

Mirror Diffusion Models for Constrained and Watermarked Generation

Guan-Horng Liu, Tianrong Chen, Evangelos A. Theodorou[†], Molei Tao[†]
 Georgia Institute of Technology, USA
 {ghliu, tianrong.chen, evangelos.theodorou, mtao}@gatech.edu

Abstract

Modern successes of diffusion models in learning complex, high-dimensional data distributions are attributed, in part, to their capability to construct diffusion processes with analytic transition kernels and score functions. The tractability results in a simulation-free framework with stable regression losses, from which reversed, generative processes can be learned at scale. However, when data is confined to a constrained set as opposed to a standard Euclidean space, these desirable characteristics appear to be lost based on prior attempts. In this work, we propose **Mirror Diffusion Models (MDM)**, a new class of diffusion models that generate data on convex constrained sets without losing any tractability. This is achieved by learning diffusion processes in a dual space constructed from a mirror map, which, crucially, is a standard Euclidean space. We derive efficient computation of mirror maps for popular constrained sets, such as simplices and ℓ_2 -balls, showing significantly improved performance of MDM over existing methods. For safety and privacy purposes, we also explore constrained sets as a new mechanism to embed invisible but quantitative information (*i.e.*, watermarks) in generated data, for which MDM serves as a compelling approach. Our work brings new algorithmic opportunities for learning tractable diffusion on complex domains.

1 Introduction

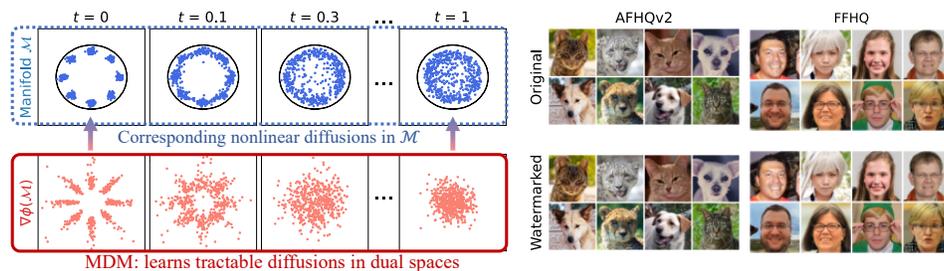


Figure 1: **Mirror Diffusion Models (MDM)** is a new class of diffusion models for convex constrained manifolds $\mathcal{M} \subseteq \mathbb{R}^d$. **(left)** Instead of learning score-approximate diffusions on \mathcal{M} , MDM applies a *mirror map* $\nabla\phi$ and learns *tractable* diffusions in its *unconstrained dual-space* $\nabla\phi(\mathcal{M}) = \mathbb{R}^d$. **(right)** We also present MDM for watermarked generation, where generated contents (*e.g.*, images) live in a high-dimensional *token* constrained set \mathcal{M} that is certifiable only from the private user.

Diffusion models [1–3] have emerged as powerful generative models with their remarkable successes in synthesizing high-fidelity data such as images [4, 5], audio [6, 7], and 3D geometry [8, 9]. These models work by progressively diffusing data to noise, and learning the score functions (often

[†]Equal advising.

Table 1: Comparison of different diffusion models for constrained generation on $\mathcal{M} \subseteq \mathbb{R}^d$. Rather than learning reflected diffusions on \mathcal{M} with approximate scores, our Mirror Diffusion constructs a mirror map $\nabla\phi$ and lift the diffusion processes to the *unconstrained dual space* $\nabla\phi(\mathcal{M}) = \mathbb{R}^d$, inheriting favorable features from standard Euclidean-space diffusion models. Constraints are satisfied by construction via the inverse map $\nabla\phi^*$; that is, $\nabla\phi^*(y) \in \mathcal{M}$ for all $y \in \nabla\phi(\mathcal{M})$.

Diffusion models	Domain	Tractable conditional score	Simulation-free training	Regression objective
<i>Reflected Diffusion</i>				
Fishman et al. [18]	\mathcal{M}	✗	✗	✗
Lou and Ermon [25]	\mathcal{M}	✗	\triangle^2	✓
<i>Mirror Diffusion</i>				
This work (ours)	$\nabla\phi(\mathcal{M})$	✓	✓	✓

parameterized by neural networks) to reverse the processes [10]; the reversed processes thereby provide transport maps that generate data from noise. Modern diffusion models [11–13] often employ diffusion processes whose transition kernels are analytically available. This characteristic enables *simulation-free* training, bypassing the necessity to simulate the underlying diffusion processes by directly sampling the diffused data. It also leads to tractable conditional score functions and simple regression objectives, facilitating computational scalability for high-dimensional problems. In standard Euclidean spaces, tractability can be accomplished by designing the diffusion processes as linear stochastic differential equations, but doing so for non-Euclidean spaces is nontrivial.

Recent progress of diffusion models has expanded their application to non-Euclidean spaces, such as Riemannian manifolds [14, 15], where data live in a curved geometry. The development of generative models on manifolds has enabled various new applications, such as modeling earth and climate events [16] and learning densities on meshes [17]. In this work, we focus on generative modeling on *constrained sets* (also called *constrained manifolds* [18]) that are defined by a set of inequality constraints. Such constrained sets, denoted by $\mathcal{M} \subseteq \mathbb{R}^d$, are also ubiquitous in several scientific fields such as computational statistics [19], biology [20], and robotics [21, 22].

Previous endeavors in the direction of constrained generation have primarily culminated in reflected diffusions [23, 24], which reflect the direction at the boundary $\partial\mathcal{M}$ to ensure that samples remain inside the constraints. Unfortunately, reflected diffusions do not possess closed-form transition kernels [18, 25], thus necessitating the approximation of the conditional score functions that can hinder learning performance. Although simulation-free training is still possible², it comes with a computational overhead due to the use of geometric techniques [25]. From a theoretical standpoint, reflected Langevin dynamics are widely regarded for their extended mixing time [26, 27]. This raises practical concerns when simulating reflected diffusions, as prior methods often adopted special treatments such as thresholding and discretization [18, 25].

An alternative diffusion for constrained sampling is the Mirror Langevin Dynamics (MLD [28]). MLD, as a subclass of Riemannian Langevin Dynamics [29, 30], is tailored specifically to convex constrained sampling. Specifically, MLD constructs a strictly convex function ϕ so that its gradient map, often referred to as the *mirror map* $\nabla\phi : \mathcal{M} \rightarrow \mathbb{R}^d$, defines a nonlinear mapping from the initial constrained set to an *unconstrained dual space*. Despite extensive investigation of MLD [31–34], its potential as a foundation for designing diffusion generative models remains to be comprehensively examined. Thus, there is a need to explore the potential of MLD for designing new, effective, and tailored diffusion models for constrained generation.

With this in mind, we propose **Mirror Diffusion Models (MDM)**, a new class of diffusion generative models for convex constrained sets. Similar to MLD, MDM utilizes mirror maps $\nabla\phi$ that transform data distributions on \mathcal{M} to its dual space $\nabla\phi(\mathcal{M})$. From there, MDM learns a *dual-space diffusion* between the mirrored distribution and Gaussian prior in the dual space. Note that this is in contrast to MLD, which constructs invariant distributions on the initial constrained set; MDM instead considers dual-space priors. Since the dual space is also unconstrained Euclidean space $\nabla\phi(\mathcal{M}) = \mathbb{R}^d$, MDM

²Lou and Ermon [25] proposed a semi-simulation-free training with the aid of geometric techniques.

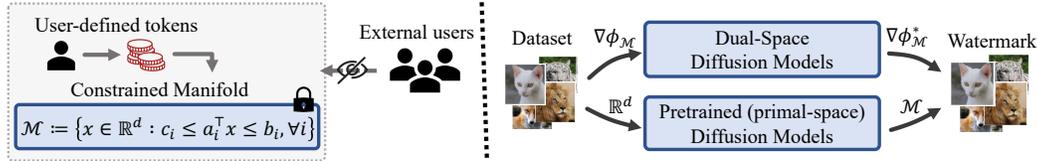


Figure 2: MDM for watermarked generation: **(left)** We first construct a constrained set \mathcal{M} based on a set of user-defined tokens private to other users. **(right)** MDM can be instantiated by either learning the corresponding dual-space diffusions, or projecting pretrained, *i.e.*, unwatermarked, diffusion models onto \mathcal{M} . In both cases, MDM embeds watermarks that are certifiable only from the user.

preserves many important features of standard diffusion models, such as simulation-free training and tractable conditional scores, which yields simple regression objectives; see Table 1. After learning the dual-space distribution, MDM applies the inverse mapping $\nabla\phi^*$, transforming mirrored samples back to the constrained set. We propose efficient construction of mirror maps and demonstrate that MDM outperforms prior reflected diffusion models on common constrained sets, such as ℓ_2 -balls and simplices.

Moreover, we show that MDM also stands as a novel approach for *watermarked generation*, a technique that aims to embed undetectable information (*i.e.*, watermarks) into generated contents to prevent unethical usage and ensure copyright protection. This newly emerging goal nevertheless attracted significant attention in natural language generation [35–37], while what we will present is a general algorithmic strategy demonstrated for image generations.

To generate watermarked contents that are private to individual users, we begin by constructing a constrained set \mathcal{M} based on user-defined tokens (see Figure 2). MDM can then be instantiated to learn the dual-space distribution. Alternatively, MDM can distill pretrained diffusion models [38] by projecting the generated data onto the constraint set. In either case, MDM generates watermarked contents that can be certified only by users who know the tokens. Our empirical results suggest that MDM serves as a competitive approach to prior watermarking techniques for diffusion models, such as those proposed by Zhao et al. [39], by achieving lower (hence better) FID values.

In summary, we present the following contributions:

- We introduce **Mirror Diffusion Models (MDM)**, a new class of diffusion models that utilizes mirror maps to enable constrained generation with Euclidean-space diffusion models.
- We propose efficient computation of mirror maps on common constrained sets, including balls and simplices. Our results demonstrate that MDM consistently outperforms previous reflected diffusion models.
- As a novel application of constrained generation, we explore MDM’s potential for watermarking and copyrighting diffusion models where constraints serve as private tokens. We show that it yields competitive performance compared to prior methods.

2 Preliminary on Euclidean-space Diffusion Models

Notation $\mathcal{M} \subseteq \mathbb{R}^d$ denotes a convex constrained set. We preserve $x \in \mathcal{M}$ and $y \in \mathbb{R}^d$ to better distinguish variables on constrained sets and standard d -dimensional Euclidean space. $\mathbf{I} \in \mathbb{R}^{d \times d}$ denotes the identity matrix. x_i denotes the i -th element of a vector x . Similarly, $[A]_{ij}$ denotes the element of a matrix A at i -th row and j -th column.

Diffusion process Given $y_0 \in \mathbb{R}^d$ sampled from some Euclidean-space distribution, conventional diffusion models define the diffusion process as a *forward* Markov chain with the joint distribution:

$$q(y_{1:T}|y_0) = \prod_{t=1}^T q(y_t|y_{t-1}), \quad q(y_t|y_{t-1}) := \mathcal{N}(y_t; \sqrt{1 - \beta_t}y_{t-1}, \beta_t\mathbf{I}), \quad (1)$$

which progressively injects Gaussian noise to y_0 such that, for a sufficiently large T and a properly chosen noise schedule $\beta_t \in (0, 1)$, y_T approaches an approximate Gaussian, *i.e.*, $q(y_T) \approx \mathcal{N}(0, \mathbf{I})$.

Equation (1) has tractable marginal densities, and two of which that are of particular interest are:

$$q(y_t|y_0) = \mathcal{N}(y_t; \sqrt{\bar{\alpha}_t}y_0, (1 - \alpha_t)\mathbf{I}), \quad q(y_{t-1}|y_t, y_0) = \mathcal{N}(y_{t-1}; \tilde{\mu}_t(y_t, y_0), \tilde{\beta}_t\mathbf{I}), \quad (2)$$

$$\tilde{\mu}_t(y_t, y_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}y_0 + \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}y_t, \quad \tilde{\beta}_t := \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t, \quad \bar{\alpha}_t := \prod_{s=0}^t(1 - \beta_s).$$

The tractable marginal $q(y_t|y_0)$ enables direct sampling of y_t without simulating the forward Markov chain (1), *i.e.*, $y_t|y_0$ can be sampled in a *simulation-free* manner. It also suggests a closed-form score function $\nabla \log q(y_t|y_0)$. The marginal $q(y_{t-1}|y_t, y_0)$ hints the optimal reverse process given y_0 .

Generative process The generative process, which aims to *reverse* the forward diffusion process (1), is defined by a *backward* Markov chain with the joint distribution:

$$p_\theta(y_{0:T}) = p_T(y_T) \prod_{t=1}^T p_\theta(y_{t-1}|y_t), \quad p_\theta(y_{t-1}|y_t) := \mathcal{N}(y_{t-1}; \mu_\theta(y_t, t), \tilde{\beta}_t\mathbf{I}). \quad (3)$$

Here, the mean μ_θ is typically parameterized by some DNNs, *e.g.*, U-net [40], with θ . Equations (1) and (3) imply an evidence lower bound (ELBO) for maximum likelihood training, indicating that the optimal prediction of $\mu_\theta(y_t, t)$ should match $\tilde{\mu}_t(y_t, y_0)$ in Equation (2).

Parameterization and training There exists many different ways to parameterize $\mu_\theta(y_t, t)$, and each of them leads to a distinct training objective. For instance, Ho et al. [2] found that

$$\mu_\theta(y_t, t) := \frac{1}{\sqrt{1 - \beta_t}} \left(y_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(y_t, t) \right), \quad (4)$$

achieves better empirical results compared to directly predicting y_0 or $\tilde{\mu}_t$. This parameterization aims to predict the “noise” injected to $y_t \sim q(y_t|y_0)$, as the ELBO reduces to a simple regression:

$$\mathcal{L}(\theta) := \mathbb{E}_{t, y_0, \epsilon} [\lambda(t) \|\epsilon - \epsilon_\theta(y_t, t)\|]. \quad (5)$$

where $y_t = \sqrt{\bar{\alpha}_t}y_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, and t sampled uniformly from $\{1, \dots, T\}$. The weighting $\lambda(t)$ is often set to 1. Once ϵ_θ is properly trained, we can generate samples in Euclidean space by simulating Equation (3) backward from $y_T \sim \mathcal{N}(0, \mathbf{I})$.

3 Mirror Diffusion Models (MDM)

In this section, we present a novel class of diffusion models, Mirror Diffusion Models, which are designed to model probability distributions $p_{\text{data}}(x)$ supported on a convex constrained set $\mathcal{M} \subseteq \mathbb{R}^d$ while retaining the computational advantages of Euclidean-space diffusion models.

3.1 Dual-Space Diffusion

Mirror map Following the terminology in Li et al. [34], let $\phi : \mathcal{M} \rightarrow \mathbb{R}$ be a twice differentiable³ function that is strictly convex and satisfying that $\lim_{x \rightarrow \partial\mathcal{M}} \|\nabla\phi(x)\| \rightarrow \infty$ and $\nabla\phi(\mathcal{M}) = \mathbb{R}^d$. We call its gradient map $\nabla\phi : \mathcal{M} \rightarrow \mathbb{R}^d$ the *mirror map* and its image, $\nabla\phi(\mathcal{M}) = \mathbb{R}^d$, as its *dual/mirror space*. Let $\phi^* : \mathbb{R}^d \rightarrow \mathbb{R}$ be the dual function of ϕ defined by

$$\phi^*(y) = \sup_{x \in \mathcal{M}} \langle x, y \rangle - \phi(x) \quad (6)$$

A notable property of this dual function $\phi^*(y)$ is that its gradient map *reverses* the mirror map. In other words, we have $\nabla\phi^* = (\nabla\phi)^{-1}$, implying that

$$\nabla\phi^*(\nabla\phi(x)) = x \text{ for all } x \in \mathcal{M}, \quad \nabla\phi(\nabla\phi^*(y)) = y \text{ for all } y \in \nabla\phi(\mathcal{M}) = \mathbb{R}^d.$$

The functions $(\nabla\phi, \nabla\phi^*)$ define a nonlinear, yet bijective, mapping between the convex constrained set \mathcal{M} and Euclidean space \mathbb{R}^d . As a result, MDM does not require building diffusion models on \mathcal{M} , as done in previous approaches, but rather they learn a standard Euclidean-space diffusion model in the corresponding dual space $\nabla\phi(\mathcal{M}) = \mathbb{R}^d$.

³We note that standard mirror maps require twice differentiability to induce a (Riemannian) metric on the mirror space for constructing MLD. For our MDM, however, twice differentiability is needed only for Equation (8), and continuous differentiability, *i.e.*, C^1 , would suffice for training.

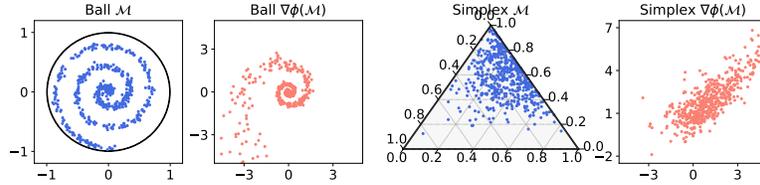


Figure 3: Examples of how mirror maps $\nabla\phi$ pushforward constrained distributions to unconstrained ones, for **(left)** an ℓ_2 -ball where $\mathcal{M} := \{x \in \mathbb{R}^2 : \|x\|_2^2 < R\}$ and **(right)** a simplex Δ_3 where $\mathcal{M} := \{x \in \mathbb{R}^2 : \sum_{i=1}^2 x_i \leq 1, x_i \geq 0\}$. Note that for simplex, $x_3 = 1 - x_1 - x_2$ is a redundant coordinate; see Section 3.2 for more details.

Essentially, the goal of MDM is to model the *dual-space distribution*:

$$\tilde{p}_{\text{data}}(y) := ([\nabla\phi_{\#}]p_{\text{data}})(x), \quad (7)$$

where $\#$ is the push-forward operation and $y = \nabla\phi(x)$ is the mirror of data $x \in \mathcal{M}$.

Training and generation MDM follows the same training procedure of Euclidean-space diffusion models, except with an additional push-forward given by Equation (7). Similarly, its generation procedure includes an additional step that maps the dual-space samples y_0 , generated from Equation (3), back to the constrained set via $\nabla\phi^*(y_0) \in \mathcal{M}$.

Tractable variational bound Given a datapoint $x \in \mathcal{M}$, its mirror $y_0 = \nabla\phi(x) \in \mathbb{R}^d$, and the dual-space Markov chain $\{y_1, \dots, y_T\}$, the ELBO of MDM can be computed by

$$L_{\text{ELBO}}(x) := \log |\det \nabla^2 \phi^*(y_0)| + \tilde{L}_{\text{ELBO}}(y_0), \quad (8)$$

where we apply the change of variables theorem [41] w.r.t. the mapping $x = \nabla\phi^*(y_0)$. The dual-space ELBO $\tilde{L}_{\text{ELBO}}(y_0)$ shares the same formula with the Euclidean-space ELBO [42, 43],

$$\tilde{L}_{\text{ELBO}}(y_0) := D_{\text{KL}}(q(y_T|y_0) \parallel p(y_T)) + \sum_{t=1}^{T-1} D_{\text{KL}}(q(y_t|y_{t+1}, y_0) \parallel p_{\theta}(y_t|y_{t+1})) - \log p_{\theta}(y_0|y_1). \quad (9)$$

It should be noted that (8,9) provide a *tractable* variational bound for constrained generation. This stands in contrast to concurrent methods relying on reflected diffusions, whose ELBO entails either intractable [18] or approximate [25] scores that could be restricted when dealing with more complex constrained sets. While, in practice, MDM is trained using the regression objective in Equation (5), computing tractable ELBO might be valuable for independent purposes such as likelihood estimation.

3.2 Efficient Computation of Mirror Maps

What remains to be answered pertains to the construction of mirror maps for common constraint sets such as ℓ_2 -balls, simplices, polytopes, and their products. In particular, we seek efficient, preferably closed-form, computation for $\nabla\phi$, $\nabla\phi^*$ and $\nabla^2\phi^*$.

ℓ_2 -Ball Let $\mathcal{M} := \{x \in \mathbb{R}^d : \|x\|_2^2 < R\}$ denote the ℓ_2 -ball of radius R in \mathbb{R}^d . We consider the common log-barrier function:

$$\phi_{\text{ball}}(x) := -\gamma \log(R - \|x\|_2^2). \quad (10)$$

where $\gamma \in \mathbb{R}^+$. The mirror map, its inverse, and the Hessian can be computed analytically by

$$\begin{aligned} \nabla\phi_{\text{ball}}(x) &= \frac{2\gamma x}{R - \|x\|_2^2}, & \nabla\phi_{\text{ball}}^*(y) &= \frac{Ry}{\sqrt{R\|y\|_2^2 + \gamma^2 + \gamma}}, \\ \nabla^2\phi_{\text{ball}}^*(y) &= \frac{R}{R\sqrt{\|y\|_2^2 + \gamma^2 + \gamma}} \left(\mathbf{I} - \frac{R}{\left(R\sqrt{\|y\|_2^2 + \gamma^2 + \gamma}\right) \sqrt{R\|y\|_2^2 + \gamma^2 + \gamma}} yy^\top \right). \end{aligned} \quad (11)$$

We refer to Appendix A for detailed derivation. Figure 3 visualizes such a mirror map for an ℓ_2 -ball in 2D. It is worth noting that while the mirror map of log-barriers generally does not admit an analytical inverse, this is not the case for the ℓ_2 -ball constraints.

Simplex Given a $(d+1)$ -dimensional simplex, $\Delta_{d+1} := \{x \in \mathbb{R}^{d+1} : \sum_{i=1}^{d+1} x_i = 1, x_i \geq 0\}$, we follow standard practices [18, 25] by constructing the constrained set $\mathcal{M} := \{x \in \mathbb{R}^d : \sum_{i=1}^d x_i \leq 1, x_i \geq 0\}$ and define $x_{d+1} := 1 - \sum_{i=1}^d x_i$.⁴

While conventional log-barriers may remain a viable option, a preferred alternative that is widely embraced in the MLD literature for enforcing simplices [28, 44] is the entropic function [45, 46]:

$$\phi_{\text{simplex}}(x) := \sum_{i=1}^d x_i \log(x_i) + \left(1 - \sum_{i=1}^d x_i\right) \log\left(1 - \sum_{i=1}^d x_i\right). \quad (12)$$

The mirror map, its inverse, and the hessian of the dual function can be computed analytically by

$$\begin{aligned} [\nabla \phi_{\text{simplex}}(x)]_i &= \log x_i - \log\left(1 - \sum_{i=1}^d x_i\right), & [\nabla \phi_{\text{simplex}}^*(y)]_i &= \frac{e^{y_i}}{1 + \sum_{i=1}^d e^{y_i}}, \\ [\nabla^2 \phi_{\text{simplex}}^*(y)]_{ij} &= \frac{e^{y_i}}{1 + \sum_{i=1}^d e^{y_i}} \mathbb{I}(i=j) - \frac{e^{y_i} e^{y_j}}{\left(1 + \sum_{i=1}^d e^{y_i}\right)^2}, \end{aligned} \quad (13)$$

where $\mathbb{I}(\cdot)$ is an indicator function. Again, we leave the derivation of Equation (13) to Appendix A. Figure 3 visualizes the entropic mirror map for a 3-dimensional simplex Δ_3 .

Polytope Let the constrained set be $\mathcal{M} := \{x \in \mathbb{R}^d : c_i < a_i^\top x < b_i, \forall i \in \{1, \dots, m\}\}$, where $c_i, b_i \in \mathbb{R}$ and $a_i \in \mathbb{R}^d$ are linearly independent to each other. We consider the standard log-barrier:

$$\phi_{\text{polytope}}(x) := - \sum_{i=1}^m (\log(\langle a_i, x \rangle - c_i) + \log(b_i - \langle a_i, x \rangle)) + \sum_{j=m+1}^d \frac{1}{2} \langle a_j, x \rangle^2, \quad (14)$$

$$\nabla \phi_{\text{polytope}}(x) = \sum_{i=1}^m s_i(\langle a_i, x \rangle) a_i + \sum_{j=m+1}^d \langle a_j, x \rangle a_j, \quad (15)$$

where the monotonic function $s_i : (c_i, b_i) \rightarrow \mathbb{R}$ is given by $s_i = \frac{-1}{\langle a_i, x \rangle - c_i} + \frac{-1}{\langle a_i, x \rangle - b_i}$, and $\{a_j\}$ do not impose any constraints and can be chosen arbitrarily as long as they span \mathbb{R}^d with $\{a_i\}$ (see below for more details); hence $\nabla \phi_{\text{polytope}}(\mathcal{M}) = \mathbb{R}^d$. While its inverse, $\nabla \phi_{\text{polytope}}^*$, does not admit closed-form in general, it does when all d constraints are *orthonormal*:

$$\nabla \phi_{\text{polytope}}^*(y) = \sum_{i=1}^m s_i^{-1}(\langle a_i, y \rangle) a_i + \sum_{j=m+1}^d \langle a_j, y \rangle a_j. \quad (16)$$

Here, $s_i^{-1} : \mathbb{R} \rightarrow (c_i, b_i)$ is the inverse of s_i . Essentially, (15,16) manipulate the coefficients of the bases from which the polytope set is constructed. These manipulations are nonlinear yet bijective, defined uniquely by s_i and s_i^{-1} , for each orthonormal basis a_i or, equivalently, for each constraint.

Naively implementing the mirror map via (15,16) can be problematic due to (i) numerical instability of s_i at boundaries (see Figure 4), (ii) the lack of closed-form solution for its inverse s_i^{-1} , and (iii) an *one-time* computation of orthonormal bases, which scales as $\mathcal{O}(d^3)$ for orthonormalization methods such as Gram-Schmidt, Householder or Givens [e.g., 47]. Below, we devise a modification for efficient and numerically stable computation, which will be later adopted for watermarked generation.

Improved computation for Polytope (15,16) Given the fact that only the first m orthonormal bases $\{a_1, \dots, a_m\}$ matter in the computation of (15,16), as the remaining $(d-m)$ orthonormal bases merely involve orthonormal projections, we can simplify the computational (15,16) to

$$\nabla \phi_{\text{poly}}(x) = x + \sum_{i=1}^m (s_i(\langle a_i, x \rangle) - \langle a_i, x \rangle) a_i, \quad \nabla \phi_{\text{poly}}^*(y) = y + \sum_{i=1}^m (s_i^{-1}(\langle a_i, y \rangle) - \langle a_i, y \rangle) a_i, \quad (17)$$

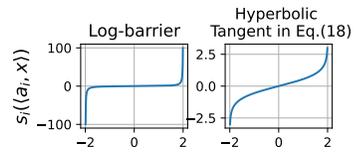


Figure 4: Comparison between s_i induced by standard log-barriers vs. hyperbolic tangents in Eq. (18).

⁴The transformation allows x to be bounded in a constrained set $\mathcal{M} \subseteq \mathbb{R}^d$ rather than a hyperplane in \mathbb{R}^{d+1} .

which, intuitively, add and subtract the coefficients of the first m orthonormal bases while leaving the rest intact. This reduces the complexity of computing orthonormal bases to $\mathcal{O}(md^2) \approx \mathcal{O}(d^2)$ and improves the numerical accuracy of inner-product multiplication, which can be essential for high-dimensional applications when d is large.

To address the instability of s_i and intractability of s_i^{-1} for log-barriers, we re-interpret the mirror maps (15,16) as the changes of coefficient bases. As this implies that any nonlinear bijective mapping with a tractable inverse would suffice, we propose a rescaling of the hyperbolic tangent:

$$s_i(\langle a_i, x \rangle) := (\tanh^{-1} \circ \text{rescale}_i)(\langle a_i, x \rangle), \quad s_i^{-1}(\langle a_i, y \rangle) := (\text{rescale}_i^{-1} \circ \tanh)(\langle a_i, y \rangle), \quad (18)$$

where $\text{rescale}_i(z) := 2 \frac{z-c_i}{(b_i-c_i)} - 1$ rescales the range from (c_i, b_i) to $(-1, 1)$. It is clear from Figure 4 that, compared to the s_i induced by the log-barrier, the mapping (18) is numerically stable at the boundaries and admits tractable inverse.

Complexity Table 2 summarizes the complexity of the mirror maps for each constrained set. We note that all computations are parallelizable, inducing nearly no computational overhead.

Table 2: Complexity of $\nabla\phi$ and $\nabla\phi^*$ for each constrained set.

ℓ_2 -Ball	Simplex	Polytope
$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(md)$

Remark 1. Equations (17) and (18) can be generalized to non-orthonormal constraints by adopting $\langle \tilde{a}_i, x \rangle$ and $\langle \tilde{a}_i, y \rangle$, where \tilde{a}_i is the i -th row of the matrix $\tilde{\mathbf{A}} := (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ and $\mathbf{A} := [a_1, \dots, a_m]$. Indeed, when \mathbf{A} is orthonormal, we recover $\tilde{a}_i = a_i$. We leave more discussions to Appendix B.

4 Related Work

Constrained sampling Sampling from a probability distribution on a constrained set has been a long-standing problem not only due to its extensive applications in, *e.g.*, topic modeling [48, 49] and Bayesian inference [50, 51], but its unique challenges in non-asymptotic analysis and algorithmic design [28, 32–34, 52, 53]. Mirror Langevin is a special instance of endowing Langevin dynamics with a Riemannian metric [30, 54], and it chooses a specific reshaped geometry so that dynamics have to travel infinite distance in order to cross the boundary of the constrained set. It is a sampling generalization of the celebrated Mirror Descent Algorithm [45] for convex constrained optimization, and the convexity of constrained set leads to the existence of a mirror map. However, Mirror Langevin is designed to draw samples from an unnormalized density supported on a constrained set. This stands in contrast to our MDM, which instead tackles problems of constrained *generation*, where only samples, not unnormalized density, are available, sharing much similarity to conventional generative modeling [3, 55]. In addition, Mirror Langevin can be understood to go back and forth between primal (constrained) and mirror (unconstrained) spaces, while MDM only does one such round trip.

Diffusion Models in non-Euclidean spaces There has been growing interest in developing diffusion generative models that can operate on domains that are not limited to standard Euclidean spaces, *e.g.*, discrete domains [56, 57] and equality constraints [58]. Seminar works by De Bortoli et al. [14] and Huang et al. [15] have introduced diffusion models on Riemannian manifolds, which have shown promising results in, *e.g.*, biochemistry [59, 60] and rotational groups [61, 62]. Regarding diffusion models on constrained manifolds, most concurrent works consider similar convex constrained sets such as simplices and polytopes [18, 25, 63]. Our MDM works on the same classes of constraints and includes additionally ℓ_2 -ball, which is prevalent in social science domains such as opinion dynamics [64, 65]. It is also worth mentioning that our MDM may be conceptually similar to Simplex Diffusion (SD) [63] which reconstructs samples on simplices similar to the mirror mapping. However, SD is designed specifically for simplices and Dirichlet distributions, while MDM is applicable to a broader class of convex constrained sets. Additionally, SD adopts Cox-Ingersoll-Ross processes [66], yielding a framework that, unlike MDM, is not simulation-free.

Latent Diffusion Models (LDMs) A key component to our MDM is the construction of mirror map, which allows us to define diffusion models in a different space than the original constrained set, thereby alleviating many computational difficulties. This makes MDM a particular type of LDMs [5, 67, 68], which generally consider structurally advantageous spaces for learning diffusion models. While LDMs typically *learn* such latent spaces either using pretrained model or on the fly using, *e.g.*, variational autoencoders, MDM instead derives analytically a latent (dual) space through a mirror map tailored specifically to the constrained set. Similar to LDM [69], the diffusion processes of MDM on the initial constrained set are implicitly defined and can be highly nonlinear.

Table 3: Results of ℓ_2 -ball constrained sets on five synthetic datasets of various dimension d . We report Sliced Wasserstein [70] w.r.t. 1000 samples, averaged over three trials, and include constraint violations for unconstrained diffusion models. Note that Reflected Diffusion [18] and MDM satisfy constraints by design. Our MDM clearly achieves similar or better performance to standard diffusion models while fully respecting the constraints. Other metrics, e.g., \mathcal{W}_1 , are reported in Appendix C. sc

	$d = 2$	$d = 2$	$d = 6$	$d = 8$	$d=20$
Sliced Wasserstein ↓					
DDPM [2]	0.0704 ± 0.0095	0.0236 ± 0.0048	0.0379 ± 0.0015	0.0231 ± 0.0020	0.0200 ± 0.0034
Reflected [18]	0.0642 ± 0.0181	0.0491 ± 0.0103	0.0609 ± 0.0055	0.0881 ± 0.0010	0.0574 ± 0.0065
MDM (ours)	0.0544 ± 0.0070	0.0214 ± 0.0025	0.0467 ± 0.0096	0.0292 ± 0.0017	0.0159 ± 0.0044
Constraint violation (%) ↓					
DDPM [2]	0.00 ± 0.00	0.00 ± 0.00	8.67 ± 0.87	13.60 ± 0.62	19.33 ± 1.29

Table 4: Results of **simplices constrained sets** on five synthetic datasets of various dimension d .

	$d = 3$	$d = 3$	$d = 7$	$d = 9$	$d=20$
Sliced Wasserstein ↓					
DDPM [2]	0.0089 ± 0.0002	0.0110 ± 0.0032	0.0047 ± 0.0004	0.0053 ± 0.0003	0.0031 ± 0.0003
Reflected [18]	0.0233 ± 0.0019	0.0336 ± 0.0009	0.0411 ± 0.0039	0.0897 ± 0.0112	0.0231 ± 0.0011
MDM (ours)	0.0074 ± 0.0008	0.0169 ± 0.0016	0.0051 ± 0.0006	0.0040 ± 0.0003	0.0027 ± 0.0000
Constraint violation (%) ↓					
DDPM [2]	0.73 ± 0.12	14.40 ± 1.39	11.63 ± 0.90	27.53 ± 0.57	68.83 ± 1.66

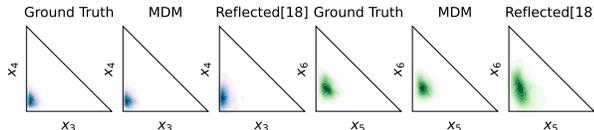


Figure 5: Comparison between MDM and Reflected Diffusion [18] in modeling a Dirichlet distribution on a 7-dimensional simplex. We visualize the joint densities between $x_{3:4}$ and $x_{5:6}$.

Table 5: Runtime and memory complexity w.r.t. DDPM [2].

	Runtime	Memory
MDM	108%	100%
Reflected Diff. [18]	>1200%	905%

5 Experiment

5.1 Constrained Generation on Balls and Simplices

Setup We evaluate the performance of **Mirror Diffusion Model (MDM)** on common constrained generation problems, such as ℓ_2 -ball and simplex constrained sets with dimensions d ranging from 2 to 20. Following Fishman et al. [18], we consider mixtures of Gaussian (Figure 1) and Spiral (Figure 3) for ball constraints and Dirichlet distributions [48] with various concentrations for simplex constraints. We compare MDM against standard unconstrained diffusion models, such as DDPM [2], and their constrained counterparts, such as Reflected Diffusion [18], using the same time-embedded fully-connected network and 1000 sampling time steps. Evaluation metrics include Sliced Wasserstein distance [70] and constraint violation. Other details are left to Appendix C.

MDM surpasses Reflected Diffusion on all tasks. Tables 3 and 4 summarize our quantitative results. For all constrained generation tasks, our MDM surpasses Reflected Diffusion by a large margin, with larger performance gaps as the dimension d increases. Qualitatively, Figure 5 demonstrates how MDM better captures the underlying distribution than Reflected Diffusion. Additionally, Table 5 reports the relative complexity of both methods compared to simulation-free diffusion models such as DDPM. While Reflected Diffusion requires extensive computation due to the use of implicit score matching [71] and assertion of boundary reflection at every propagation step, our MDM constructs efficient, parallelizable, mirror maps in unconstrained dual spaces (see Section 3), thereby enjoying preferable simulation-free complexity and better numerical scalability.

MDM matches DDPM without violating constraints. It should be highlighted that, while DDPM remains comparable to MDM in terms of distributional metric, the generated samples often violate the designated constrained sets. As shown in Tables 3 and 4, these constraint violations worsen as the dimension increases. In contrast, MDM is by design violation-free; hence marrying the best of both.

Table 6: The 50k-FID for unconditional watermarked generation. *Watermark precision* denotes the percentage of constraint-satisfied images that are generated by MDMs, controlled by loosening/tightening the constrained sets. As MDM-dual is trained on constraint-projected images, we also report its *FIDs** w.r.t. the shifted distributions. The prior watermarked diffusion model [39] reported 5.03 on FFHQ and 4.32 on AFHQv2.

	FFHQ 64×64			AFHQv2 64×64		
<i>Precision</i>	59.3%	71.8%	93.3%	56.9%	75.0%	92.7%
MDM-proj	2.54	2.59	3.08	2.10	2.12	2.30
MDM-dual	2.96	4.57	15.74	2.23	2.86	6.79
	2.65*	2.93*	4.40*	2.21*	2.32*	3.05*

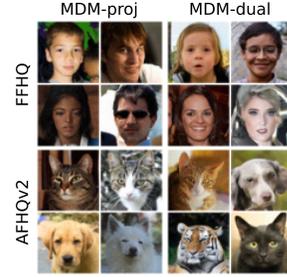


Figure 6: Uncurated watermarked samples by MDMs. More samples can be found in Appendix C.

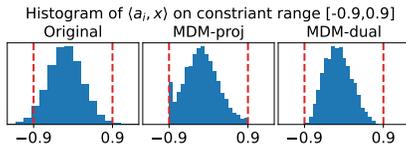


Figure 7: Histogram of $\langle a_i, x \rangle$ for original and MDM-watermarked images x . Dashed lines indicate the constrained set.

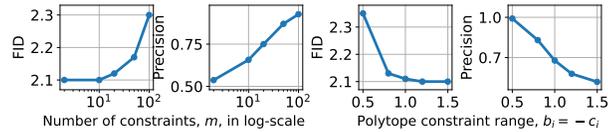


Figure 8: Ablation studies on how (top) number of constraints m and (bottom) their ranges, *i.e.*, (c_i, b_i) , affect the FID and precision of MDM-proj on AFHQv2.

5.2 Watermarked Generation on High-Dimensional Image Datasets

Setup We present MDM as a new approach for watermarked generation, where the watermark is attached to samples within an orthonormal polytope $\mathcal{M} := \{x \in \mathbb{R}^d : c_i < a_i^\top x < b_i, \forall i\}$. These parameters $\{a_i, b_i, c_i\}_{i=1}^m$ serve as the *private tokens* visible only to individual users (see Figure 2). While our approach is application-agnostic, we demonstrate MDM mainly on image datasets, including both unconditional and conditional generation. Given a designated watermark precision, defined as the percentage of constraint-satisfied images that are generated by MDMs, we randomly construct tokens as m orthonormalized Gaussian random vectors and instantiate MDMs by either projecting unwatermarked diffusion models onto \mathcal{M} (**MDM-proj**), or learning the corresponding dual-space diffusions (**MDM-dual**). More precisely, MDM-proj projects samples generated by pretrained diffusion models to a constraint set whose parameters (*i.e.*, tokens) are visible only to the private user. In contrast, MDM-dual *learns* a dual-space diffusion model from the constrained-projected samples; hence, like other MDMs in Section 5.1, it is constraint-dependent. Other details are left to Appendix C.

Unconditional watermark generation on FFHQ and AFHQv2 We first test out MMD on FFHQ [72] and AFHQv2 [73] on unconditional 64×64 image generation. Following Zhao et al. [39], we adopt EDM parameterization [38] and report in Table 6 the performance of MDM in generating watermarked images, quantified by the Fréchet Inception Distance (FID) [74]. Compared to prior watermarked approaches, which require additional training of latent spaces [39], **MDM stands as a competitive approach for watermarked generation** by directly constructing analytic dual spaces from the tokens (*i.e.*, constraints) themselves. Intriguingly, despite the fact that MDM-dual learns a smoother dual-space distribution compared to the truncated one from MDM-proj (see Figure 7), the latter consistently achieves lower (hence better) FIDs. Since samples generated by MDM-dual still remain close to the watermarked distribution, as indicated in Table 6 by the low *FIDs** w.r.t. the shifted training statistics, we conjecture that the differences may be due to the significant distributional shift induced by naively projecting the training data onto the constraint set. This is validated, partly, in the ablation study from Figure 8, where we observe that both the FID and watermark precision improve as the number of constraints m decreases, and as the constraint ranges (c_i, b_i) loosen. We highlight this fundamental trade off between watermark protection and generation quality. Examples of watermarked images are shown in Figure 6 and Appendix C.

Conditional generation on ImageNet256 Finally, we consider conditional watermarked generation on ImageNet 256×256. Specifically, we focus on image restoration tasks, where the goal



Figure 9: Conditional watermarked generation on ImageNet 256×256 . Specifically, we consider the JPEG restoration task, where, given degraded, low-quality inputs y (*upper-left*), we wish to generate their corresponding clean images x (*upper-right*) by learning $p(x|y)$. It is clear that both MDM-dual and MDM-proj are capable of solving this conditional generation task, generating clean images that additionally embed invisible watermarks, *i.e.*, $x \in \mathcal{M}$.

is to generate clean, watermarked, images conditioned on degraded—restored with JPEG in this case—inputs. Similar to unconditional generation, we consider a polytope constraint set whose parameters are chosen such that the watermark yields high precision ($> 95\%$) and low false positive rate ($< 0.001\%$). Specifically, we set $m = 100$ and $b = -c = 1.2$. We initialize the networks with pretrained checkpoints from Liu et al. [75].

Figure 9 reports the qualitative results. It is clear that both MDM-dual and MDM-proj are capable of solving this conditional generation task, generating clean images that additionally embed invisible watermarks, *i.e.*, $x \in \mathcal{M}$. Note that all non-MDM-generated images, despite being indistinguishable, actually violate the polytope constraint, whereas MDM-generated images always satisfy the constraint. Overall, our results suggest that both MDM-dual and MDM-proj scale to high-dimensional applications and are capable of embedding invisible watermarks in high-resolution images.

6 Conclusion and Limitation

We developed Mirror Diffusion Models (MDMs), a dual-space diffusion model that enjoys simulation-free training for constrained generation. We showed that MDM outperforms prior methods in standard constrained generation, such as balls and simplices, and offers a competitive alternative in generating watermarked contents. It should be noted that MDM concerns mainly *convex* constrained sets, which, despite their extensive applications, limits the application of MDM to general constraints. It will be interesting to combine MDM with other dynamic generative models, such as flow-based approaches.

Broader Impact

Mirror Diffusion Models (MDMs) advance the recent development of diffusion models to complex domains subjected to convex constrained sets. This opens up new possibilities for MDMs to serve as preferred models for generating samples that live in *e.g.*, simplices and balls. Additionally, MDMs introduce an innovative application of constrained sets as a watermarking technique. This has the potential to address concerns related to unethical usage and safeguard the copyright of generative models. By incorporating constrained sets into the generating process, MDMs offer a means to prevent unauthorized usage and ensure the integrity of generated content.

Acknowledgements

The authors would like to thank Bogdon Vlahov for sharing the computational resources. GHL, TC, ET are supported by ARO Award #W911NF2010151 and DoD Basic Research Office Award HQ00342110002. MT is partially supported by NSF DMS-1847802, NSF ECCS-1942523, Cullen-Peck Scholarship, and GT-Emory AI.Humanity Award.

References

- [1] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.
- [2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.
- [4] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [6] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2021.
- [7] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [8] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3D using 2D diffusion. In *International Conference on Learning Representations (ICLR)*, 2023.
- [9] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [10] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- [11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.
- [12] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [13] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.
- [14] Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [15] Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

- [16] Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [17] Ricky T. Q. Chen and Yaron Lipman. Riemannian flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.
- [18] Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael Hutchinson. Diffusion models for constrained domains. *Transactions on Machine Learning Research (TMLR)*, 2023.
- [19] Ben J Morris. Improved bounds for sampling contingency tables. *Random Structures & Algorithms*, 21(2):135–146, 2002.
- [20] Nathan E Lewis, Harish Nagarajan, and Bernhard O Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, 2012.
- [21] Li Han and Lee Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Robotics: Science and Systems (RSS)*, 2006.
- [22] Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2):3–9, 1985.
- [23] Ruth J Williams. Reflected brownian motion with skew symmetric data in a polyhedral domain. *Probability Theory and Related Fields*, 75(4):459–485, 1987.
- [24] Andrey Pilipenko. *An introduction to stochastic differential equations with reflection*, volume 1. Universitätsverlag Potsdam, 2014.
- [25] Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning (ICML)*, 2023.
- [26] Sebastien Bubeck, Ronen Eldan, and Joseph Lehec. Finite-time analysis of projected langevin monte carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [27] Kanji Sato, Akiko Takeda, Reiichiro Kawai, and Taiji Suzuki. Convergence error analysis of reflected gradient langevin dynamics for globally optimizing non-convex constrained problems. *arXiv preprint arXiv:2203.10215*, 2022.
- [28] Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [29] Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and geometry of Markov diffusion operators*, volume 103. Springer, 2014.
- [30] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2): 123–214, 2011.
- [31] Kelvin Shuangjian Zhang, Gabriel Peyré, Jalal Fadili, and Marcelo Pereyra. Wasserstein control of mirror langevin monte carlo. In *Conference on Learning Theory (COLT)*, 2020.
- [32] Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu, Philippe Rigollet, and Austin Stromme. Exponential ergodicity of mirror-Langevin diffusions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [33] Kwangjun Ahn and Sinho Chewi. Efficient constrained sampling via the mirror-Langevin algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [34] Ruilin Li, Molei Tao, Santosh S Vempala, and Andre Wibisono. The mirror langevin algorithm converges with vanishing bias. In *International Conference on Algorithmic Learning Theory (ALT)*, 2022.

- [35] Michael Liebrecht, Roman Schleifer, Anna Buadze, Dinesh Bhugra, and Alexander Smith. Generating scholarly content with chatgpt: ethical challenges for medical publishing. *The Lancet Digital Health*, 5(3):e105–e106, 2023.
- [36] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2023.
- [37] Jürgen Rudolph, Samson Tan, and Shannon Tan. Chatgpt: Bullshit spewer or the end of traditional assessments in higher education? *Journal of Applied Learning and Teaching*, 6(1), 2023.
- [38] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [39] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015.
- [41] Peter D Lax. Change of variables in multiple integrals. *The American mathematical monthly*, 106(6):497–501, 1999.
- [42] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.
- [43] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [44] Sébastien Bubeck and Ronen Eldan. The entropic barrier: a simple and optimal universal self-concordant barrier. In *Conference on Learning Theory (COLT)*, 2015.
- [45] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.
- [46] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.
- [47] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [48] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 2003.
- [49] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.
- [50] Lixin Lang, Wen-shiang Chen, Bhavik R Bakshi, Prem K Goel, and Sridhar Ungarala. Bayesian estimation via sequential monte carlo sampling—constrained dynamic systems. *Automatica*, 43(9):1615–1622, 2007.
- [51] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.
- [52] Yann Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305:805–808, 1987.
- [53] Nicolas Brosse, Alain Durmus, Éric Moulines, and Marcelo Pereyra. Sampling from a log-concave distribution with compact support with proximal Langevin Monte Carlo. In *Conference on Learning Theory (COLT)*, 2017.

- [54] Sam Patterson and Yee Whye Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [55] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.
- [56] Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Semi-discrete normalizing flows through differentiable tessellation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [57] Xingchao Liu, Lemeng Wu, Mao Ye, et al. Learning diffusion bridges on constrained domains. In *International Conference on Learning Representations (ICLR)*, 2022.
- [58] Mao Ye, Lemeng Wu, and Qiang Liu. First hitting diffusion models for generating manifold, graph and categorical data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [59] Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [60] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations (ICLR)*, 2023.
- [61] Adam Leach, Sebastian M Schmon, Matteo T Degiacomi, and Chris G Willcocks. Denoising diffusion probabilistic models on $SO(3)$ for rotational alignment. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2022.
- [62] Julen Urain, Niklas Funk, Georgia Chalvatzaki, and Jan Peters. $SE(3)$ -DiffusionFields: Learning cost functions for joint grasp and motion optimization through diffusion. In *IEEE International Conference on Robotics and Automation (ICRA), Workshop Track*, 2022.
- [63] Pierre H Richemond, Sander Dieleman, and Arnaud Doucet. Categorical SDEs with simplex diffusion. *arXiv preprint arXiv:2210.14784*, 2022.
- [64] Marco Caponigro, Anna Chiara Lai, and Benedetto Piccoli. A nonlinear model of opinion formation on the sphere. *Discrete & Continuous Dynamical Systems-A*, 35(9):4241–4268, 2015.
- [65] Jason Gaitonde, Jon Kleinberg, and Éva Tardos. Polarization in geometric opinion dynamics. In *ACM Conference on Economics and Computation*, pages 499–519, 2021.
- [66] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. A theory of the term structure of interest rates. In *Theory of valuation*, pages 129–164. World Scientific, 2005.
- [67] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [68] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *Transactions on Machine Learning Research (TMLR)*, 2022.
- [69] Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang, and Il-Chul Moon. Maximum likelihood training of implicit nonlinear diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [70] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.
- [71] Yang Song, Sahaj Garg, Jiabin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

- [72] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [73] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [74] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [75] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I²SB: Image-to-Image Schrödinger bridge. In *International Conference on Machine Learning (ICML)*, 2023.
- [76] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [77] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- [78] Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.
- [79] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

A Derivation of Mirror Mappings

Here, we provide addition derivation of $\nabla\phi^*$. Computation of $\nabla\phi(x)$ and $\nabla^2\phi^*(y)$ follow straightforwardly by differentiating $\phi(x)$ and $\nabla\phi^*(y)$ w.r.t. x and y , respectively.

ℓ_2 -Ball Since the gradient map also reverses the mirror map, we aim to rewrite $y = \frac{2\gamma}{R-\|x\|_2^2}x$ as $x = f(y) = \nabla\phi_{\text{ball}}^*(y)$. Solving the second-order polynomial,

$$\|y\|_2^2 = \left(\frac{2\gamma}{R-\|x\|_2^2}\right)^2 \|x\|_2^2, \quad (19)$$

yields

$$\|x\|_2^2 = R + \frac{2\gamma}{\|y\|_2^2} \left(\gamma - \sqrt{R\|y\|_2^2 + \gamma^2}\right). \quad (20)$$

With that, we can rewrite Equation (11) by

$$x = \frac{R - \|x\|_2^2}{2\gamma} y \stackrel{(20)}{=} \frac{\sqrt{R\|y\|_2^2 + \gamma^2} - \gamma}{\|y\|_2^2} y = \frac{R}{\sqrt{R\|y\|_2^2 + \gamma^2} + \gamma} y.$$

Simplex Standard calculations in convex analysis [28] shows

$$\phi_{\text{simplex}}^*(y) = \log \left(1 + \sum_i^d e^{y_i} \right). \quad (21)$$

Differentiating Equation (21) w.r.t. y yields $\nabla\phi_{\text{simplex}}^*$ in Equation (13).

Polytope Since the gradient map also reverses the mirror map, we aim to inverse

$$y = \sum_{i=1}^m s_i \langle a_i, x \rangle a_i + \sum_{j=m+1}^d \langle a_j, x \rangle a_j. \quad (22)$$

When all d constraints are orthonormal, taking inner product between y and each a yields

$$\langle a_i, y \rangle = s_i \langle a_i, x \rangle, \quad \langle a_j, y \rangle = \langle a_j, x \rangle. \quad (23)$$

Therefore, we can reconstruct x from y via

$$\begin{aligned} x &= \sum_{i=1}^m \langle a_i, x \rangle a_i + \sum_{j=m+1}^d \langle a_j, x \rangle a_j \\ &\stackrel{(23)}{=} \sum_{i=1}^m s_i^{-1} \langle a_i, y \rangle a_i + \sum_{j=m+1}^d \langle a_j, y \rangle a_j, \end{aligned}$$

which defines $x = \nabla\phi_{\text{polytope}}^*(y)$. For completeness, the Hessian can be presented compactly as

$$\nabla^2\phi_{\text{polytope}}^*(y) = \mathbf{I} + \mathbf{A}\Sigma\mathbf{A}^\top, \quad (24)$$

where \mathbf{I} is the identity matrix, $\mathbf{A} := [a_1, \dots, a_m]$ is a d -by- m matrix whose column vector a_i corresponds to each constraint, and $\Sigma \in \mathbb{R}^{m \times m}$ is a diagonal matrix with leading entries

$$[\Sigma]_{ii} = \frac{\partial s_i^{-1}(z)}{\partial z} \Big|_{z=\langle a_i, y \rangle} - 1 \stackrel{(18)}{=} \frac{b_i - c_i}{2} (1 - \tanh^2(\langle a_i, y \rangle)) - 1.$$

B Additional Remarks on Polytope

Derivation of Equation (17) Since the subspaces spanned by $\{a_i\}$ and $\{a_j\}$ are orthogonal to each other, we can rewrite (15) as

$$\nabla \phi_{\text{polytope}}(x) = \sum_{i=1}^m s_i(\langle a_i, x \rangle) a_i + \left(x - \sum_{i=1}^m \langle a_i, x \rangle a_i \right) = x + \sum_{i=1}^m (s_i(\langle a_i, x \rangle) - \langle a_i, x \rangle) a_i.$$

$\nabla \phi_{\text{polytope}}^*(y)$ follows similar derivation.

Generalization to non-orthonormal constraints The mirror maps of a polytope, as described in Equations (15) to (17), can be seen as operations that manipulate the coefficients associated with the bases defined by the constraints. This understanding allows us to extend the computation to non-orthonormal constraints by identifying the corresponding “coefficients” through a change of bases, utilizing the reproducing formula:

$$x = \sum_{i=1}^d \langle \tilde{a}_i, x \rangle a_i, \text{ where } \tilde{a}_i \text{ is the } i\text{-th row of } (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top,$$

and $\mathbf{A} := [a_1, \dots, a_m]$. Similarly, we have $y = \sum_{i=1}^d \langle \tilde{a}_i, y \rangle a_i$. Applying similar derivation leads to

$$\nabla \phi_{\text{poly}}(x) = x + \sum_{i=1}^m (s_i(\langle \tilde{a}_i, x \rangle) - \langle \tilde{a}_i, x \rangle) a_i, \quad \nabla \phi_{\text{poly}}^*(y) = y + \sum_{i=1}^m (s_i^{-1}(\langle \tilde{a}_i, y \rangle) - \langle \tilde{a}_i, y \rangle) a_i.$$

C Experiment Details & Additional Results

Table 7: The concentration parameter α of each Dirichlet distribution in simplices constrained sets.

d	3	3	7	9	20
α	[2, 4, 8]	[1, 0.1, 5]	[1, 2, 2, 4, 4, 8, 8]	[1, 0.5, 2, 0.3, 0.6, 4, 8, 8, 2]	[0.2, 0.4, \dots , 4, 4.2]

Table 8: Hyperparameters of the polytope $\mathcal{M} := \{x \in \mathbb{R}^d : c_i < a_i^\top x < b_i\}$ for each dataset and watermark precision. Note that we fix $b = b_i = -c_i$ in practice.

<i>Precision</i>	FFHQ 64×64 (uncon)			AFHQv2 64×64 (uncon)		
	59.3%	71.8%	93.3%	56.9%	75.0%	92.7%
Number of constraints m	7	20	100	4	20	100
Constraint range b	1.05	1.05	1.05	0.9	0.9	0.9

Dataset & constrained sets

- *ℓ_2 -balls constrained sets*: For $d = 2$, we consider the Gaussian Mixture Model (with variance 0.05) and the Spiral shown respectively in Figures 1 and 3. For $d = \{6, 8, 20\}$, we place d isotropic Gaussians, each with variance 0.05, at the corner of each dimension, and reject samples outside the constrained sets.
- *Simplices constrained sets*: We consider Dirichlet distributions [48], $\text{Dir}(\alpha)$, with various concentration parameters α detailed in Table 7.
- *Hypercube constrained sets*: For all dimensions $d = \{2, 3, 6, 8, 20\}$, we place d isotropic Gaussians, each with variance 0.2, at the corner of each dimension, and either reject ($d = \{2, 3, 6, 8\}$) or reflect ($d = 20$) samples outside the constrained sets.
- *Watermarked datasets and polytope constrained sets*: We follow the same data preprocessing from EDM⁵ [38] and rescale both FFHQ and AFHQv2 to 64×64 image resolution. For the polytope constrained sets $\mathcal{M} := \{x \in \mathbb{R}^d : c_i < a_i^\top x < b_i, \forall i\}$, we construct a_i from orthonormalized Gaussian random vectors and detail other hyperparameters in Table 8.

⁵<https://github.com/NVlabs/edm>, released under Nvidia Source Code License.

Implementation All methods are implemented in PyTorch [76]. We adopt ADM⁶ and EDM⁵ [38] respectively as the MDM’s diffusion backbones for constrained and watermarked generation. We implemented Reflected Diffusion [18] by ourselves as their codes have not yet been made available by the submission time (May 2023), and used the official implementation⁷ of Reflected Diffusion [25] in Table 11. We also implemented Simplex Diffusion [63], but as observed in previous works [25], it encountered computational instability especially when computing the modified Bessel functions.

Training For constrained generation, all methods are trained with AdamW [77] and an exponential moving average with the decay rate of 0.99. As standard practices, we decay the learning rate by the decay rate 0.99 every 1000 steps. For watermarked generation, we follow the default hyperparameters from EDM⁵ [38]. All experiments are conducted on two TITAN RTXs and one RTX 2080.

Network For constrained generation, all networks take (y, t) as inputs and follow

$$\text{out} = \text{out_mod}(\text{norm}(y_{\text{mod}}(y) + t_{\text{mod}}(\text{timestep_embedding}(t))))),$$

where $\text{timestep_embedding}(\cdot)$ is the standard sinusoidal embedding. t_{mod} and out_mod consist of 2 fully-connected layers (Linear) activated by the Sigmoid Linear Unit (SiLU) [78]:

$$t_{\text{mod}} = \text{out_mod} = \text{Linear} \rightarrow \text{SiLU} \rightarrow \text{Linear}$$

and y_{mod} consists of 3 residual blocks, *i.e.*, $y_{\text{mod}}(y) = y + \text{res_mod}(\text{norm}(y))$, where

$$\text{res_mod} = \text{Linear} \rightarrow \text{SiLU} \rightarrow \text{Linear} \rightarrow \text{SiLU} \rightarrow \text{Linear} \rightarrow \text{SiLU} \rightarrow \text{Linear}$$

All Linear’s have 128 hidden dimension. We use group normalization [79] for all norm. For watermarked generation, we use EDM parameterization⁵ [38].

Evaluation We compute the Wasserstein and Sliced Wasserstein distances using the `geomloss`⁸ and `ot`⁹ packages, respectively. The Maximum Mean Discrepancy (MMD) is based on the popular package https://github.com/ZongxianLee/MMD_Loss.Pytorch, which is unlicensed. For watermarked generation, we follow the same evaluation pipeline from EDM⁵ [38] by first generating 50,000 watermarked samples and computing the FID w.r.t. the training statistics.

False-positive rate Similar to Kirchenbauer et al. [36], we reject the null hypothesis and detect the watermark if the sample produces no violation of the polytope constraint, *i.e.*, if $x \in \mathcal{M}$. Hence, the false-positive samples are those that are actually true null hypothesis (*i.e.*, not generated by MDM) yet accidentally fall into the constraint set, hence being mistakenly detected as watermarked. Specifically, the false-positive rates of our MDMs are respectively 0.07% and 0.08% for FFHQ and AFHQv2. Lastly, we note that the fact that both MDM-proj and MDM-dual generate samples that always satisfy the constraint readily implies 100% recall and 0% Type II error.

C.1 Additional Results

Tractable variational bound in Equation (8) Figure 10 demonstrates how MDM faithfully captures the variational bound to the negative log-likelihood (NLL) of 2-dimensional GMM.

More constrained sets, distributional metrics, & baseline Tables 9 and 10 expand the analysis in Tables 3 and 4 with additional distributional metrics such as Wasserstein-1 (\mathcal{W}_1) and Maximum Mean Discrepancy (MMD). Additionally, Table 11 reports the results of hypercube $[0, 1]^d$ constrained set, a special instance of polytopes, and includes additional baseline from Lou and Ermon [25], which approximate the

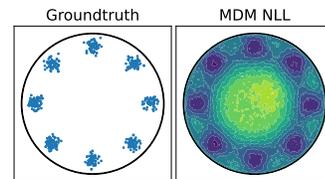


Figure 10: Tractable variational bound by our MDM.

⁶<https://github.com/openai/guided-diffusion>, released under MIT License.

⁷<https://github.com/louaaron/Reflected-Diffusion>, latest commit (65d05c6) at submission, unlicensed.

⁸<https://github.com/jeanfeydy/geomloss>, released under MIT License.

⁹https://pythonot.github.io/gen_modules/ot.sliced.html#ot.sliced.sliced_wasserstein_distance, released under MIT License.

intractable scores in Reflected Diffusion using eigenfunctions tailored specifically to hypercubes, rather than implicit score matching as in Fishman et al. [18]. Consistently, our findings conclude that the MDM is the *only* constrained-based diffusion model that achieves comparable or better performance to DDPM. These results affirm the effectiveness of MDM in generating high-quality samples within constrained settings, making it a reliable choice for constrained generative modeling.

More watermarked samples Figures 11 and 12 provide additional qualitative results on the watermarked samples generated by MDMs.



Figure 11: FFHQ 64×64 unconditional watermarked samples generated by **(left)** MDM-proj and **(right)** MDM-dual from the same set of random seeds. Despite the fact that some images, such as the one in the first row and sixth column, were altered possibly due to the change of dual-space distribution (see Figure 7), they look realistic and remain close to the data distribution.



Figure 12: AFHQv2 64×64 unconditional watermarked samples generated by **(left)** MDM-proj and **(right)** MDM-dual from the same set of random seeds. Despite the fact that some images, such as the one in the fifth row and first column, were altered possibly due to the change of dual-space distribution (see Figure 7), they all look realistic and remain close to the data distribution.

Table 9: Expanded results of ℓ_2 -ball constrained sets, where we include additional distributional metrics such as \mathcal{W}_1 and Maximum Mean Discrepancy (MMD), all computed with 1000 samples and averaged over three trials. Consistently, our findings conclude that the MDM is the *only* constrained-based diffusion model that achieves comparable or better performance to DDPM.

	$d = 2$	$d = 2$	$d = 6$	$d = 8$	$d=20$
$\mathcal{W}_1 \downarrow$ (unit: 10^{-2})					
DDPM [2]	0.66 ± 0.15	0.14 ± 0.03	0.52 ± 0.09	0.58 ± 0.10	3.45 ± 0.50
Reflected [18]	0.55 ± 0.29	0.46 ± 0.17	3.11 ± 0.40	10.13 ± 0.21	19.42 ± 0.13
MDM (ours)	0.46 ± 0.07	0.12 ± 0.04	0.72 ± 0.39	1.05 ± 0.26	2.63 ± 0.31
<hr/>					
$\text{MMD} \downarrow$ (unit: 10^{-2})					
DDPM [2]	0.67 ± 0.23	0.23 ± 0.07	0.37 ± 0.19	0.75 ± 0.24	0.98 ± 0.42
Reflected [18]	0.58 ± 0.46	5.03 ± 1.17	2.34 ± 0.14	28.82 ± 0.66	14.83 ± 0.62
MDM (ours)	0.52 ± 0.36	0.27 ± 0.19	0.54 ± 0.12	0.35 ± 0.23	0.50 ± 0.17
<hr/>					
Constraint violation (%) \downarrow					
DDPM [2]	0.00 ± 0.00	0.00 ± 0.00	8.67 ± 0.87	13.60 ± 0.62	19.33 ± 1.29

Table 10: Expanded results of simplices constrained sets.

	$d = 3$	$d = 3$	$d = 7$	$d = 9$	$d=20$
$\mathcal{W}_1 \downarrow$ (unit: 10^{-2})					
DDPM [2]	0.01 ± 0.00	0.02 ± 0.01	0.03 ± 0.00	0.05 ± 0.00	0.11 ± 0.00
Reflected [18]	0.06 ± 0.01	0.12 ± 0.00	0.62 ± 0.08	3.57 ± 0.05	0.98 ± 0.02
MDM (ours)	0.01 ± 0.00	0.01 ± 0.01	0.03 ± 0.00	0.05 ± 0.00	0.13 ± 0.00
<hr/>					
$\text{MMD} \downarrow$ (unit: 10^{-2})					
DDPM [2]	0.72 ± 0.07	0.72 ± 0.30	0.74 ± 0.10	0.97 ± 0.22	1.12 ± 0.07
Reflected [18]	3.91 ± 0.95	15.12 ± 1.36	16.48 ± 1.04	131.44 ± 2.65	57.90 ± 2.07
MDM (ours)	0.44 ± 0.16	0.50 ± 0.26	0.42 ± 0.08	0.55 ± 0.13	0.61 ± 0.03
<hr/>					
Constraint violation (%) \downarrow					
DDPM [2]	0.73 ± 0.12	14.40 ± 1.39	11.63 ± 0.90	27.53 ± 0.57	68.83 ± 1.66

Table 11: Results of hypercube $[0, 1]^d$ constrained sets.

	$d = 2$	$d = 3$	$d = 6$	$d = 8$	$d=20$
Sliced Wasserstein \downarrow (unit: 10^{-2})					
DDPM [2]	2.24 ± 1.22	2.17 ± 0.65	2.05 ± 0.41	2.01 ± 0.16	1.54 ± 0.01
Reflected [25]	3.75 ± 1.20	6.58 ± 1.18	2.77 ± 0.06	3.50 ± 0.69	3.37 ± 0.46
Reflected [18]	19.05 ± 1.51	17.16 ± 0.88	11.90 ± 0.43	7.49 ± 0.13	4.32 ± 0.23
MDM (ours)	3.00 ± 0.72	1.92 ± 0.81	1.75 ± 0.17	1.85 ± 0.34	3.35 ± 0.64
<hr/>					
$\mathcal{W}_1 \downarrow$ (unit: 10^{-2})					
DDPM [2]	0.07 ± 0.05	0.22 ± 0.07	1.65 ± 0.14	3.30 ± 0.16	16.74 ± 0.12
Reflected [25]	0.20 ± 0.12	1.21 ± 0.39	2.53 ± 0.04	4.82 ± 0.42	25.47 ± 0.20
Reflected [18]	4.40 ± 0.57	6.01 ± 0.97	9.34 ± 0.56	9.84 ± 0.24	25.27 ± 0.36
MDM (ours)	0.08 ± 0.03	0.20 ± 0.07	1.57 ± 0.08	3.34 ± 0.23	20.59 ± 1.19
<hr/>					
$\text{MMD} \downarrow$ (unit: 10^{-2})					
DDPM [2]	0.27 ± 0.26	0.32 ± 0.14	0.69 ± 0.21	0.81 ± 0.23	0.73 ± 0.07
Reflected [25]	0.92 ± 0.53	3.56 ± 1.31	1.16 ± 0.04	2.09 ± 0.70	2.83 ± 0.58
Reflected [18]	32.26 ± 3.19	26.64 ± 5.07	29.83 ± 1.42	15.84 ± 0.89	7.21 ± 0.68
MDM (ours)	0.27 ± 0.09	0.29 ± 0.17	0.39 ± 0.14	0.61 ± 0.23	0.62 ± 0.05
<hr/>					
Constraint violation (%) \downarrow					
DDPM [2]	9.37 ± 0.12	17.57 ± 1.27	41.70 ± 1.30	59.30 ± 1.39	94.47 ± 0.64