \exp(-(d+5)\langle x, w\rangle^2)$.*

Proof

Let $\beta := d + 5$. Let us also denote

$$\widetilde{Z}_w := \sum_{x \in \{\pm 1\}^d} \exp\big(-\beta (w^\top x)^2\big),$$

so that

$$q_w(x) \;=\; \frac{\exp\big(-\beta(w^\top x)^2\big)}{\widetilde{Z}_w}.$$

For $x \in S_w$, we have $w^\top x = 0$, hence $\exp(-\beta(w^\top x)^2) = 1$. Thus

$$\sum_{x \in S_w} \exp\big(-\beta(w^\top x)^2\big) = |S_w|.$$

For $x \notin S_w$, we have $w^\top x \in \mathbb{Z} \setminus \{0\}$, so $|w^\top x| \geq 1$ and therefore

$$\exp\big(-\beta(w^\top x)^2\big) \;\leq\; e^{-\beta}.$$

Hence

$$\widetilde{Z}_w = \sum_{x \in S_w} 1 \;+\; \sum_{x \in \{\pm 1\}^d \setminus S_w} \exp\big(-\beta(w^\top x)^2\big) \;\leq\; |S_w| + (2^d - |S_w|)\, e^{-\beta} \;\leq\; |S_w| + 2^d e^{-\beta}.$$

Therefore

$$q_w(S_w) \;=\; \frac{|S_w|}{\widetilde{Z}_w} \;\geq\; \frac{|S_w|}{|S_w| + 2^d e^{-\beta}}.$$

We lower bound $|S_w|$ and upper bound $2^d e^{-\beta}$. Since $S_w \neq \emptyset$, pick some $x^\star \in S_w$. Then also $-x^\star \in S_w$ because $w^\top(-x^\star) = -w^\top x^\star = 0$. For $d \geq 2$, $x^\star \neq -x^\star$, hence $|S_w| \geq 2$. Next, with $\beta = d + 5$, we have $2^d e^{-\beta} = 2^d e^{-(d+5)} = e^{-5}\left(\frac{2}{e}\right)^d \leq e^{-5} < \frac{1}{100}$. Combining the bounds,

$$q_w(S_w) \;\geq\; \frac{|S_w|}{|S_w| + 2^d e^{-\beta}} \;\geq\; \frac{2}{2 + 1/100} \;=\; \frac{200}{201}$$

as claimed. ∎

Appendix D. Omitted details from Section 5

We recall notation from Section 5. Fix $d, r \in \mathbb{N}$. Suppose that $p \in \Delta(\mathbb{R}^d)$ satisfies Assumption 2.2 with parameter $C_{\mathsf{norm}} \geq 1$. Fix $L \in \mathbb{R}^{r \times d}$, and define $p^\star \in \Delta(\mathbb{R}^d)$ by

$$p^\star(x) \propto p(x) e^{\frac{1}{2}\|Lx\|_2^2}.$$

D.1. Analysis of discretization error

The Hubbard-Stratonovich transform (Lemma 5.4) decomposes $p^\star$ as a mixture of linear-tilted distributions $p(x; L^\top z)$ (Definition 3.1) where z ranges continuously over $\mathbb{R}^d$. The following lemma shows that the range of z can be discretized with small error:

Lemma D.1 *Fix $\epsilon \in (0, 1/100)$ and $\gamma, R > 0$. Let $D := \sup_{x \in \mathrm{supp}(p)} \|Lx\|_2$. Let $\mathcal{S} := \gamma\mathbb{Z}^r \cap \mathscr{B}_{r,2}(R)$. Define*

$$q(x) \propto \sum_{z \in \mathcal{S}} Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z).$$

If $R \geq D + 2\sqrt{r} + 2\sqrt{\log(1/\epsilon)}$ and $\gamma \leq \epsilon/D$, then

$$\mathsf{TV}(q, p^\star) \leq 20\epsilon.$$

Proof For any $z, z' \in \mathbb{R}^r$, we have

$$e^{-D\|z-z'\|_2} \le \frac{Z(z)}{Z(z')} \le e^{D\|z-z'\|_2}$$

and thus, for all $x \in \mathbb{R}^d$,

$$e^{-2D\|z-z'\|_2} \le \frac{p(x; L^\top z)}{p(x; L^\top z')} \le e^{2D\|z-z'\|_2}.$$

For each $z \in \mathcal{S}$ define $B(z) \subset \mathbb{R}^r$ by $B(z) := z + \mathscr{B}_{r,\infty}(\gamma/2)$. Then the sets $(B(z) : z \in \gamma\mathbb{Z}^r)$ partition $\mathbb{R}^d$, so by Lemma 5.4,

$$\begin{aligned} p^\star(x) &\propto \int_{\mathbb{R}^r} Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z)\,\mathrm{d}z \\ &= \sum_{z \in \gamma\mathbb{Z}^r} \int_{B(z)} Z(z') e^{-\frac{1}{2}\|z'\|_2^2} p(x; L^\top z')\,\mathrm{d}z'. \end{aligned}$$

For convenience, define

$$p_R^\star(x) \propto f(x) := \sum_{z \in \mathcal{S}} \int_{B(z)} Z(z') e^{-\frac{1}{2}\|z'\|_2^2} p(x; L^\top z')\,\mathrm{d}z'.$$

We observe that for any $x \in \mathrm{supp}(p)$,

$$\begin{aligned} 1 &\ge \frac{\sum_{z \in \mathcal{S}} \int_{B(z)} e^{-\frac{1}{2}\|z'\|_2^2 + \langle Lx, z'\rangle}\,\mathrm{d}z'}{\int_{\mathbb{R}^r} e^{-\frac{1}{2}\|z\|_2^2 + \langle Lx, z\rangle}\,\mathrm{d}z} \\ &\ge \frac{\int_{\mathscr{B}_{r,2}(R)} e^{-\frac{1}{2}\|z\|_2^2 + \langle Lx, z\rangle}\,\mathrm{d}z}{\int_{\mathbb{R}^r} e^{-\frac{1}{2}\|z\|_2^2 + \langle Lx, z\rangle}\,\mathrm{d}z} \\ &= \Pr_{z \sim \mathcal{N}(Lx, I_r)}[\|z\|_2 \le R] \\ &\ge 1 - \epsilon \end{aligned}$$

by Lemma D.2, the assumption that $R \ge D + 2\sqrt{r} + 2\sqrt{\log(1/\epsilon)}$, and the fact that $\|Lx\|_2 \le D$. It follows that

$$\begin{aligned} (1-\epsilon) \int_{\mathbb{R}^d} p(x) \int_{\mathbb{R}^r} e^{-\frac{1}{2}\|z'\|_2^2 + \langle Lx, z'\rangle}\,\mathrm{d}z'\,\mathrm{d}x &\le \int_{\mathbb{R}^d} p(x) \sum_{z \in \mathcal{S}} \int_{B(z)} e^{-\frac{1}{2}\|z'\|_2^2 + \langle Lx, z'\rangle}\,\mathrm{d}z'\,\mathrm{d}x \\ &\le \int_{\mathbb{R}^d} p(x) \int_{\mathbb{R}^r} e^{-\frac{1}{2}\|z'\|_2^2 + \langle Lx, z'\rangle}\,\mathrm{d}z'\,\mathrm{d}x \end{aligned}$$

and hence

$$p_R^\star(x) = \frac{p(x) \sum_{z \in \mathcal{S}} \int_{B(z)} e^{-\frac{1}{2}\|z'\|_2^2 + \langle Lx, z'\rangle}\,\mathrm{d}z'}{\int_{\mathbb{R}^d} p(x) \sum_{z \in \mathcal{S}} \int_{B(z)} e^{-\frac{1}{2}\|z'\|_2^2 + \langle Lx, z'\rangle}\,\mathrm{d}z'\,\mathrm{d}x} \in [(1-\epsilon)p^\star(x), (1-\epsilon)^{-1}p^\star(x)]$$

which means that

$$\mathsf{TV}(p^\star, p_R^\star) = \int_{\mathbb{R}^d} |p^\star(x) - p_R^\star(x)| \, \mathrm{d}x \le 2\epsilon \int_{\mathbb{R}^d} p^\star(x) \, \mathrm{d}x = 2\epsilon.$$

Next, we compare $p_R^\star$ with q. Write

$$g(x) := \sum_{z \in \mathcal{S}} Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z).$$

Observe that for any $z, z' \in \mathscr{B}_{r,2}(R)$ with $z' \in B(z)$, we have $\|z - z'\|_2 \le \gamma \le \epsilon/R$, and therefore $|Z(z) - Z(z')| \le 2\epsilon Z(z)$ and $|p(x; L^\top z) - p(x; L^\top z')| \le 4\epsilon p(x; L^\top z)$ and $|e^{-\frac{1}{2}\|z\|_2^2} - e^{-\frac{1}{2}\|z'\|_2^2}| \le 2\epsilon \cdot e^{-\frac{1}{2}\|z\|_2^2}$. Thus,

$$\begin{aligned}
\int_{\mathbb{R}^d} |f(x) - g(x)| \, \mathrm{d}x &\le \sum_{z \in \mathcal{S}} \int_{B(z)} \int_{\mathbb{R}^d} \left| Z(z') e^{-\frac{1}{2}\|z'\|_2^2} p(x; L^\top z') - Z(z) e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z) \right| \, \mathrm{d}x \, \mathrm{d}z' \\
&\le \sum_{z \in \mathcal{S}} \int_{B(z)} \int_{\mathbb{R}^d} \left| Z(z') - Z(z) \right| e^{-\frac{1}{2}\|z\|_2^2} p(x; L^\top z) \, \mathrm{d}x \, \mathrm{d}z' \\
&+ \sum_{z \in \mathcal{S}} \int_{B(z)} \int_{\mathbb{R}^d} Z(z') \left| e^{-\frac{1}{2}\|z'\|_2^2} - e^{-\frac{1}{2}\|z\|_2^2} \right| p(x; L^\top z) \, \mathrm{d}x \, \mathrm{d}z' \\
&+ \sum_{z \in \mathcal{S}} \int_{B(z)} \int_{\mathbb{R}^d} Z(z') e^{-\frac{1}{2}\|z'\|_2^2} \left| p(x; L^\top z) - p(x; L^\top z') \right| \, \mathrm{d}x \, \mathrm{d}z' \\
&\le 9\epsilon \cdot \sum_{z \in S} \int_{B(z)} \int_{\mathbb{R}^d} Z(z') e^{-\frac{1}{2}\|z'\|_2^2} p(x; L^\top z') \, \mathrm{d}x \, \mathrm{d}z' \\
&= 9\epsilon \int_{\mathbb{R}^d} f(x) \, \mathrm{d}x
\end{aligned}$$

where the last inequality holds by using the preceding observations and the fact that $\epsilon < 1/100$. It follows from Lemma D.3 that $\mathsf{TV}(p_R^\star, q) \le 18\epsilon$. Combining the above bounds gives $\mathsf{TV}(p^\star, q) \le 20\epsilon$ as claimed. ■

D.2. Technical lemmas

Lemma D.2 (Concentration of χ^2-random variable) *Fix $d \in \mathbb{N}$ and let $Z \sim \mathcal{N}(0, I_d)$. Then for any $\epsilon > 0$,*

$$\Pr[\|Z\|_2 > 2\sqrt{d} + 2\sqrt{\log(1/\epsilon)}] \le \epsilon.$$

Lemma D.3 *Let $f, g : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ be integrable, and let $p, q \in \Delta(\mathbb{R}^d)$ be defined by $p(x) \propto f(x)$ and $q(x) \propto g(x)$. Then*

$$\mathsf{TV}(p, q) \le \frac{2 \int |f(x) - g(x)| \, \mathrm{d}x}{\int f(x) \, \mathrm{d}x}.$$

Proof Set $Z_f := \int f(x) \, \mathrm{d}x$ and $Z_g := \int g(x) \, \mathrm{d}x$. Then we have

$$\mathsf{TV}(p, q) = \int \left| \frac{f(x)}{Z_f} - \frac{g(x)}{Z_g} \right| \mathrm{d}x$$

$$
\begin{aligned}
&\leq \int \left| \frac{f(x)}{Z_f} - \frac{g(x)}{Z_f} \right| \mathrm{d}x + \int \left| \frac{g(x)}{Z_f} - \frac{g(x)}{Z_g} \right| \mathrm{d}x \\
&\leq \frac{1}{Z_f} \int |f(x) - g(x)| \, \mathrm{d}x + \left| \frac{1}{Z_f} - \frac{1}{Z_g} \right| \int g(x) \, \mathrm{d}x \\
&\leq \frac{1}{Z_f} \int |f(x) - g(x)| \, \mathrm{d}x + \frac{|Z_f - Z_g|}{Z_f} \\
&\leq \frac{1}{Z_f} \int |f(x) - g(x)| \, \mathrm{d}x + \frac{\int |f(x) - g(x)| \, \mathrm{d}x}{Z_f} \\
&= \frac{2}{Z_f} \int |f(x) - g(x)| \, \mathrm{d}x
\end{aligned}
$$

as claimed. ■