Gauge Flow Matching: Efficient Constrained Generative Modeling over General Convex Set and Beyond

Anonymous authors
Paper under double-blind review

Abstract

Generative models, particularly diffusion and flow-matching approaches, have achieved remarkable success across diverse domains, including image synthesis and robotic planning. However, a fundamental challenge persists: ensuring generated samples strictly satisfy problem-specific constraints — a crucial requirement for physics-informed problems, safety-critical applications, watermark embedding, etc. Existing approaches, such as mirror maps and reflection methods, either have limited applicable constraint sets or introduce significant computational overhead. In this paper, we develop gauge flow matching (GFM), a simple yet efficient framework for constrained generative modeling. Our GFM approach introduces a novel bijective gauge mapping to transform generation over arbitrary compact convex sets into an equivalent process over the unit ball, which allows low-complexity feasibility-ensuring operations such as reflection. The generated samples are then mapped back to the original domain for output. We prove that our GFM framework guarantees strict constraint satisfaction, with low generation complexity and bounded distribution approximation errors. We further extend our GFM framework to two popular non-convex settings, namely, star-convex and geodesic-convex sets. Extensive experiments show that GFM outperforms existing methods in generation speed and quality across multiple benchmarks.

1 Introduction

Generative models have emerged as powerful tools for learning complex data distributions, achieving remarkable success in diverse applications ranging from image generation to scientific simulation. Recent advances, particularly in diffusion models and flow-matching approaches, have further pushed the boundaries of what's possible in areas such as photorealistic image synthesis, molecular design, and robotic trajectory planning (Ramesh et al., 2022; Betker et al., 2023; Chi et al., 2023; Abramson et al., 2024; Zeni et al., 2025).

However, many real-world applications necessitate generation under specific constraints. For instance, protein synthesis requires adherence to structural constraints within amino acid chains. Image generation may demand precise watermark placement or consistency with physical laws. Robotic manipulation must respect joint limits and ensure obstacle avoidance. These constraints are not merely optional considerations but fundamental requirements of their respective problem domains. Generated samples must strictly satisfy these constraints to be both meaningful and practically useful within their intended applications.

Existing approaches to constrained generative modeling face significant limitations (see Table 1). They either have limited applicable constraints (e.g., box and simplex) or lack a strict feasibility guarantee for generated samples. To date, developing an efficient framework for constrained generation with feasibility guarantees over general compact sets, convex or not, remains largely open. This work proposes Gauge Flow Matching (GFM), addressing these challenges with the following contributions:

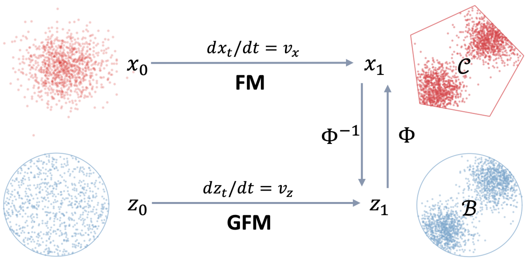


Figure 1: Gauge flow matching framework.

▷ In Sec. 4, we propose a *bi-Lipschitz* bijective gauge mapping, generalized from the one in (Tabas & Zhang, 2022a), to transform generation over general *compact convex* sets to an equivalent process over a unit ball, which allows low-complexity feasibility-ensuring operations such as reflection (Fishman et al., 2023).

▷ In Sec. 5, we prove that the GFM framework guarantees strict constraint satisfaction and bounded distribution approximation error while incurring significantly lower computational complexity compared to SOTA constrained generative models such as regular reflection or projection-based methods.

▷ In Sec. 6, we further extend GFM to certain non-convex constraints, in particular *star-convex* and *geodesic-convex* sets, while inheriting the computational merits in the convex setting.

▷ In Sec. 7, we provide extensive empirical studies to demonstrate our framework's efficiency in terms of feasibility, approximation capability, and inference complexity compared to SOTA constrained generative models.

2 RELATED WORK

Ensuring feasibility is fundamental in both theoretical and practical domains. Significant research efforts have been devoted to this area, including the development of constrained optimization algorithms and methods to guarantee neural network (NN) output feasibility (See Appendix A for a comprehensive review of these approaches).

For constrained generative modeling, feasibility strategies differ fundamentally between conventional and modern generative models. While traditional VAEs and GANs can directly incorporate existing NN output feasibility methods from Appendix A, diffusion and flow-based models present unique challenges. These challenges arise from their generation mechanism, which relies on forward integration with NN-approximated score functions or vector fields, rather than direct NN outputs. Table 1 summarizes specialized approaches addressing these challenges, with discussions as follows:

Reflected Process: These approaches leverage reflection mechanisms to constrain generation trajectories within feasible regions. Different methods have been proposed for training score functions under reflection terms: RDM^a employs implicit score matching (Fishman et al., 2023), RDM^b develops an approximated denoising score matching approach (Lou & Ermon, 2023), and RSB utilizes iterative proportional fitting (Deng et al., 2024). RFM extends this framework to flow-based generation over convex sets by incorporating reflected directions into ODEs (Xie et al., 2024). While effective, these reflection-based methods incur significant computational overhead during forward integration, requiring boundary localization and complex reflection calculations. A recent Metropolis sampling approach addresses the computational burden of reflection calculations through rejection sampling (Fishman et al., 2024), though it lacks strict feasibility guarantees. Further, for reflection-based models, the support set of initial distribution needs to be the same as the target set $\mathcal{C}$, which could cause significant additional complexity for both training and inference, since sampling from a general (convex) set $\mathcal{C}$ is non-trivial (Kook & Vempala, 2024).

Bijective Map: These approaches utilize bijective mappings to transform constrained domains. RDM^a maps simplexes to unit cubes, enabling scalable denoising score matching (Lou & Ermon, 2023). MDM employs mirror maps to transform constrained data space into unconstrained dual space (Liu et al., 2024b), though its applicability is limited to simple convex sets like balls and simplexes. NAMM generalizes this approach using neural networks to approximate an invertible map for general sets (Feng et al., 2024), though it lacks theoretical guarantees for feasibility and distribution approximation. Bijective mapping has also been applied in the continuous embedding of discrete categorical data (Davis et al., 2024). To date, such bijective mapping-based methods either work for very limited sets or lack theoretical guarantees.

Guided Generation: These methods incorporate auxiliary terms to guide the generation process toward constraint satisfaction. Ω-Bridge leverages Doob's h-transform to construct diffusion bridges over constrained domains, incorporating time-dependent force terms (Liu & Wu, 2023). Log-barrier diffusion models maintain feasibility through barrier functions (Fishman et al., 2023). PDM enforces constraints via iterative projection (Christopher et al., 2024), which has also been successfully applied for inverse problems (Chung et al., 2022; Song et al., 2023). The gradient of projection distance can also serve as the guidance term for constraint satisfaction (Naderiparizi et al., 2025). Despite their

Table 1: Existing study on constrained diffusion/flow-matching models over **continuous** domain.

Method (ref. in Sec. 2)	Constraint setting $\mathcal{C}$	Initial support set[1]	Feasibility guarantee[2]	Wasserstein bound	Low generation complexity[3]
DM/FM	-	$\mathbb{R}^n$	-	✓	$\mathcal{O}(\mathrm{NFE} \cdot n^2)$
RDM	Convex	$\mathcal{C}$	✓	-	✗
RSB	Smooth + Bounded	$\mathcal{C}$	✓	✓	✗
RFM	Convex	$\mathcal{C}$	✓	✓	✗
Metropolis sampling	Manifold	$\mathcal{C}$	✗	-	✗
MDM	Ball/Simplex	$\mathbb{R}^n$	✓	-	✓
NAMM	(Non)-Convex	$\mathbb{R}^n$	✗	-	✓
Barrier-based	Convex	$\mathcal{C}$	✓	-	✗
Projection-based	Convex	$\mathbb{R}^n$	✓	-	✗
Penalty-based	General	$\mathbb{R}^n$	✗	-	✓
Gauge Flow Matching	Convex	$\mathcal{B}$	✓	✓	✓

[1] The support set of initial/prior distribution matters when preparing the initial samples for training and generation, since sampling from the general convex set, even following a simple uniform distribution, is computationally expensive (Kook & Vempala, 2024).

[2] Feasibility guarantee means the generated samples satisfy the target constraints strictly. The feasibility guarantees in (Fishman et al., 2024) hold with probability for a certain stepsize.

[3] The low generation complexity of constrained generative models indicates the complexity is matched with regular DM/FM models, dominated by neural network evaluation ($\mathcal{O}(n^2)$) and scaled linearly with the number of function evaluations (NFE).

strict feasibility guarantees, solving projection is expensive, and there is a lack of analysis of the distribution approximation error for projection-guided generation.

Training/Fine-tuning: Penalty function for constraint violation can be incorporated into diffusion model training objectives to improve feasibility (Li et al., 2024). Lagrangian-based training with dual variable updates has also been applied to handle additional constraints (Khalafi et al., 2024). Post-training fine-tuning with reward functions (e.g., penalties) offers another pathway to improve constraint satisfaction (Fan & Lee, 2023; Uehara et al., 2024; Domingo-Enrich et al., 2024; Zhu et al., 2025).

In summary, existing works either have limited applicable scenarios or lack performance guarantees. In this work, we propose a novel gauge mapping-based approach for constrained generative modeling. While sharing conceptual similarities with mirror map-based methods, our approach distinguishes itself through the broader applicability, theoretical analysis, and computational efficiency.

3 Problem Statement

We consider flow matching-based generative modeling for a data distribution p_{data} over a general compact *convex* [1] set $\mathcal{C} \subset \mathbb{R}^n$. The vanilla flow-matching model (Lipman et al., 2022; Liu et al., 2022b) is trained by matching the designed conditional flow (e.g., linear flow) as:

$$\min \quad \mathcal{L}(v_\theta) = \mathbb{E}_{x_0, x_1, t}\left[\left\|v_\theta(x_t, t) - (x_1 - x_0)\right\|^2\right], \tag{1}$$

where $x_t = (1-t)x_0 + tx_1$ with $x_0 \sim p_0$, $x_1 \sim p_1$, and $t \sim \mathcal{U}((0,1))$. The minimizer of the flow matching loss in (1) yields a vector field that transforms a simple initial distribution, e.g., Gaussian $p_0 = \mathcal{N}(0, I)$, into the target data distribution $p_1 = p_{\mathrm{data}}$ (Liu et al., 2022b). In practice, the vector field is parameterized by a neural network v_θ and optimized using samples from the target distribution according to (1). Sample generation is achieved through forward integration $x_1 = x_0 + \int_0^1 v_\theta(x_t, t)dt$, initializing from a Gaussian sample x_0 and following the learned vector field v_θ.

[1] Compact convex set includes *linear-equality* and *convex-inequality* constraints. In this work, we consider the *convex-inequality* in the formulation without loss of generality. For *linear-equality*, it can be embedded in an unconstrained subspace by selecting independent variables and reconstructing the dependent variables via closed-from equality solving (Tordesillas et al., 2023; Donti et al., 2020; Liang et al., 2023; Ding et al., 2023), see Appendix B for details. For unbounded constraints, we may add additional box constraints to enforce physically meaningful limits.

Open issue: However, the generated samples often exhibit deviations from the constraint set $\mathcal{C}$ due to a phenomenon known as *error propagation* (Li & van der Schaar, 2023). This occurs when the approximation errors of NN-based vector fields accumulate throughout the discretized integration process, ultimately resulting in significant deviation of the generated samples from the constraint sets. Such phenomena have been observed in both flow-matching and diffusion-based models (Benton et al., 2023; Li et al., 2023b) and easily result in feasibility issues in the constrained generative tasks. Existing approaches addressing feasibility issues suffer from either limited applicability or high computational complexity (see Table 1).

4 Gauge Flow Matching over Convex Sets

To address these limitations of existing methods and enable efficient constrained generative modeling, we introduce our GFM framework. It employs gauge mapping—an explicit bijective mapping between two convex sets—to transform complex constrained generative modeling into an equivalent modeling over a simple unit ball. The framework **(i)** builds the flow-matching model for transformed data distribution over a unit ball through inverse gauge mapping; and **(ii)** generates samples over a unit ball via a closed-form reflection and transforms them back to the original space through forward gauge mapping.

4.1 Generalized Gauge Mapping between Convex Sets

We first introduce a bijective mapping between two compact convex sets in Euclidean space, known as gauge mapping (Tabas & Zhang, 2022a):

Definition 4.1 (Gauge mapping)**.** Let $\gamma_{\mathcal{C}}(x, x^\circ) = \inf\{\lambda \geq 0 \mid x \in \lambda(\mathcal{C} - x^\circ)\}$ be the *Gauge/Minkowski function* (Blanchini & Miani, 2008) given an interior point $x^\circ \in \mathrm{int}(\mathcal{C})$. The *gauge mapping* $\Phi : \mathcal{B} \to \mathcal{C}$ can be defined between a unit p-norm ball and a compact convex set:

$$\Phi(z) = \frac{\|z\|_p}{\gamma_{\mathcal{C}}(z, x^\circ)} z + x^\circ, \; \forall z \in \mathcal{B}, \qquad \Phi^{-1}(x) = \frac{\gamma_{\mathcal{C}}(x - x^\circ, x^\circ)}{\|x - x^\circ\|_p}(x - x^\circ), \; \forall x \in \mathcal{C}, \tag{2}$$

As shown in Fig. 2, the gauge mapping $\Phi(\cdot)$ establishes a continuous bijective correspondence (i.e., *homeomorphism*) between any compact convex set and a unit p-norm ball: $\mathcal{C} = \Phi(\mathcal{B})$ and $\mathcal{B} = \Phi^{-1}(\mathcal{C})$. Intuitively, the gauge mapping transforms a unit ball into the convex set by first *translating* the unit ball to the interior point ($0 \to x^\circ$), then *scaling* along every radial direction from this interior point such that the boundary of the ball becomes aligned with the boundary of the convex set (e.g., $z_1 \to x_1$). As a result, all level sets of $\mathcal{B}$ are mapped to level sets of $\mathcal{C}$ (e.g., $z_2 \to x_2$). The inverse mapping is similarly constructed by inverse scaling and translation back to the origin.

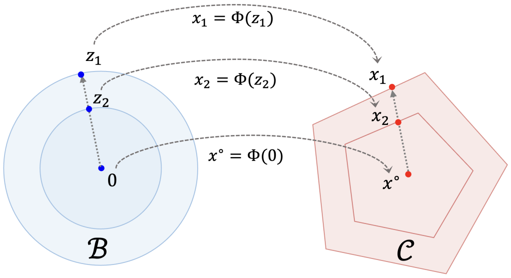


Figure 2: Gauge mapping illustration.

Remark 1 (Literature of Gauge function/map)**.** Gauge/Minkowski functions $\gamma_{\mathcal{C}}$ have been extensively studied in the control and online learning communities for Lyapunov function design and efficient online optimization (Blanchini, 1995; Raković & Lazar, 2014; Mhammedi, 2022; Lu et al., 2023). While gauge mapping Φ was proposed in (Tabas & Zhang, 2022a), specifically between a polytope and a cube, we generalize it to arbitrary pairs of compact convex sets and provide efficient computation methods. Specifically, in the gauge mapping calculation, (i) gauge function $\gamma_{\mathcal{C}}(x, x^\circ)$ has *closed-form* expressions for common convex sets (e.g., linear, quadratic, and convex cones) and can be efficiently computed via *bisection* methods for general convex constraints; (ii) the interior point x° is obtained *once for all* by solving a convex feasibility problem offline (see details in Appendix B).

Further, we characterize the *bi-Lipschitz* property of gauge mapping, as will be demonstrated in Sec. 5, which is essential for establishing the regularity of transformed data distribution and controlling the distribution approximation error in our GFM framework.

Proposition 4.1 (Bi-Lipschitz Property of the Gauge Mapping)**.** *Let $\mathcal{C} \subset \mathbb{R}^n$ be a compact convex set and let $x^\circ \in \operatorname{int}(\mathcal{C})$ be an interior point. Define the inner and outer radii with respect to x° as*

$$r_\mathrm{i} := \sup\{r \geq 0 : \mathcal{B}_2(x^\circ, r) \subseteq \mathcal{C}\}, \quad r_\mathrm{o} := \inf\{r \geq 0 : \mathcal{C} \subseteq \mathcal{B}_2(x^\circ, r)\},$$

such that $\mathcal{B}_2(x^\circ, r_\mathrm{i}) \subseteq \mathcal{C} \subseteq \mathcal{B}_2(x^\circ, r_\mathrm{o})$. Then the gauge mapping Φ between $\mathcal{C}$ and 2-norm ball $\mathcal{B}_2$ satisfies the following bounds:

$$\textit{Forward Lipschitz:} \quad \mathrm{L}_\Phi \leq 2\, r_\mathrm{o} + r_\mathrm{o}^2 / r_\mathrm{i}, \qquad \textit{Inverse Lipschitz:} \quad \mathrm{L}_{\Phi^{-1}} \leq 2/r_\mathrm{i} \tag{3}$$

The bi-Lipschitz property of gauge mapping depends on interior point selection. When a near-boundary interior point is selected (i.e., $r_\mathrm{i} \to 0$), the bi-Lipschitz constant approaches infinity, which severely distorts the data distribution and creates significant challenges for generative modeling. In practice, we aim to identify a "central" interior point (where r_o is close to r_i) to minimize the bi-Lipschitz constant. Such an interior point can be approximately determined through constraint residual minimization (Tordesillas et al., 2023), which involves linear optimization over the target convex set and can be solved in polynomial time. While we derive explicit Lipschitz bounds for gauge mapping from a 2-norm ball to a convex set, these bi-Lipschitz bounds can be extended to arbitrary pairs of compact convex sets. For a more detailed introduction to gauge mapping and its properties, we refer readers to Appendix B.

```
Algorithm 1 Training in GFM
Input: Data samples: $x_1 \sim p_{\mathrm{data}}$ over compact convex set $\mathcal{C}$, and gauge mapping between $\mathcal{C}$ and $\mathcal{B}$.
  1: Inverse transformation: $z_1 = \Phi^{-1}(x_1)$
  2: Regular flow matching over $\mathcal{B}$ in Eq. (4)
Output: Trained NN vector field $v_\theta$.
```

```
Algorithm 2 Generation in GFM
Input: Prior samples $z_0 \sim \mathcal{U}(\mathcal{B})$, NN vector field $v_\theta$, and gauge mapping between $\mathcal{C}$ and $\mathcal{B}$.
  1: ODE solver with reflection over $\mathcal{B}$ in Eq. (5)
  2: Forward transformation: $x_1 = \Phi(z_1)$
Output: Generated feasible samples $x_1$.
```

4.2 Training Phase of GFM

Given the gauge mapping Φ between the convex set $\mathcal{C}$ and a unit ball $\mathcal{B}$, a flow matching model is trained on the transformed space as:

$$\min \quad \mathcal{L}(v_\theta) = \mathbb{E}_{z_0, z_1, t}\left[\|v_\theta(z_t, t) - (z_1 - z_0)\|^2\right], \tag{4}$$

where the initial samples $z_0 \sim q_0$ are sampled as a simple prior distribution q_0 (e.g., uniform) supported on a unit ball $\mathcal{B}$, the terminal samples $z_1 \sim q_1$ are transformed from the samples from the original data distribution via inverse gauge mapping as $z_1 = \Phi^{-1}(x_1)$, and $z_t = (1-t)z_0 + t z_1$. In essence, GFM models the transformed data distribution as $q_{\mathrm{data}} = \Phi^{-1}_{\#} p_{\mathrm{data}}$ over a unit ball $\mathcal{B}$, where $\#$ is the push-forward operator. We then leverage the regular flow matching training approach to train a neural network vector field v_θ following (4).

Remark 2 (Comparison to Mirror Maps)**.** Mirror map-based generative models also employ a bijective mapping to transform constrained distributions to the unconstrained dual space (Liu et al., 2024b). However, it is only computationally tractable for simple convex sets (e.g., ball and simplex), and it maps near-boundary samples to *infinity* in the dual spaces, challenging the transformed generative modeling theoretically and practically.

In contrast, gauge mapping is computationally efficient for any compact convex set and maintains bounded Lipschitz constants for both sides (Prop. 4.1). These properties are crucial for the regularity of transformed data distribution and efficient flow-matching training, as will be analyzed in Sec. 5.

4.3 Inference Phase of GFM

After training, we generate samples within the unit ball $\mathcal{B}$ following the NN vector field v_θ. To constrain the generation trajectory within $\mathcal{B}$, we apply an additional reflection term (Xie et al., 2024):

$$z_1 = z_0 + \int_0^1 \big(v_\theta(z_t, t)\mathrm{d}t + \mathrm{d}\boldsymbol{L}_t\big), \tag{5}$$

where z_0 is sampled from a unit ball following a prior distribution (e.g., uniform), and $\boldsymbol{L}_t$ is the reflection term when z_t hits the constraint boundary (Xie et al., 2024). Finally, we recover the sample to the original space following the forward gauge mapping as $x_1 = \Phi(z_1)$.

Remark 3 (Comparison to Regular Reflection)**.** Reflection-based mechanisms are well-established in previous works to keep the generated samples within the constraint set (Lou & Ermon, 2023; Fishman et al., 2023; Deng et al., 2024; Xie et al., 2024). However, they face several limitations: **(i)** the reflection term is computationally expensive beyond simple sets (e.g., ball and simplex); **(ii)** existing reflection generative models require a prior distribution within the target constraint set $\mathcal{C}$, which is challenging to sample from (even for uniform distribution) over general convex sets during training or generation (Kook & Vempala, 2024). These computational issues prevent reflection methods from being applied to more complex sets.

In contrast, after transforming the data distribution over a unit ball through inverse gauge mapping, we can easily sample from a unit ball (e.g., uniformly), implement a closed-form reflection term with $\mathcal{O}(n)$ algorithmic complexity, and batch computation for multiple samples, thus ensuring efficient sample generation within the ball and strict feasibility after mapping to the original data space.

5 Performance Analysis of GFM

Proposition 5.1 (Regularity of Transformed Data Distribution)**.** *Assume that the original data distribution p_{data} satisfies regularity conditions (Assumption 1 in (Wan et al., 2024)). The transformed data distribution $q_{\mathrm{data}} = \Phi_{\#}^{-1} p_{\mathrm{data}}$ by a bi-Lipschitz homeomorphism (e.g., gauge mapping) also satisfies those regularity conditions.*

We first assume that the original data distribution p_{data} satisfies regularity conditions following (Wan et al., 2024), ensuring well-posedness of standard flow matching over the data distribution. Verifying the regularity of the transformed distribution $q_{\mathrm{data}} = \Phi_{\#}^{-1} p_{\mathrm{data}}$ is essential for establishing the existence and well-posedness of our gauge flow matching model. Under the bi-Lipschitz properties of gauge mapping established in Prop. 4.1, we can verify that these regularity conditions are preserved.

It is worth noting that theoretical analyses of flow matching models in the literature assume various regularity conditions for data distributions (Benton et al., 2023; Gao et al., 2024a;b). We verify a general condition from recent work (Wan et al., 2024). Detailed proofs and further discussion on other common regularity conditions are provided in Appendix D.

Proposition 5.2 (Wasserstein Bound of GFM)**.** *Let the NN approximation error be $\epsilon_\theta^2 = \mathbb{E}_{t,z_t}\|v_\theta(z_t,t) - u(z_t,t)\|^2$, where $z_t \sim p_t$ and p_t is the probability density at time t driven by the target vector field u. Assume that v_θ is L_θ-Lipschitz for $z \in \mathcal{B}$ and $t \in [0,1]$. Denote the induced probability distribution p_θ^{gr} under v_θ with reflected generation, the Wasserstein-2 distance between the data distribution p_{data} and the approximated distribution p_θ^{gr} is bounded by $\mathcal{W}_2(p_{data}, p_\theta^{gr}) \leq \mathrm{L}_\Phi \mathrm{e}^{1/2+\mathrm{L}_\theta}\epsilon_\theta$.*

The Wasserstein error of GFM is bounded by the Lipschitz of gauge mapping multiplied by the distribution error in the transformed space under reflected generation. To reduce the distribution approximation error, we can regularize the neural network Lipschitz constant L_θ or optimize the loss function such that ϵ_θ is minimized, following standard flow matching training pipelines (Liu et al., 2022b). Specific to our GFM framework, we can further optimize the Lipschitz constant of the gauge mapping L_Φ. As discussed following Prop. 4.1, we can select a "central" interior point for the gauge mapping by solving convex optimization problems, thereby reducing the bi-Lipschitz constants.

Proposition 5.3 (Inference Complexity for GFM)**.** *Consider a compact convex set $\mathcal{C} \subset \mathbb{R}^n$ defined by constraints $g_i(x) \leq 0$, for $i = 1, 2, \ldots, m$. The generation complexity of GFM (forward integration and gauge mapping calculation) is $\mathcal{O}(\mathrm{NFE} \cdot n^2 + m \cdot C)$, where $C = \max_{1\leq i\leq m}\{C_i\}$ is for gauge function calculation and varies by constraint type:*

▷ *(i) For linear constraints $g_i(x) = a^\top x - b \leq 0$, $C_i = \mathcal{O}(n)$; (ii) For quadratic constraints $g_i(x) = x^\top Qx + a^\top x - b \leq 0$, $C_i = \mathcal{O}(n^2)$; (iii) For second-order cone constraints $g_i(x) = \|A^\top x + p\|_2 - (a^\top x + b) \leq 0$, $C_i = \mathcal{O}(nk)$; (iv) For matrix cone constraints $g_i(x) = \sum_{j=1}^n x_j \cdot F_j + F_0 \succeq 0$, $C_i = \mathcal{O}(nk^2 + k^3)$; where $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, $Q \in \mathbb{S}_{++}^{n\times n}$, $A \in \mathbb{R}^{n\times k}$, $p \in \mathbb{R}^k$, and $F_j \in \mathbb{R}^{k\times k}$.*

▷ *For general convex function $g_i(x)$, $C_i = \mathcal{O}(c_i \log \epsilon_{\mathrm{bis}}^{-1})$ using bisection, where c_i is the complexity to evaluate $g_i(\cdot)$ given a point and ϵ_{bis} is the error tolerance in bisection.*

The forward integration complexity of our model aligns with regular flow matching approaches, requiring NFE (Number of Function Evaluations) multiplied by the evaluation complexity of v_θ. The additional reflection over a unit ball incurs negligible overhead with $\mathcal{O}(n)$ complexity compared to the NN forward calculation ($\mathcal{O}(n^2)$). For gauge mapping computation, given that the interior point is pre-computed offline, (i) for common convex sets, it can be solved efficiently (Fig. 8), and (ii) in general case, the bisection algorithm achieves linear convergence with minimal per-iteration complexity, merely requiring calculating the constraint function without solving any optimization problem (Mhammedi, 2022).

6 Gauge Flow Matching beyond Convex Sets

We extend the gauge mapping principle to two important classes of non-convex constraint sets, broadening GFM's applicability to richer geometric settings. We include formal definitions in Appendix C and provide empirical studies in Sec. 7.

▷ **Star-convex** sets are bounded regions where the entire boundary is visible from a designated interior point (e.g., $\ell_{0.5}$-norm ball). These sets arise in robotic planning and chance-constrained optimization (Charnes & Cooper, 1959; Hansen et al., 2020; Liu et al., 2022a). The key insight is that compact star-convex sets preserve the essential properties needed for gauge function construction—both the gauge function and bijective mapping to the unit ball extend naturally from convex case (Licht, 2024).

▷ **Geodesic-convex** sets are regions on Riemannian manifolds where geodesics connecting any two points are unique and remain entirely within the set. These constraints widely arise in geometric learning applications (Chen & Lipman, 2023; Miller et al., 2024; Zaghen et al., 2025). The key insight is that geodesic-convexity ensures the exponential map at any interior point provides a local diffeomorphism (Lee, 2006), allowing gauge mapping construction in the tangent space.

7 Empirical Study

We conduct extensive simulations across illustrative examples, robotics benchmarks, and high-dimensional constrained sampling tasks to demonstrate the effectiveness of our GFM framework. Detailed experimental settings and data descriptions are provided in Appendix E.

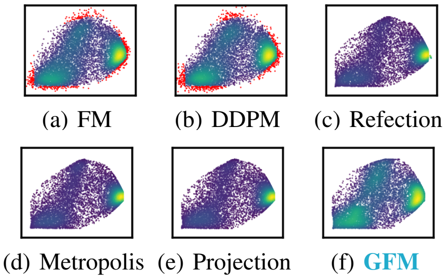
Baselines: We compare against the following constrained generative models, selected for their applicability to diverse constraint types: (i) **FM**: Vanilla flow matching from Gaussian to target distribution with linear conditional flow (Lipman et al., 2022; Liu et al., 2022b); (ii) **DM**: Vanilla diffusion model with variance preserving diffusion process (Ho et al., 2020; Song et al., 2020); (iii) **Reflection**: Reflection term is applied when generated samples hit the constraint boundary (Xie et al., 2024); (iv) **Metropolis**: Metropolis sampling for approximating reflection-based generation (Fishman et al., 2024); (v) **Projection**: Orthogonal projection are applied when generated samples violate constraints (Christopher et al., 2024); (vi) **GFM**: Our framework in Sec. 4.

Metrics: We evaluate those baselines based on (i) constraint satisfaction (i.e., feasibility) ratio (%) of 10,000 generated samples, (ii) distribution approximation error, which is measured by Maximum Mean Discrepancy (MMD) between data samples and generated samples (Fishman et al., 2023), (iii) average per-epoch training time, including prior sampling, data transformation (in GFM only), and NN training, and (iv) inference time for generating 1,000 batched samples (unless otherwise specified, including prior sampling, forward integration, and data transformation (in GFM only).

7.1 Synthetic Examples: Convex, Star-convex, and Geodesic-convex Set

We first evaluate GFM's performance in *convex*, *star-convex*, and *geodesic-convex* domains. As shown in Figures 3 - 5, we observe: vanilla diffusion/flow-matching models fail to guarantee sample feasibility, especially when probability density concentrates near boundaries. Existing reflection, projection, and Metropolis methods face limitations with complex constraints due to: (i) expensive prior distribution sampling, *increasing both training and inference time*; (ii) constraint-specific implementation requirements incurring *longer inference time* and *limited constraint settings*.

Our GFM framework transforms generative modeling over diverse convex sets into a simpler process over a unit ball, enabling efficient prior sampling and batch-executable reflection calculations with computational efficiency comparable to vanilla FM. After generation within the unit ball, samples are mapped back to the target constraint set via low-complexity gauge mappings, ensuring strict feasibility. Further, due to the bi-Lipschitz property (Prop. 4.1) of the gauge mapping, the approximation error of GFM remains comparable to vanilla FM. This property ensures that the distortion introduced by our transformation is bounded and controlled, preserving the fidelity of the generated samples while guaranteeing their feasibility within the target constraint set.



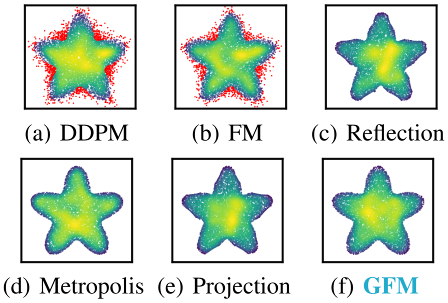
(a) FM (b) DDPM (c) Refection

(d) Metropolis (e) Projection (f) GFM

	Feasibility	MMD↓	Training	Inference
	(%)	$(\times 10^{-3})$	(s)	(s)
DDPM	95.0	4.79	0.17	0.59
FM	95.9	8.57	0.18	0.29
Refection	100	25.9	6.40	14.0
Metropolis	100	130	6.40	6.12
Projection	100	93.5	6.40	7.12
GFM	**100**	**3.50**	0.18	0.63

[1] Reflection, Metropolis, and Projection models share the same NN-velocity model, and thus have the same training time per epoch.

Figure 3: Performance over joint linear and quadratic convex sets



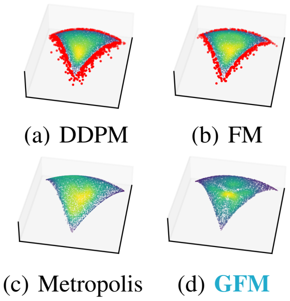
(a) DDPM (b) FM (c) Reflection

(d) Metropolis (e) Projection (f) GFM

	Feasibility	MMD↓	Training	Inference
	(%)	$(\times 10^{-3})$	(s)	(s)
DDPM	91.1	6.03	0.20	0.70
FM	93.4	4.92	0.22	0.39
Reflection	100	7.96	5.58	12.4
Metropolis	100	24.6	5.58	8.72
Projection	100	7.89	5.58	347
GFM	**100**	5.01	0.22	0.74

[1] Sampling from a prior distribution over a non-convex domain is time-consuming for reflection and Metropolis methods.
[2] Projection onto non-convex sets incurs significant computational cost.

Figure 4: Performance over star-convex sets



(a) DDPM (b) FM

(c) Metropolis (d) GFM

	Feasibility	MMD↓	Training	Inference
	(%)	$(\times 10^{-3})$	(s)	(s)
DDPM	93.4	2.84	0.22	0.73
FM	96.1	1.20	0.21	0.35
Metropolis	100	48.3	3.57	213
GFM	**100**	1.29	0.23	1.44

[1] Sampling from prior distributions over constrained manifolds is computationally expensive, resulting in long Metropolis inference times.
[2] High rejection rates further increase Metropolis inference time.
[3] Reflection and projection methods for constrained manifolds are not implemented due to computational issues.

Figure 5: Performance over geodesic-convex sets

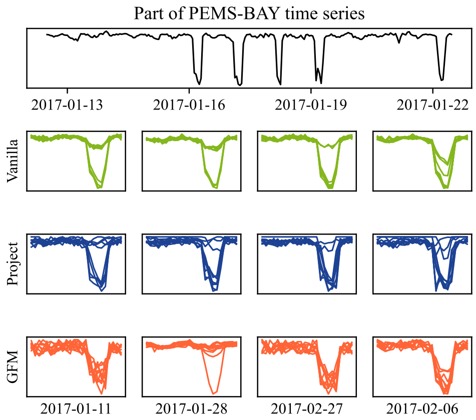
7.2 Constrained Configurational Modeling of Robotic Arms

We apply GFM to robotic arm control tasks following (Fishman et al., 2023; 2024). This involves learning distributions over joint locations and manipulability ellipsoids represented by symmetric positive-definite matrices with trace constraints. Figure 11 (Appendix E.6) shows generated velocity manipulation ellipsoids and trajectories. GFM successfully models this distribution while maintaining constraint satisfaction.

7.3 Constrained Time Series Generation for Traffic Data

Time series prediction is crucial for engineering applications, yet real-world data often suffers from imbalance and scarcity due to sensor failures or privacy concerns. Synthetic data provides a promising solution for augmenting limited datasets (Narasimhan et al., 2024). We evaluate GFM on generating realistic traffic sequences from PEMS-BAY (Li et al., 2018) subject to physical (e.g., capacity bounds) and statistical (e.g., average volume) constraints that define a convex second-order cone, ensuring physically plausible and statistically consistent sequences. See details in Appendix E.8.

Table 6 shows that GFM achieves **100%** constraint satisfaction versus 88.5% for vanilla flow matching, demonstrating the necessity of constraint-aware generation. Projection baselines also achieve 100% feasibility but increase inference time from 0.31s to 49.7s (160× slower), while GFM requires only 0.43s. Critically, GFM preserves superior distribution quality with a KS statistic of 0.35 (p-value 0.42) compared to projection methods. Figure 6 confirms GFM captures characteristic traffic



Methods	FM	Proj.	**GFM**
Feasibility (%)	88.5	100	**100**
KS ↓	0.30	0.37	0.35
p-value ↑	0.43	0.20	0.42
Training (s)	0.07	0.08	0.09
Inference (s)	0.31	49.7	0.43

[1] The Kolmogorov–Smirnov (KS) statistic measures the maximum distance between two empirical cumulative distribution functions. A KS value of 0 indicates identical distributions.

[2] The p-value is the probability of observing a KS statistic at least as large as the computed value under the identical distributions hypothesis. Larger p-values suggest stronger evidence of distributional similarity.

[3] Inference time is measured in a single generation with batch size 600.

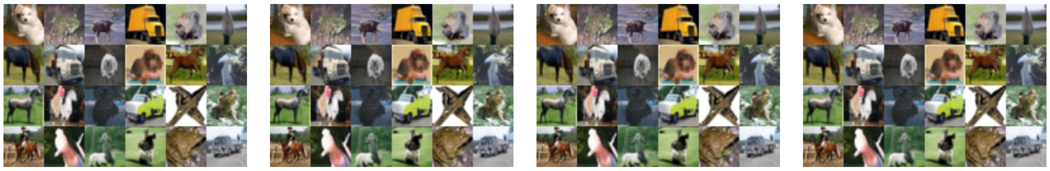
Figure 6: Performance over constrained conditional time series generations.

patterns—periodic fluctuations and sharp drops—while strictly satisfying constraints, achieving optimal balance between feasibility, fidelity, and efficiency for practical traffic data augmentation.

7.4 Constrained Image Generation with Embedded Watermarks

Watermarking generative models is critical for provenance tracking and copyright protection in AI-generated content. Following (Liu et al., 2024b), we partition pixels $\mathbf{x} \in [0, 1]^d$ into unconstrained public pixels $\mathbf{x}_1$ and watermark pixels $\mathbf{x}_2 \in \mathcal{C}$, where $\mathcal{C}$ is a random polytope for watermarking. Baselines are fine-tuned from a publicly available checkpoint of flow matching models (Tong et al., 2024), which share the same U-Net architecture (34M), fine-tuning hyperparameters (200 epochs), and sampling settings (100-step Euler method) on CIFAR-10. See details in Appendix E.9.

Table 7 shows that GFM maintains **100%** constraint satisfaction, achieves better FID than other constrained generative models (projection or reflection), while having similar training/inference complexity to vanilla methods. This demonstrates that GFM enables efficient, high-quality watermark embedding without compromising visual fidelity or constraint guarantees.



(a) FM (b) Projection (c) Reflection (d) GFM

Methods	FID (50K)	Feasibility (%)	Epoch training (s)	5K images inference (s)	Final transformation (s)
FM	6.63	76.66	76.1	159.9	–
Projection	11.24	100	110.7	596.1	–
Reflection	10.86	100	107.9	171.8	–
GFM	10.51	100	79.5	160.6	0.004

[1] We select 32 pixels as watermarks constrained via a polytope. The selected pixels in the original dataset are projected onto the polytope, while other public pixels remain unchanged in the watermarked dataset.

[2] Mirror map implementation requires an orthogonal basis polytope (Liu et al., 2024b) and is excluded for randomly sampled polytopes with general coefficients.

Figure 7: Constrained Image Generation with Embedded Watermarks

7.5 Solution Generation for (Relaxed) Combinatorial Problems

We further evaluate our approach on high-dimensional (n =10,000) solution generation for relaxed combinatorial problems (Kook & Vempala, 2024). The target distribution follows a log-concave density and is constrained by a positive semidefinite cone and a set of linear inequalities. This target distribution encapsulates several important classes of semidefinite relaxations for classical combinatorial optimization problems (e.g, max-cut and minimum-volume covering problem). We prepare the dataset following this distribution described in (Kook & Vempala, 2024) and train our GFM model. As shown in Table 7, our method achieves **100%** feasibility rate while vanilla models fail to satisfy the constraints. GFM also maintains generation quality comparable to standard flow-matching models.

Table 7: Solution sampling for relaxed combinatorial problems.

Method	DDPM	FM	**GFM**	DDPM	FM	**GFM**	DDPM	FM	**GFM**
	$n = 10 \times 10$			$n = 50 \times 50$			$n = 100 \times 100$		
Feasibility (%)	47.2	0	**100**	0	0	**100**	0	0	**100**
MMD ($\times 10^{-2}$)	5.35	9.71	9.70	43.4	43.4	43.5	85.8	85.8	85.8
Training (s)	0.27	0.28	0.28	0.28	0.28	0.28	0.67	0.91	0.91
Inference (s)	1.08	0.65	1.12	2.02	1.62	10.29	6.03	5.16	12.75

Regarding inference time, the gauge mapping has an explicit-form computation, with computational cost primarily stemming from matrix decomposition operations (via torch.linalg.eigvalsh) for `1,000` samples in a batch. This overhead can be further reduced through the application of advanced linear algebra packages (Van de Geijn, 1997). We do not implement reflection methods since sampling the prior distribution in such a high-dimensional constraint set is computationally expensive (e.g., if we apply Ball walk, it will incur $\mathcal{O}(n^8 \log n)$ mixing complexity (Kook & Vempala, 2024)). For the projection methods, it will solve 10^5 projection problems over the PSD cone at most (`1,000` samples and `100` integration steps), which is unaffordable on our computing device.

7.6 Scalability Tests and Ablation Study

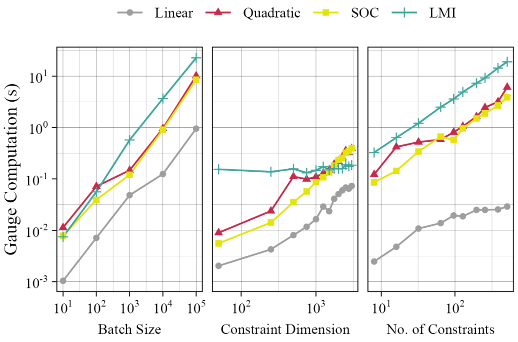


Figure 8: Gauge function computation.

Scalability of gauge mapping: We evaluate GFM's scalability in high-dimensional settings by measuring the computational cost of gauge function calculations. For closed-form gauge calculation on four common convex sets (Fig. 8), our approach maintains efficiency across dimensions up to `3,000`, demonstrating practical applicability for high-dimensional tasks. For bisection-based gauge calculation on polynomial constraints (sum-of-squares formulation, Table 12), a degree-4 polynomial with 50 variables containing `1,758,276` monomial terms can efficiently compute gauge functions for `1,000` samples in `0.492` seconds. Details are provided in Appendix E.10.

Impacts of the interior point (IP) selection: Following Prop. 4.1, the IP selection determines the bi-Lipschitz property of gauge mapping and consequently affects generation quality. We compare *central* versus *near-boundary* IPs to demonstrate this claim, with results in Table 13.

Comparison to Mirror Map: We compare the Mirror Map and Gauge Map in terms of generative modeling over the simplex in 2 and 50 dimensions, with results presented in Appendix E.5.

Impacts of generation strategies: While we adopt *reflection* to keep samples within the unit ball (Sec. 4.3), alternatives like *projection* can also be efficiently implemented. Table 13 shows that projection is faster but leads to a larger MMD, which motivates theoretical analysis in future works.

8 Concluding and Limitations

We introduced Gauge Flow Matching (GFM), a framework that transforms generative modeling over arbitrary compact convex sets into an equivalent process over the unit ball through a generalized gauge mapping. This approach guarantees strict constraint satisfaction with low computational complexity and bounded distribution approximation errors, and can be extended to important non-convex settings. Our experiments demonstrate that GFM outperforms existing methods in both speed and sample quality across multiple benchmarks. Despite these advances, several limitations warrant future investigation. (i) Extending the framework to more general non-convex sets presents an important direction, which could be realized through decomposition of the non-convex set into disjoint convex or star-convex subsets, followed by conditional generative modeling over those subsets. (ii) While our primary focus is on continuous domains, extending GFM to discrete generation through relaxation or embedding techniques represents a promising avenue for future work (Davis et al., 2024). (iii) Adapting GFM to constrained one-step generation models could further improve generation efficiency beyond the current multi-step approach (Song et al., 2023; Frans et al., 2024).

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145. PMLR, 2017.

Claude JP Bélisle, H Edwin Romeijn, and Robert L Smith. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2):255–266, 1993.

Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *arXiv preprint arXiv:2305.16860*, 2023.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Franco Blanchini. Nonquadratic lyapunov functions for robust control. *Automatica*, 31(3):451–461, 1995.

Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*, volume 78. Springer, 2008.

Abraham Charnes and William W Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

Bingqing Chen, Priya L Donti, Kyri Baker, J Zico Kolter, and Mario Bergés. Enforcing policy feasibility constraints through differentiable projection for energy optimization. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pp. 199–210, 2021.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3387–3395, 2019.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

Oscar Davis, Samuel Kessler, Mircea Petrache, Ismail Ilkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. *arXiv preprint arXiv:2405.14664*, 2024.

Wei Deng, Yu Chen, Nicole Tianjiao Yang, Hengrong Du, Qi Feng, and Ricky TQ Chen. Reflected schrödinger bridge for constrained generative modeling. *arXiv preprint arXiv:2401.03228*, 2024.

Shutong Ding, Jingya Wang, Yali Du, and Ye Shi. Reduced policy optimization for continuous control with hard constraints. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2020.

Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.

Berthy T Feng, Ricardo Baptista, and Katherine L Bouman. Neural approximate mirror maps for constrained diffusion models. *arXiv preprint arXiv:2406.12816*, 2024.

Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael Hutchinson. Diffusion models for constrained domains. *arXiv preprint arXiv:2304.05364*, 2023.

Nic Fishman, Leo Klarner, Emile Mathieu, Michael Hutchinson, and Valentin De Bortoli. Metropolis sampling for constrained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Thomas Frerix, Matthias Nießner, and Daniel Cremers. Homogeneous linear inequality constraints for neural network activations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 748–749, 2020.

Yuan Gao, Jian Huang, and Yuling Jiao. Gaussian interpolation flows. *Journal of Machine Learning Research*, 25(253):1–52, 2024a.

Yuan Gao, Jian Huang, Yuling Jiao, and Shurong Zheng. Convergence of continuous normalizing flows for learning probability distributions. *arXiv preprint arXiv:2404.00551*, 2024b.

Guillermo Hansen, Irmina Herburt, Horst Martini, and Maria Moszyńska. Starshaped sets. *Aequationes mathematicae*, 94:1001–1092, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Noémie Jaquier, Leonel Rozo, Darwin G Caldwell, and Sylvain Calinon. Geometry-aware manipulability learning, tracking, and transfer. *The International Journal of Robotics Research*, 40(2-3): 624–650, 2021.

Shervin Khalafi, Dongsheng Ding, and Alejandro Ribeiro. Constrained diffusion models via dual training. *arXiv preprint arXiv:2408.15094*, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andrei V Konstantinov and Lev V Utkin. A new computationally simple approach for implementing neural networks with output hard constraints. In *Doklady Mathematics*, volume 108, pp. S233–S241. Springer, 2023.

Yunbum Kook and Santosh S Vempala. Gaussian cooling and dikin walks: The interior-point method for logconcave sampling. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 3137–3240. PMLR, 2024.

Anastasis Kratsios, Behnoosh Zamanlooy, Tianlin Liu, and Ivan Dokmanić. Universal approximation under constraints is possible with transformers. In *International Conference on Learning Representations*, 2021.

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the wasserstein distance. *arXiv preprint arXiv:2212.06359*, 2022.

John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

Anjian Li, Zihan Ding, Adji Bousso Dieng, and Ryne Beeson. Efficient and guaranteed-safe non-convex trajectory optimization with constrained diffusion model. *arXiv preprint arXiv:2403.05571*, 2024.

Meiyi Li and Javad Mohammadi. Toward rapid, optimal, and feasible power dispatch through generalized neural mapping. *arXiv preprint arXiv:2311.04838*, 2023.

Meiyi Li, Soheil Kolouri, and Javad Mohammadi. Learning to solve optimization problems with hard linear constraints. *IEEE Access*, 2023a.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

Yangming Li and Mihaela van der Schaar. On error propagation of diffusion models. *arXiv preprint arXiv:2308.05021*, 2023.

Yangming Li, Zhaozhi Qian, and Mihaela van der Schaar. Do diffusion models suffer error propagation? theoretical analysis and consistency regularization. *arXiv preprint arXiv:2308.05021*, 2023b.

Enming Liang and Minghua Chen. Efficient bisection projection to ensure neural-network solution feasibility for optimization over general set. In *International Conference on Machine Learning*. PMLR, 2025.

Enming Liang, Minghua Chen, and Steven H. Low. Low complexity homeomorphic projection to ensure neural-network solution feasibility for optimization over (non-)convex set. In *International Conference on Machine Learning*. PMLR, 2023.

Enming Liang, Minghua Chen, and Steven H. Low. Homeomorphic projection to ensure neural-network solution feasibility for constrained optimization. *Journal of Machine Learning Rsearch*, 2024.

Martin Werner Licht. On the geometry of star domains and the spectra of hodge-laplace operators under domain approximation. *arXiv preprint arXiv:2411.01655*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Chenghao Liu, Enming Liang, and Minghua Chen. Characterizing resnet's universal approximation capability. In *Forty-first International Conference on Machine Learning*, 2024a.

Guan-Horng Liu, Tianrong Chen, Evangelos Theodorou, and Molei Tao. Mirror diffusion models for constrained and watermarked generation. *Advances in Neural Information Processing Systems*, 36, 2024b.

Tianyu Liu, Qianhao Wang, Xingguang Zhong, Zhepei Wang, Chao Xu, Fu Zhang, and Fei Gao. Star-convex constrained optimization for visibility planning with application to aerial inspection. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7861–7867. IEEE, 2022a.

Xingchao Liu and Lemeng Wu. Learning diffusion bridges on constrained domains. In *international conference on learning representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pp. 22675–22701. PMLR, 2023.

Zhou Lu, Nataly Brukhim, Paula Gradu, and Elad Hazan. Projection-free adaptive regret with membership oracles. In *International Conference on Algorithmic Learning Theory*, pp. 1055–1073. PMLR, 2023.

Zakaria Mhammedi. Efficient projection-free online convex optimization with membership oracle. In *Conference on Learning Theory*, pp. 5314–5390. PMLR, 2022.

Benjamin Kurt Miller, Ricky TQ Chen, Anuroop Sriram, and Brandon M Wood. Flowmm: Generating materials with riemannian flow matching. *arXiv preprint arXiv:2406.04713*, 2024.

Saeid Naderiparizi, Xiaoxuan Liang, Berend Zwartsenberg, and Frank Wood. Constrained generative modeling with manually bridged diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19607–19615, 2025.

Sai Shankar Narasimhan, Shubhankar Agarwal, Litu Rout, Sanjay Shakkottai, and Sandeep P Chinchali. Constrained posterior sampling: Time series generation with hard constraints. *arXiv preprint arXiv:2410.12652*, 2024.

Rahul Nellikkath and Spyros Chatzivasileiadis. Physics-informed neural networks for minimising worst-case violations in dc optimal power flow. In *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 419–424. IEEE, 2021.

Xiang Pan, Tianyu Zhao, Minghua Chen, and Shengyu Zhang. Deepopf: A deep neural network approach for security-constrained dc optimal power flow. *IEEE Transactions on Power Systems*, 36(3):1725–1735, 2020.

Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H Low. Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems. *IEEE Systems Journal*, pp. 42–47, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Saša V Raković and Mircea Lazar. The minkowski–lyapunov equation for linear dynamics: Theoretical foundations. *Automatica*, 50(8):2015–2024, 2014.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

R Tyrrell Rockafellar. Second-order convex analysis. *J. Nonlinear Convex Anal*, 1(1-16):84, 1999.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Daniel Tabas and Baosen Zhang. Computationally efficient safe reinforcement learning for power systems. In *2022 American Control Conference (ACC)*, pp. 3303–3310. IEEE, 2022a.

Daniel Tabas and Baosen Zhang. Safe and efficient model predictive control using neural networks: An interior point approach. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 1142–1147. IEEE, 2022b.

Mathieu Tanneau and Pascal Van Hentenryck. Dual lagrangian learning for conic optimization. *arXiv preprint arXiv:2402.03086*, 2024.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pp. 1–34, 2024.

Jesus Tordesillas, Jonathan P How, and Marco Hutter. Rayen: Imposition of hard convex constraints on neural networks. *arXiv preprint arXiv:2307.08336*, 2023.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

Robert A Van de Geijn. *Using PLAPACK–parallel Linear Algebra Package*. MIT press, 1997.

Zhengchao Wan, Qingsong Wang, Gal Mishne, and Yusu Wang. Elucidating flow matching ode dynamics with respect to data geometries. *arXiv e-prints*, pp. arXiv–2412, 2024.

Runzhong Wang, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang, and Junchi Yan. Linsatnet: the positive linear satisfiability neural networks. In *International Conference on Machine Learning*, pp. 36605–36625. PMLR, 2023.

Tianyu Xie, Yu Zhu, Longlin Yu, Tong Yang, Ziheng Cheng, Shiyue Zhang, Xiangyu Zhang, and Cheng Zhang. Reflected flow matching. *arXiv preprint arXiv:2405.16577*, 2024.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Olga Zaghen, Floor Eijkelboom, Alison Pouplin, and Erik J Bekkers. Towards variational flow matching on general geometries. *arXiv preprint arXiv:2502.12981*, 2025.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, pp. 1–3, 2025.

Liyuan Zheng, Yuanyuan Shi, Lillian J Ratliff, and Baosen Zhang. Safe reinforcement learning of control-affine systems with vertex networks. In *Learning for Dynamics and Control*, pp. 336–347. PMLR, 2021.

Kaizhen Zhu, Mokai Pan, Yuexin Ma, Yanwei Fu, Jingyi Yu, Jingya Wang, and Ye Shi. Unidb: A unified diffusion bridge framework via stochastic optimal control. *arXiv preprint arXiv:2502.05749*, 2025.

Contents

LLM USAGE

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript.

A RELATED WORK ON NN FEASIBILITY

Research on ensuring neural network feasibility can be categorized into several approaches:

Basic Constraint Handling: Specialized activation functions (Sigmoid/Softmax) address basic constraints such as box or simplex (Pan et al., 2022; Donti et al., 2020). Penalty of output constraint violations can be incorporated into loss functions to improve NN feasibility (Cheng et al., 2019; Pan et al., 2020; Nellikkath & Chatzivasileiadis, 2021).

Strict Satisfaction Methods: For exact equality constraint satisfaction, prediction-then-reconstruct or completion techniques can be applied (Donti et al., 2020; Pan et al., 2022; Liang et al., 2023). For more general constraint enforcement, orthogonal/L2 projection is often employed. However, solving the projection problem either by iterative solver or equivalent optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019; Chen et al., 2021; Wang et al., 2023) is computationally intensive in real-time. More efficient biseciton-based projection can also be applied at the cost of a minor optimality loss (Liang et al., 2024; Liang & Chen, 2025).

Sampling Approach: To guarantee feasibility, an inner approximation of the original constraint set can be constructed. For linear constraints, vertex networks employ a convex combination of sampled vertexes and rays to ensure policy feasibility (Frerix et al., 2020; Zheng et al., 2021). For general compact but fixed constraint sets, probabilistic transformer utilizes feasible samples to ensure feasibility (Kratsios et al., 2021). However, scalability remains a challenge due to the exponential growth in required samples with increasing problem dimensionality.

Gauge function. These works utilize gauge functions (Blanchini & Miani, 2008) to constrain the NN. A closed-form bijection, known as gauge mapping, between a hypercube and a polytope is used to bound the NN output within the polytope (Tabas & Zhang, 2022a;b; Li et al., 2023a). For fixed convex constraints, a series of works apply gauge functions to find feasible boundary solutions (Tordesillas et al., 2023; Konstantinov & Utkin, 2023; Li & Mohammadi, 2023; Tanneau & Van Hentenryck, 2024).

These approaches are typically designed for the end-to-end neural network structure, which can not be directly applied to the diffusion or flow-matching based generative model due to the forward integration calculation.

B GAUGE MAPPING OVER GENERAL CONVEX SETS

A general compact convex set encompasses both linear equality and convex inequality constraints.

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid Ax = b,\ g_1(x) \leq 0, \cdots, g_m(x) \leq 0\}, \tag{6}$$

where $A \in \mathbb{R}^{r \times n}, b \in \mathbb{R}^r$, and $g_1(x), \ldots, g_m(x)$ are convex functions.

This section presents a systematic approach to handling such sets by first eliminating linear equality constraints, followed by computing gauge mappings for the remaining inequality constraints.

B.1 HANDLING LINEAR EQUALITY CONSTRAINTS

Without loss of generality, assuming $\mathrm{rank}(A) = r$, we partition the decision variable x into $x_1 \in \mathbb{R}^{n-r}$ and $x_2 \in \mathbb{R}^r$. Accordingly, we partition matrix A into $A = [A_1, A_2]$, where $A_1 \in \mathbb{R}^{r \times (n-r)}$ and $A_2 \in \mathbb{R}^{r \times r}$. The equality constraint $Ax = b$ can then be written as:

$$A_1 x_1 + A_2 x_2 = b \tag{7}$$

By choosing the partition such that A_2 has full rank r, we can express x_2 explicitly in terms of x_1:

$$x_2 = \phi(x_1) = A_2^{-1}(b - A_1 x_1) \tag{8}$$

This transformation reduces the original set to one with only inequality constraints:

$$\mathcal{C}^s = \{x_1 \in \mathbb{R}^{n-r} \mid g([x_1, \phi(x_1)]) \leq 0\} \tag{9}$$

Therefore, we only consider the inequality constraints in the main body of this work.

B.2 Gauge Mapping for Inequality Constraints

Without loss of generality, we consider $\mathcal{C} = \{x \mid g_1(x) \leq 0, \ldots, g_m(x) \leq 0\}$.

We define the following metrics for a compact convex set.

Definition B.1 (Point-to-boundary distance and its inverse (Tordesillas et al., 2023))**.** Let $\mathcal{C} \subset \mathbb{R}^n$ be a compact convex set and $x^\circ \in \text{int}(\mathcal{C})$ an interior point. For any unit vector $v \in \mathbb{S}^{n-1} = \{u \in \mathbb{R}^n \mid \|u\| = 1\}$, we define the interior-point-to-boundary distance function $d_\mathcal{C} : \text{int}(\mathcal{C}) \times \mathbb{S}^{n-1} \to \mathbb{R}_+$ along direction v as

$$d_\mathcal{C}(x^\circ, v) = \sup\{\lambda \geq 0 \mid x^\circ + \lambda v \in \mathcal{C}\}. \tag{10}$$

The inverse distance function $\kappa_\mathcal{C} : \text{int}(\mathcal{C}) \times \mathbb{S}^{n-1} \to \mathbb{R}_+$ is defined as $\kappa_\mathcal{C}(x^\circ, v) := 1/d_\mathcal{C}(x^\circ, v)$.

For a compact convex set, the minimum and maximum interior point-to-boundary distances are bounded and are defined as:

- the minimum point-to-boundary distance: $r_\text{i} = \arg\sup_{r \geq 0}\{\mathcal{B}_2(x^\circ, r) \subseteq \mathcal{C}\}$.
- the maximum point-to-boundary distance: $r_\text{o} = \arg\inf_{r \geq 0}\{\mathcal{C} \subseteq \mathcal{B}_2(x^\circ, r)\}$.
- thus, we have $\mathcal{B}_2(x^\circ, r_\text{i}) \subseteq \mathcal{C} \subseteq \mathcal{B}_2(x^\circ, r_\text{o})$

B.2.1 Gauge Function and Properties

We then introduce the gauge function:

Definition B.2 (Gauge/Minkowski function (Blanchini & Miani, 2008))**.** Let $\mathcal{C} \subset \mathbb{R}^n$ be a compact convex set with a non-empty interior. The Gauge/Minkowski function $\gamma_\mathcal{C} : \mathbb{R}^n \times \text{int}(\mathcal{C}) \to \mathbb{R}_+$ is defined as

$$\gamma_\mathcal{C}(x, x^\circ) = \inf\{\lambda \geq 0 \mid x \in \lambda(\mathcal{C} - x^\circ)\}, \tag{11}$$

where $x^\circ \in \text{int}(\mathcal{C})$ is an interior point of $\mathcal{C}$.

The Gauge function generalizes the concept of a norm. For a set $\mathcal{C}$ that is symmetric about the origin, the gauge function $\gamma_\mathcal{C}(x, 0)$ defines a norm. In particular, when $\mathcal{C} = \mathcal{B}_p = \{x \in \mathbb{R}^n \mid \|x\|_p \leq 1\}$ is the unit ball of the p-norm, we have $\gamma_{\mathcal{B}_p}(x, 0) = \|x\|_p$. More generally, the gauge function satisfies the following properties for all $x, y \in \mathbb{R}^n$ and $\alpha \geq 0$:

Lemma 1 (Properties of gauge function)**.** *The gauge function $\gamma_\mathcal{C}(x, x^\circ)$ with respect to a compact convex set $\mathcal{C}$ and an interior point $x^\circ \in \text{int}(\mathcal{C})$ satisfies the following properties:*

- *Non-negativity: $\gamma_\mathcal{C}(x, x^\circ) \geq 0$*
- *Positive homogeneity: $\gamma_\mathcal{C}(\alpha x, x^\circ) = \alpha\gamma_\mathcal{C}(x, x^\circ)$ for $\alpha \geq 0$.*
- *Subadditivity: $\gamma_\mathcal{C}(x + y, x^\circ) \leq \gamma_\mathcal{C}(x, x^\circ) + \gamma_\mathcal{C}(y, x^\circ)$.*
- *Convexity: induced by positive homogeneity and subadditivity.*
- *Differentiability: under convexity, the gauge function is twice differentiable almost everywhere by Alexandrov's theorem (Rockafellar, 1999).*
- *Equivalent formulations based on (inverse) distance function:*

$$\gamma_\mathcal{C}(x, x^\circ) = \kappa_\mathcal{C}(x^\circ, x/\|x\|) \cdot \|x\| = \frac{\|x\|}{d_\mathcal{C}(x^\circ, x/\|x\|)} \tag{12}$$

$$\gamma_\mathcal{C}(x/\|x\|, x^\circ) = \kappa_\mathcal{C}(x^\circ, x/\|x\|) = \frac{1}{d_\mathcal{C}(x^\circ, x/\|x\|)} \tag{13}$$

- *Upper/lower bounds: the gauge function is bounded as:* $\gamma_{\mathcal{C}}(x, x^\circ) \in [\|x\|/r_{\mathrm{o}}, \|x\|/r_{\mathrm{i}}]$
- *Lipschitz: the gauge function has Lipschitz constant as* $\frac{\|\gamma_{\mathcal{C}}(x,x^\circ)-\gamma_{\mathcal{C}}(y,x^\circ)}{\|x-y\|}\| \leq \frac{1}{r_{\mathrm{i}}}$

Proof. The non-negativity, positive homogeneity, and subadditivity are provided in (Blanchini & Miani, 2008). Convexity and differentiability are naturally implied by those properties. The equivalent formulation is derived from their definitions.

The upper and lower bounds are derived as:

$$\gamma_{\mathcal{C}}(x, x^\circ) = \frac{x}{d_{\mathcal{C}}(x^\circ, x/\|x\|)} \in [\|x\|/r_{\mathrm{o}}, \|x\|/r_{\mathrm{i}}] \tag{14}$$

The Lipschitz constant of the gauge function is derived as:

$$\gamma_{\mathcal{C}}(x, x^\circ) \leq \gamma_{\mathcal{C}}(x-y, x^\circ) + \gamma_{\mathcal{C}}(y, x^\circ) \tag{15}$$

$$\gamma_{\mathcal{C}}(x, x^\circ) - \gamma_{\mathcal{C}}(y, x^\circ) \leq \gamma_{\mathcal{C}}(x-y, x^\circ) \leq \|x-y\|/r_{\mathrm{i}} \tag{16}$$

where the inequality (15) is derived by the positive homogeneity. Similarly, we have:

$$\gamma_{\mathcal{C}}(y, x^\circ) - \gamma_{\mathcal{C}}(x, x^\circ) \leq \gamma_{\mathcal{C}}(y-x, x^\circ) \leq \|x-y\|/r_{\mathrm{i}} \tag{17}$$

Combining two inequalities, we have the Lipschitz as:

$$\frac{\|\gamma_{\mathcal{C}}(x, x^\circ) - \gamma_{\mathcal{C}}(y, x^\circ)}{\|x-y\|}\| \leq \frac{1}{r_{\mathrm{i}}} \tag{18}$$

□

B.2.2 Gauge Mapping and Properties

Based on the gauge function, we can construct a bijective mapping between any pair of compact convex sets as:

Definition B.3 (Gauge Mapping)**.** Let $\mathcal{Z}, \mathcal{X} \subset \mathbb{R}^n$ be compact convex sets with interior points $z^\circ \in \mathrm{int}(\mathcal{Z})$ and $x^\circ \in \mathrm{int}(\mathcal{X})$, respectively.

The gauge mapping $\Phi : \mathcal{Z} \to \mathcal{X}$ is defined as:

$$\Phi(z) = \frac{\gamma_{\mathcal{Z}}(z - z^\circ, z^\circ)}{\gamma_{\mathcal{X}}(z - z^\circ, x^\circ)}(z - z^\circ) + x^\circ,\ z \in \mathcal{Z} \tag{19}$$

The inverse mapping $\Phi^{-1} : \mathcal{X} \to \mathcal{Z}$ is given by:

$$\Phi^{-1}(x) = \frac{\gamma_{\mathcal{X}}(x - x^\circ, x^\circ)}{\gamma_{\mathcal{Z}}(x - x^\circ, z^\circ)}(x - x^\circ) + z^\circ,\ x \in \mathcal{X} \tag{20}$$

In essence, the gauge mapping scales the boundary of a convex set from an interior point to another convex set and translates it to its interior point. When $\mathcal{Z}$ is a unit p-norm ball, the gauge mapping is simplified in Def. 4.1 as:

$$\Phi(z) = \frac{\|z\|_p}{\gamma_{\mathcal{C}}(z, x^\circ)}z + x^\circ,\ \forall z \in \mathcal{B}, \quad \Phi^{-1}(x) = \frac{\gamma_{\mathcal{C}}(x - x^\circ, x^\circ)}{\|x - x^\circ\|_p}(x - x^\circ),\ \forall x \in \mathcal{C}, \tag{21}$$

Further, the gauge mapping between $\mathcal{B}_2$ and $\mathcal{C}$ can be simplified as:

$$\Phi(z) = d_{\mathcal{C}}(x^\circ, z/\|z\|) \cdot z + x^\circ,\ \forall z \in \mathcal{B}, \quad \Phi^{-1}(x) = \frac{x - x^\circ}{d_{\mathcal{C}}(x^\circ, x - x^\circ/\|x - x^\circ\|)},\ \forall x \in \mathcal{C} \tag{22}$$

Proposition B.1 (Properties of gauge mapping)**.** *The gauge mapping between any pair of compact convex sets satisfies the following properties:*

- *it is invertible.*
- *it is continuous everywhere and twice differentiable almost everywhere in both directions.*

- *it is a bi-Lipschitz homeomorphism.*

Proof. First, the invertibility can be easily verified by:

$$\begin{aligned}\Phi(\Phi^{-1}(x)) &= \Phi(\frac{\gamma_{\mathcal{X}}(x-x^\circ, x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ, z^\circ)}(x-x^\circ)+z^\circ) && (23)\\ &= \frac{\gamma_{\mathcal{Z}}(\frac{\gamma_{\mathcal{X}}(x-x^\circ,x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ,z^\circ)}(x-x^\circ), z^\circ)}{\gamma_{\mathcal{X}}(\frac{\gamma_{\mathcal{X}}(x-x^\circ,x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ,z^\circ)}(x-x^\circ), x^\circ)}\frac{\gamma_{\mathcal{X}}(x-x^\circ, x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ, z^\circ)}(x-x^\circ)+x^\circ && (24)\\ &= \frac{\frac{\gamma_{\mathcal{X}}(x-x^\circ,x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ,z^\circ)}\gamma_{\mathcal{Z}}((x-x^\circ), z^\circ)}{\frac{\gamma_{\mathcal{X}}(x-x^\circ,x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ,z^\circ)}\gamma_{\mathcal{X}}((x-x^\circ), x^\circ)}\frac{\gamma_{\mathcal{X}}(x-x^\circ, x^\circ)}{\gamma_{\mathcal{Z}}(x-x^\circ, z^\circ)}(x-x^\circ)+x^\circ && (25)\\ &= (x-x^\circ)+x^\circ = x && (26)\end{aligned}$$

where Eq. (25) is derived by the positive homogeneity of the gauge function.

Second, the continuity and differentiability are derived from the properties of the gauge function and elemental compositions in the gauge mapping.

Third, the homeomorphism is derived from the continuity in both directions and the invertibility.

Proof of Bi-Lipschitz Properties of Gauge Mapping in Prop. 4.1

To derive the bi-Lipschitz properties, we first consider the gauge mapping between $\mathcal{B}_2$ and $\mathcal{C}$ and derive its Lipschitz constants shown in Prop. 4.1 as follows,

$$\Phi(z) = \frac{\|z\|_2}{\gamma_{\mathcal{C}}(z, x^\circ)}\, z + x^\circ,\ \forall z \in \mathcal{B}_2 \tag{27}$$

Differentiating $\Phi(z)$ with respect to z (using the product and quotient rules) yields

$$\frac{\partial \Phi}{\partial z} = \frac{\|z\|_2}{\gamma_{\mathcal{C}}(z, x^\circ)}\, I + \frac{zz^\top}{\gamma_{\mathcal{C}}(z, x^\circ)\|z\|_2} - \frac{\|z\|_2}{\gamma_{\mathcal{C}}(z, x^\circ)^2}\, z\Big(\nabla_z \gamma_{\mathcal{C}}(z, x^\circ)\Big)^\top. \tag{28}$$

Taking the operator norm and applying the triangle inequality gives

$$\begin{aligned}\Big\|\frac{\partial \Phi}{\partial z}\Big\| &\leq \frac{\|z\|_2}{\gamma_{\mathcal{C}}(z, x^\circ)} + \frac{\|z\|_2}{\gamma_{\mathcal{C}}(z, x^\circ)} + \frac{\|z\|_2^2}{\gamma_{\mathcal{C}}(z, x^\circ)^2}\Big\|\nabla_z \gamma_{\mathcal{C}}(z, x^\circ)\Big\| \\ &\leq r_{\mathrm{o}} + r_{\mathrm{o}} + \frac{r_{\mathrm{o}}^2}{r_{\mathrm{i}}},\end{aligned} \tag{29}$$

where in the last inequality we have used the facts that (i) for $z \in \mathcal{B}_2$ one has $\|z\|_2 \leq 1$, (ii) the gauge function satisfies $\gamma_{\mathcal{C}}(z, x^\circ) \in [\|z\|/r_{\mathrm{o}}, \|z\|/r_{\mathrm{i}}]$, and (iii) $\|\nabla_z \gamma_{\mathcal{C}}(z, x^\circ)\|$ is bounded by $1/r_{\mathrm{i}}$. In summary, we obtain

$$\Big\|\frac{\partial \Phi}{\partial z}\Big\| \leq 2r_{\mathrm{o}} + \frac{r_{\mathrm{o}}^2}{r_{\mathrm{i}}}, \tag{30}$$

which proves that the forward Lipschitz constant of Φ satisfies

$$\text{Forward Lipschitz:}\quad \mathrm{L}_\Phi \leq 2r_{\mathrm{o}} + \frac{r_{\mathrm{o}}^2}{r_{\mathrm{i}}}. \tag{31}$$

Next, consider the inverse gauge mapping from $\mathcal{C}$ to the 2-norm ball as

$$\Phi^{-1}(x) = \frac{\gamma_{\mathcal{C}}(x-x^\circ, x^\circ)}{\|x-x^\circ\|_2}\,(x-x^\circ),\ \forall x \in \mathcal{C} \tag{32}$$

Differentiating with respect to x gives

$$\frac{\partial \Phi^{-1}}{\partial x} = \nabla \gamma_{\mathcal{C}}(x-x^\circ, x^\circ)\left(\frac{x-x^\circ}{\|x-x^\circ\|_2}\right)^\top + \gamma_{\mathcal{C}}(x-x^\circ, x^\circ)\cdot\frac{I - \frac{(x-x^\circ)(x-x^\circ)^\top}{\|x-x^\circ\|_2^2}}{\|x-x^\circ\|_2}. \tag{33}$$

Taking norms and again using the triangle inequality leads to

$$\left\|\frac{\partial \Phi^{-1}}{\partial x}\right\| \leq \left\|\nabla \gamma_{\mathcal{C}}(x-x^{\circ}, x^{\circ})\left(\frac{x-x^{\circ}}{\|x-x^{\circ}\|_2}\right)^{\top}\right\| + \gamma_{\mathcal{C}}(x-x^{\circ}, x^{\circ})\left\|\frac{I-\frac{(x-x^{\circ})(x-x^{\circ})^{\top}}{\|x-x^{\circ}\|_2^2}}{\|x-x^{\circ}\|_2}\right\|. \quad (34)$$

Using the bound $\gamma_{\mathcal{C}}(x-x^{\circ}, x^{\circ}) \in [\|x-x^{\circ}\|/r_{\mathrm{o}}, \|x-x^{\circ}\|/r_{\mathrm{i}}]$ and the projection term has norm at most $1/\|x-x^{\circ}\|_2$), we have

$$\left\|\frac{\partial \Phi^{-1}}{\partial x}\right\| \leq \frac{1}{r_{\mathrm{i}}} + \frac{1}{r_{\mathrm{i}}} = \frac{2}{r_{\mathrm{i}}}. \quad (35)$$

Thus, the inverse Lipschitz constant is bounded by

$$\text{Inverse Lipschitz:} \quad \mathrm{L}_{\Phi^{-1}} \leq \frac{2}{r_{\mathrm{i}}}. \quad (36)$$

This completes the proof of Prop. 4.1.

To extend the Bi-Lipschitz properties to any pair of compact convex sets, e.g., $\mathcal{Z}$ and $\mathcal{X}$, we decompose the gauge mapping as

$$\mathcal{X} = \Phi(\mathcal{Z}) = \Phi_1(\mathcal{B}_2) = \Phi_1(\Phi_2^{-1}(\mathcal{Z})) \quad (37)$$

where $\mathcal{Z} = \Phi_1(\mathcal{B}_2)$ and $\mathcal{X} = \Phi_2(\mathcal{B}_2)$.

Leveraging the inequality of Lipschitz for mapping compositions as $\mathrm{L}_{f_1 \circ f_2} \leq \mathrm{L}_{f_1}\mathrm{L}_{f_2}$, we can conclude that the gauge mapping between any pair of convex sets is bi-Lipschitz.

□

B.2.3 Computation Methods

IP computation

In practice, we seek such a "central" interior point by solving the following residual minimization problem through convex optimization in the offline phase (Tordesillas et al., 2023):

$$\min_{x^{\circ}} \eta \quad (38)$$

$$\text{s.t. } g_i(x^{\circ}) \leq \eta \quad i = 1, \cdots, m \quad (39)$$

We note that solving this convex optimization problem with a linear objective incurs only polynomial time complexity. As this computation is performed offline prior to model training, it adds negligible overhead to the overall computational cost.

Gauge function computation

Computation of gauge mapping essentially involves calculating the gauge function or (inverse) distance function. Following the established closed-form distance function calculation for several common convex sets in (Tordesillas et al., 2023), we give a summary in Table 8. It provides closed-form expressions for the inverse distance function across various constraint types. Most matrix calculations can be computed and stored offline before being applied for online inference.

When the inverse distance function lacks an explicit expression, we employ an efficient bisection algorithm detailed in algorithm 3. This algorithm supports batch processing, enabling efficient parallel computation for multiple inputs simultaneously.

Table 8: Closed-form Expressions for Inverse Distance Functions (Tordesillas et al., 2023)

Constraints	Formulation	Inverse Distance Function
Intersections	$\{g_1(x) \leq 0, \cdots, g_m(x) \leq 0\}$	$\kappa_g(x^\circ, v) = \max_{1 \leq i \leq m} \{\kappa_{g_i}(x^\circ, v)\}$
Linear	$g_L(x) = a^\top x - b \leq 0$	$\kappa_{g_L}(x^\circ, v) = \{\frac{a^\top v}{b - a^\top x^\circ}\}^+$
Quadratic	$g_Q(x) = x^\top Q x + a^\top x - b \leq 0$	$\kappa_{g_Q}(x^\circ, v) = \{1/\text{root}(A_Q, B_Q, C_Q)\}^+$
Second Order Cone	$g_S(x) = \lVert A^\top x + p \rVert_2 - (a^\top x + b) \leq 0$	$\kappa_{g_S}(x^\circ, v) = \{1/\text{root}(A_S, B_S, C_S)\}^+$
Matrix Cone	$g_M(x) = \sum_{i=1}^n x_i \cdot F_i + F_0 \succeq 0$	$\kappa_{g_M}(x^\circ, v) = \max\{\text{eig}(L^\top(-S)L)\}^+$

[1] Notation: $x, a \in \mathbb{R}^n, b \in \mathbb{R}, Q \in \mathbb{S}_+^n, A \in \mathbb{R}^{n \times k}, p \in \mathbb{R}^k, F_0, \cdots, F_n \in \mathbb{R}^{k \times k}$.

[2] $A_Q = v^\top Q v,\ B_Q = 2x^{\circ\top} Q v + a^\top v,\ C_Q = x^{\circ\top} Q x^\circ + a^\top x^\circ - b$.

[3] $A_S = (A^\top v)^\top (A^\top v) - (a^\top v)^2,\ B_S = 2(A^\top x^\circ + p)^\top (A^\top v) - 2(a^\top x^\circ + b)(a^\top v),\ C_S = (A^\top x^\circ + p)^\top (A^\top x^\circ + p) - (a^\top x^\circ + b)^2$.

[4] $H = F_0 + \sum_{i=1}^n x_i^\circ F_i\ , H^{-1} = L^\top L\ , S = \sum_{i=1}^n v_i F_i$.

[5] $(\cdot)^+ = \max(\cdot, 0)$.

[6] $\text{root}(x_1, x_2, x_3) = \frac{-x_2 \pm \sqrt{x_2^2 - 4x_1 x_3}}{2x_1}$ denotes the quadratic equation solution

[7] $\text{eig}(X) = \lambda_1, \cdots, \lambda_n$ denotes the eigenvalues satisfying $\det(X - \lambda I) = 0$. Note that only the maximum eigenvalue is needed. Thus, power iteration methods can be applied to compute it efficiently.

[8] Note that all v-independent terms can be computed only once and stored for use.

Algorithm 3 Bisection Algorithm for Point-to-Boundary Distance

```
Input: A compact convex set $\mathcal{C}$, an interior point $x^\circ \in \text{int}(\mathcal{C})$, and a unit vector $v$.
Initialize: $\alpha_l = 0$ and $\alpha_u = 1$
while $|\alpha_l - \alpha_u| \geq \epsilon$ do
    if $x^\circ + \alpha_u \cdot v \in \mathcal{C}$ then
        increase lower bound: $\alpha_l \leftarrow \alpha_u$
        double upper bound: $\alpha_u \leftarrow 2 \cdot \alpha_m$
    else
        bisection: $\alpha_m = (\alpha_l + \alpha_u)/2$
        if $x^\circ + \alpha_m \cdot v \in \mathcal{C}$ then
            increase lower bound: $\alpha_l \leftarrow \alpha_m$
        else
            decrease upper bound: $\alpha_u \leftarrow \alpha_m$
        end if
    end if
end while
Output: $d_{\mathcal{C}}(x^\circ, v) \approx \alpha_m$
```

C Gauge Mapping for Non-convex Constraints

In this section, we extend the conventional gauge mapping over convex sets in Euclidean space to certain non-convex settings, including star-convex sets and geodesic-convex sets.

C.1 Star-convex Set

Definition C.1 (Star-convex Set)**.** A set $\mathcal{S} \subset \mathbb{R}^n$ is said to be *star-convex* with respect to an interior point $x^\circ \in \text{int}(\mathcal{S})$ if for every $x \in \mathcal{S}$, the line segment connecting x° and x lies entirely within $\mathcal{S}$, i.e., $x^\circ + \theta(x - x^\circ) \in \mathcal{S}$ for all $\theta \in [0, 1]$.

The gauge mapping framework can be extended beyond convex sets to certain non-convex domains, particularly star-convex sets. This extension significantly broadens the applicability of our GFM approach to more complex geometric constraints encountered in practical applications. A star-convex set $\mathcal{S}$ is characterized by the existence of an interior point x° such that any line segment connecting x° to any point in the set remains entirely within the set. While star-convex sets lack the full convexity property, they retain a critical radial structure that allows gauge mappings to be constructed in a similar manner.

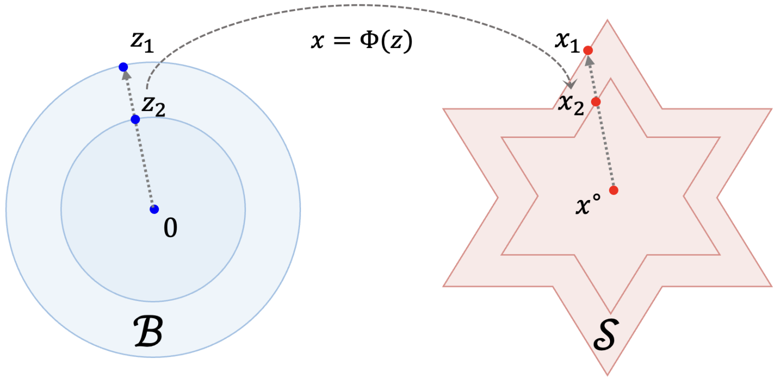


Figure 9: Gauge mapping construction for star-convex set.

For a star-convex set $\mathcal{S}$ with interior point x°, we can define the gauge function as:

$$\gamma_{\mathcal{S}}(x, x^\circ) = \inf\{\lambda \geq 0 \mid x \in \lambda(\mathcal{S} - x^\circ)\} \tag{40}$$

This formulation captures the minimum scaling factor needed to reach point x when scaling along the ray from x°. The corresponding gauge mapping $\Phi : \mathcal{B} \to \mathcal{S}$ between a unit p-norm ball and the star-convex set follows the same construction as in Definition 4.1. As illustrated in Figure 9, the gauge mapping preserves the radial structure of the star-convex set while establishing a homeomorphism with the unit ball. This property is particularly valuable for handling constraints with non-convex geometries that still maintain visibility from an interior point, such as ℓ_p-norm balls with $p < 1$ or certain non-convex polytopes arising in practical applications.

The computation of gauge functions for star-convex sets follows similar principles as for convex sets, though additional care may be needed:

- For analytically defined star-convex sets (e.g., ℓ_p-norm balls with $p < 1$), closed-form expressions for the gauge function can often be derived as Def. 4.1.
- For star-convex sets defined by the union of convex components or by non-convex inequalities, bisection methods remain applicable for computing the gauge function, though they may require specialized boundary evaluation techniques (Liang & Chen, 2025).
- Selection of the interior point x° becomes more critical for star-convex sets, as it determines the visibility region and thus the quality of the mapping. When multiple interior points are viable, selecting one that maximizes the minimum distance to the boundary often yields better numerical properties.

C.2 Geodesic-convex Set

When the underlying space is a Riemannian manifold rather than a Euclidean space, the notion of straight lines is replaced by geodesics. In this context, one speaks of geodesic convexity.

Definition C.2 (Geodesic-Convex Set)**.** Let (M, g) be a Riemannian manifold, and let $\mathcal{U} \subset M$ be a subset. $\mathcal{U}$ is *geodesically convex* if, for every pair of points $x, y \in \mathcal{U}$, there exists a unique minimizing geodesic $\eta : [0, 1] \to M$ with respect to g that satisfies $\eta(0) = x$, $\eta(1) = y$, $\eta(t) \in \mathcal{U}$, $\forall t \in [0, 1]$.

To extend the gauge mapping on a geodesic-convex set over a manifold, we replace the linear structure with the exponential map. Since $\mathcal{U} \subset M$ is geodesic-convex, the exponential map at any interior point is a diffeomorphism onto its image (Lee, 2006). Given this property, any point $x \in \mathcal{U}$ with $x \neq x^\circ$ can be uniquely represented in *geodesic polar coordinates* as $x = \exp_{x^\circ}(r_x\, v_x)$, where $x^\circ \in \mathcal{U}$ is an interior point of $\mathcal{U}$, $v_x \in \mathbb{S}^{n-1} \subset T_{x^\circ}M$ is a unit vector over the tangent space $T_{x^\circ}M$, and $r_x = d_g(x, x^\circ)$ is the geodesic distance from x° to x.

Similarly, we can establish the *geodesic gauge mapping* that transforms a unit ball from the tangent space to the geodesic-convex set on the manifold. As illustrated in Figure 10, this mapping establishes a correspondence between the unit ball in the tangent space and the geodesic-convex set on the

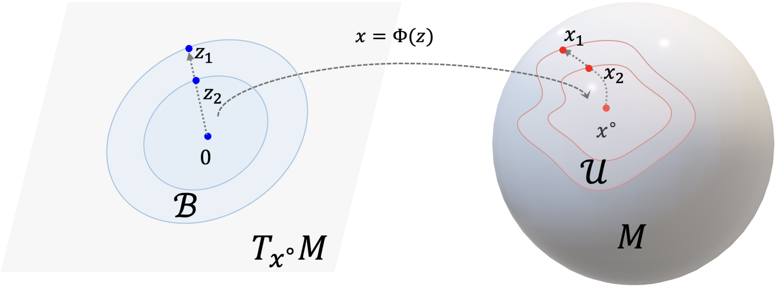


Figure 10: Gauge mapping construction for geodesic-convex set.

manifold. The construction respects the intrinsic geometry of the manifold while preserving the radial structure that is characteristic of gauge mappings.

Definition C.3 (Geodesic Gauge Mapping)**.** Without loss of generality, let $\mathcal{B} \triangleq \{z \in T_{x^\circ}M \mid \|z\|_2 \leq 1\}$ denote the unit ball in the tangent space at x°. The *geodesic gauge mapping* $\Phi : \mathcal{B} \to \mathcal{U}$ and its inverse are defined as

$$\Phi(z) = \exp_{x^\circ}(d_{\mathcal{U}}(x^\circ, z/\|z\|) \cdot z), \quad \forall z \in \mathcal{B} \tag{41}$$

$$\Phi^{-1}(x) = r_x / d_{\mathcal{U}}(x^\circ, v_x) \cdot v_x, \quad \forall x = \exp_{x^\circ}(r_x v_x) \in \mathcal{U} \tag{42}$$

where the point-to-boundary geodesic distance $d_{\mathcal{U}}(x^\circ, v_x)$ along v_x is defined as $d_{\mathcal{U}}(v_x) \triangleq \sup\{\lambda \geq 0 \mid \exp_{x^\circ}(\lambda \cdot v_x) \in \mathcal{U}\}$.

Note that the point-to-boundary geodesic distance may not admit a closed-form expression; we can apply the bisection methods similar to Alg. 3 to locate the boundary points starting from an interior point.

This extension enables our GFM framework to handle constraints on manifolds, expanding its application to domains such as inverse problems with manifold constraints and atmospheric modeling over spherical geometries.

C.3 More General Non-convex Set

When applying our methods to more general non-convex sets, a natural choice is the class of non-convex sets that are homeomorphic to a unit ball. However, such *ball-homeomorphic* sets do not generally admit a closed-form homeomorphism. One promising approach is to apply an invertible neural network to learn the target homeomorphism, which offers both feasibility and approximation guarantees (Liang et al., 2023; 2024). For more general non-convex constraints, low-complexity constrained generation schemes remain largely unexplored, particularly for NP-hard problems where even identifying an interior point poses significant computational challenges, such as non-convex quadratic equations or mixed-integer formulations. The exploration of more general and theoretically guaranteed approaches constitutes a promising direction for future research.

D Theoretical Analysis of GFM

D.1 Assumption and Preliminary

We made the following assumptions for the error analysis, which are common for error analysis for the flow/diffusion-based generative models (Kwon et al., 2022; Benton et al., 2023; Xie et al., 2024).

Assumption 1 (Regularity of Distribution (Wan et al., 2024))**.** We assume the data distribution p satisfies the following regularity conditions:

- It has finite 2-moment, i.e., $\int_{\mathbb{R}^n} \|x\|^2 p(\mathrm{d}x) < \infty$;
- Its support, $\Omega = \mathrm{supp}(p)$, has a positive reach;
- It has a non-vanishing density, i.e., there exist constants $k \geq 0$ and $c > 0$ such that for any radius $R > 0$, there exists a constant $C_R > 0$ where for any small radius $0 \leq r < c$ and any $x \in B_R(0) \cap \Omega$, we have $p(B_r(x)) \geq C_R r^k$.

The finite 2-moment requirement is standard in the literature for analyzing generative models. The positive reach condition is satisfied in our setting since we consider a convex support set $\Omega = \mathcal{C}$. The last requirement ensures the density does not vanish within the support, which prevents pathological cases where probability mass concentrates on lower-dimensional manifolds.

There are also other (more restrictive) regularity conditions for the data distribution in the literature, including the covariance condition (Benton et al., 2023) and κ-semi-log-concave/convex (Gao et al., 2024a), and convex support set (Gao et al., 2024b). It remains largely open for the minimal necessary conditions to establish the well-posedness of flow matching models.

Assumption 2 (Neural Network Approximation Properties)**.** Given a distribution p with support Ω, let $u(x,t)$ be the designed target vector field and v_θ be the neural network approximated vector field in the flow-matching model.

- v_θ is L_θ-Lipschitz for $x \in \Omega$ and $t \in [0,1]$;
- The ℓ_2 approximation error is bounded as $\epsilon_\theta^2 = \mathbb{E}_{x_t,t}\|v_\theta(x_t,t) - u(x_t,t)\|^2$, where p_t is the probability density driven by the target vector field u.

The Lipschitz constant of a trained neural network is bounded within a compact set in our setting $\Omega = \mathcal{C}$. The training loss in Eq. (1) is equivalent to the vector field approximation error up to a constant (Lipman et al., 2022). Therefore, we can minimize the approximation error through proper training of the loss function.

Lemma 2 (Error Bound for Flow Matching (Benton et al., 2023))**.** *For the vanilla flow matching model: $x_1 = x_0 + \int_0^1 v_\theta(x,t)\,\mathrm{d}t$, with induced probability distribution p_θ at $t = 1$. The Wasserstein-2 distance between the data distribution $p_{data}(x)$ and the approximated distribution $p_\theta(x)$ is bounded by*

$$\mathcal{W}_2(p_{data}(x), p_\theta(x)) \leq \mathrm{e}^{L_\theta}\epsilon_\theta \tag{43}$$

Lemma 3 (Error Bound for Reflected Flow Matching (Xie et al., 2024))**.** *For the reflected flow matching model over the convex domain $\mathcal{C}$: $x_1 = x_0 + \int_0^1 v_\theta(x,t) + R(x_t)\,\mathrm{d}t$, with induced probability distribution p_θ^r at $t = 1$. The Wassserstein-2 distance between the data distribution $p_{data}(x)$ and the approximated distribution $p_\theta^r(x)$ is bounded by*

$$\mathcal{W}_2(p_{data}(x), p_\theta^r(x)) \leq \mathrm{e}^{1/2+L_\theta}\epsilon_\theta \tag{44}$$

D.2 Proof of Regularity of GFM in Prop. 5.1

Proof. Given the original data distribution p over compact convex set $\Omega_p = \mathcal{C}$ satisfying the regularity conditions; We verify the three regularity conditions one by one for the transformed data distribution $q = \Phi_{\#}^{-1} p$ over a unit ball $\Omega_q = \mathcal{B}$, where Φ is a bi-Lipschitz homeomorphism (e.g., gauge mapping).

- **Finite second moment:** since q has bounded support, it immediately implies a finite second moment.

- **Positive reach:** The support of q is a unit ball $\Omega_q = \mathcal{B}$, which has infinite positive reach. We remark that even if the original support Ω_p does not have a positive reach (e.g., star-shaped set), after transformation, the Ω_q is always a unit ball and satisfies this regularity condition on the reach.
- **Non-vanishing density:** Given the bi-Lipschitz property, we have:

  $$L_1\|z_1 - z_2\| \leq \|\Phi(z_1) - \Phi(z_2)\| \leq L_2\|z_1 - z_2\|, \quad \forall z_1, z_2 \in \mathcal{B} \tag{45}$$

  where $L_1 = 1/L_{\Phi^{-1}}$ and $L_2 = L_\Phi$ are the (inverse) Lipschitz constants.

  For any $z \in \Omega_q \cap B_R(0)$ and any small radius $r < \frac{c}{L_2}$, where c is from the non-vanishing density property of p, we have:

  $$q(B_r(z)) = p(\Phi(B_r(z))) \tag{46}$$
  $$\geq p(B_{L_1 r}(\Phi(z))) \tag{47}$$

  since $\Phi(B_r(z)) \supseteq B_{L_1 r}(\Phi(z))$ by the bi-Lipschitz property.

  Since Φ is bi-Lipschitz, for any $R > 0$, there exists $R' > 0$ such that $\Phi(B_R(0) \cap \Omega_q) \subseteq B_{R'}(0) \cap \Omega_p$. By the non-vanishing density property of p, there exists a constant $C_{R'} > 0$ such that $p(B_{L_1 r}(\Phi(z))) \geq C_{R'}(L_1 r)^k$.

  Therefore, we have

  $$q(B_r(z)) \geq C_{R'} L_1^k r^k \tag{48}$$

  This establishes the non-vanishing density property for q.

□

D.3 Proof of Error Bound of GFM in Prop. 5.2

Proof.

$$\mathcal{W}_2^2(p_{data}, p_\theta^{gr}) = \inf_{\gamma=\Pi(p_{data}, p_\theta^{gr})} \{\int \|x_1 - x_2\|^2) d\gamma\} \tag{49}$$
$$= \inf_{\gamma=\Pi(q_{data}, q_\theta^{gr})} \{\int \|\Phi(z_1) - \Phi(z_2)\|^2) d\gamma\} \tag{50}$$
$$\leq L_\Phi^2 \inf_{\gamma=\Pi(q_{data}, q_\theta^{gr})} \{\int \|z_1 - z_2\|^2) d\gamma\} \tag{51}$$
$$\leq L_\Phi^2 \mathcal{W}_2^2(q_{data}, q_\theta^r) \tag{52}$$
$$\leq L_\Phi^2 \mathrm{e}^{1+2L_\theta} \epsilon_\theta^2 \tag{53}$$

□

where Eq. (50) is by the equivalence of the distributions under a homeomorphic push-forward mapping: $p_{\text{data}} = \Phi_\# q_{\text{data}}$ and $p^{gr}_{\text{data}} = \Phi_\# q^{qr}_{\text{data}}$; Eq. (51) is by the Lipschitz property of gauge mapping Φ shown in Prop. 4.1; Eq. (53) is the error bound of the regular reflected generation over the unit ball under Lemma 3.

Remark. The Lipschitz constant of gauge mappings between compact sets remains bounded inherently. This property stands in contrast to mirror mapping-based generative models (Liu et al., 2024b), which map open convex sets to $\mathbb{R}^n$. In the latter case, the Lipschitz constant can grow unbounded as points near the boundary are mapped to infinity, significantly complicating approximation error analysis. Our Gauge Flow Matching circumvents this limitation, providing theoretical guarantees on the Wasserstein-2 distance between the learned and data distributions. To optimize the model's performance, we can further reduce the Lipschitz constant of the gauge mapping by identifying an interior point x° that serves as the ”center” of the constraint set.

D.4 Proof of Generation Complexity of GFM in Prop. 5.3

Proof. Closed-form computation: Based on Table 8, the algorithmic complexity of naturally derived under different constraint functions.

Bisection-based computation: Based on algorithm 3, the algorithmic complexity for the bisection algorithm needs the feasibility check $x \in \mathcal{C}$, which essentially involves calculating the constraint function $g_i(x)$ for $i = 1, \dots, m$. The number of bisection steps needed to achieve a target error is derived as $\mathcal{O}(\mathrm{diam}(\mathcal{C}) \cdot \log(1/\epsilon_{\mathrm{bis}}))$.

□

E Experimental Settings

In this section, we describe the experimental setup used to generate the results reported in Section 7. Our proposed model is implemented in PyTorch (Paszke et al., 2019), with all models trained using Adam optimizer (Kingma & Ba, 2014) with hyperparameters $\beta_1 = 0.99$, $\beta_2 = 0.999$, and a learning rate of 10^{-3}. For sample generation, we solve the forward ODE using the method proposed by (Chen et al., 2018), and we follow the approach in (Xie et al., 2024) to solve the reflected ODE in Equation (5).

E.1 Constrained Generative Models Baselines

We compare our approach against the following state-of-the-art constrained generative models, selected for their ability to handle various constraint types:

- **FM**: Vanilla flow matching that transforms a Gaussian distribution to the target distribution using linear conditional flow (Lipman et al., 2022; Liu et al., 2022b).
- **DM**: Vanilla diffusion model with variance-preserving diffusion process (Ho et al., 2020; Song et al., 2020).
- **Reflection**: A method that applies reflection terms when generated samples encounter constraint boundaries (Xie et al., 2024).
- **Metropolis**: Metropolis Sampling approach for approximating reflection-based generation (Fishman et al., 2024).
- **Projection**: An approach utilizing orthogonal projection when generated samples violate constraints (Christopher et al., 2024).
- **GFM**: Our proposed framework as detailed in Section 4.

We remark that all models share the same training settings (dataset, optimizers, hyperparameters) and generation settings (ODE methods, step size). For Reflection, Metropolis, and Projection models, the same velocity models are used. Therefore, they share the same training time as reported in the Table. 5.

The additional constraint-handling mechanisms (e.g., reflection and projection) are implemented based on references. We emphasize that closed-form projection operators generally do not exist for the complex constraints in our experiments, such as intersections of polytopes and ellipsoids, and Semidefinite cone constraints. Therefore, we employ state-of-the-art optimization solvers: (i) MOSEK for convex projection and (ii) IPOPT for non-convex cases. While these are highly optimized solvers, iteratively solving a constrained optimization problem at each sampling step for each sample (whenever the ODE trajectory violates constraints) remains computationally expensive. This cost is particularly significant for high-dimensional problems or tight constraint sets where violations occur frequently throughout the sampling process.

We also remark that, in the geodesic-convex set generation tasks, the FM and DDPM models are trained to generate samples in the tangent space of a prescribed point, which is the same across all methods.

Algorithm 4 Euler method with reflection

Input: A velocity field $v(x_t, t)$, a domain $\mathcal{C}$, an initial point $x \in \mathcal{C}$, number of steps N, a function that gives outward normal vector $\boldsymbol{n}_{\partial\mathcal{C}}(x)$, and a function that computes the distance to boundary $s_{\partial\mathcal{C}}(x_t, v_t)$.

```
Let $\delta t = 1/N$, $x_0 = x$
for $i = 0, 1, 2, \dots, N-1$ do
    let $t = i/N$, $v_i = v(x_i, t)$
    update $x_{i+1} = x_i + \delta t \cdot v_i$
    if $x_{i+1} \notin \mathcal{C}$ then
        compute distance to boundary $s_i = \min\{1, s_{\partial\mathcal{C}}(x_i, \delta t \cdot v_i)\}$
        compute boundary point $x_i' = x_i + s\delta t \cdot v_i$
        compute normal vector $n_i = \boldsymbol{n}_{\partial\mathcal{C}}(x_i')$
        compute reflection term $L_i = -2\delta t(1 - s_i)v_i^T n_i n_i$
        update $x_{i+1} = x_{i+1} + L_i$
    end if
end for
```

Output: x_N

Table 9: Closed-form Expressions for Distance and Normal Vector Functions

Constraints	Formulation	Distance Function	Normal Vector
Unit Ball	$\lVert x\rVert_2 \le 1$	$s(x, v) = \{\mathrm{root}(v^T v, 2v^T x, x^T x - 1)\}^+$	$\boldsymbol{n}(x) = x/\lVert x\rVert$
Unit Cube	$\lVert x\rVert_\infty \le 1$	$s(x, v) = \min_{i=1,2,\dots,n}\{(1 - \mathrm{sign}(v_i)x_1)/v_i\}$	$\boldsymbol{n}(x) = \mathbf{e}_{\arg\max_{i=1,2,\dots,n} \lvert x_i\rvert}$
Linear	$a^T x \le b$	$s(x, v) = (b - a^T x)/a^T v$	$\boldsymbol{n}(x) = a/\lVert a\rVert$
Quadratic	$x^T Qx + a^T x \le b$	$s(x, v) = \{\mathrm{root}(A_Q, B_Q, C_Q)\}^+$	$\boldsymbol{n}(x) = -2Qx - a$

[1] Notation: $x, a \in \mathbb{R}^n, b \in \mathbb{R}, Q \in \mathbb{S}_+^n, A \in \mathbb{R}^{n\times k}, p \in \mathbb{R}^k, F_0, \cdots, F_n \in \mathbb{R}^{k\times k}$.

[2] $\boldsymbol{e}_1 = (1, 0, 0, \dots, 0), \boldsymbol{e}_2 = (0, 1, 0, 0, \dots, 0), \dots, \boldsymbol{e}_{n-1} = (0, 0, \dots, 0, 1, 0), \boldsymbol{e}_n = (0, 0, \dots, 0, 1)$.

[3] $A_Q = v^\top Qv,\ B_Q = 2x^{\circ\top} Qv + a^\top v,\ C_Q = x^{\circ\top} Qx^\circ + a^\top x^\circ - b$.

[4] $(\cdot)^+ = \max(\cdot, 0)$.

[5] $\mathrm{root}(x_1, x_2, x_3) = \frac{-x_2 \pm \sqrt{x_2^2 - 4x_1x_3}}{2x_1}$ denotes the quadratic equation solution

E.2 Reflection computation

We follow the notations in (Xie et al., 2024) and (Lou & Ermon, 2023). Let $\boldsymbol{L}_t$ be the reflection term that reflect the outward velocity at the boundary $\partial\mathcal{C}$. Given an initial point x_{init}, the reflected ordinary differential equation is

$$\mathrm{d}x_t = v(x_t, t)\mathrm{d}t + \mathrm{d}\boldsymbol{L}_t, \tag{54}$$

$$x_0 = x_{\text{init}}. \tag{55}$$

Intuitively, the reflection term $\boldsymbol{L}_t$ in equation (54) pushes the trajectory back to the domain $\mathcal{C}$ once the trajectory hits the boundary. Under mild conditions, the solution to the reflected ODE exists and is unique (Xie et al., 2024).

Empirically, the reflected ODE can be solved by numerical solvers, for example, using the Euler method. The Euler method iteratively compute the trajectory points by

$$x_{t+\delta t} = x_t + \delta t \cdot v(x_t, t), \tag{56}$$

where $t \in [0, 1)$ is an intermediate time and $\delta t > 0$ is a small time step. Euler step with reflection can be computed as

$$x_{t+\delta t} = x_t + \delta t \cdot \Big[v(x_t, t) - 2(1 - s_t)\big(v(x_t, t)\big)^T \boldsymbol{n}_{\partial\mathcal{C}} \boldsymbol{n}_{\partial\mathcal{C}}\Big] \tag{57}$$

where $s_t = \min\big\{1, \inf\{s > 0 \mid x_t + s\delta t \cdot v(x_t, t) \notin \mathcal{C}\}\big\}$ is the distance from x_t to the nearest boundary $\partial\mathcal{C}$ along direction $v(x_t, t)$, and $\boldsymbol{n}_{\partial\mathcal{C}}$ is the outward normal vector at $x_t + s\delta t \cdot v(x_t, t) \in \partial\mathcal{C}$. Algorithm 4 summarizes the Euler method with reflection and table 9 lists reflections for several common convex sets.

E.3 Performance Evaluation Metrics

Feasibility ratio computation Suppose that $x_1, x_2, \dots, x_N$ are generated samples, $g_i(x), i = 1, 2, \dots, m$ are constraints of interest. One sample x is feasible if

$$g_i(x) \leq 0, i = 1, 2, \dots, m. \tag{58}$$

The feasibility ratio of the batch samples is defined as

$$\text{feasibility}\{x_1, x_2, \dots, x_N\} = \frac{\#\{x_i \mid i = 1, 2, \dots, N, g_k(x_i) \leq 0, k = 1, 2, \dots, m\}}{N}. \tag{59}$$

In our experiments, we set the tolerance as 10^{-6}.

E.4 Constraint Formulations in Illustrative Examples

We first give the specific formulations of those constraints:

- **Convex Set**:
  $$\{x \in \mathbb{R}^2 \mid Ax \leq b,\ \|x - c\|_2 \leq 2.5,\ x^T Q x + p^T x + d \leq 0\} \tag{60}$$
  where A, b, c, Q, p, d are randomly sampled.
- **Star-convex Set**:
  $$\{x \in \mathbb{R}^2 \mid \|x\| \leq \Gamma_{\alpha,n}(x) := 1 + \alpha \sin(n \arctan(x_2/x_1))\} \tag{61}$$
  where $\alpha \in (0, 1)$ and $n \in \mathbb{Z}_+$ determine the size and number of stars, respectively.
- **Geodesic-convex Set**:
  $$\{x \in \mathbb{S}^2 \mid Ax \leq b\} \tag{62}$$
  where $\mathbb{S}^2$ is the sphere in 3-dimensional Euclidean space and is constrained by a set of linear inequalities.

The data are sampled from a mixed Gaussian distribution, whose location parameter and covariance are generated randomly.

Model settings: We model the velocity field with 2 hidden layers with exponential linear unit (ELU) activation functions. We train the models for 10000 epochs with a batch size of 256, and prior distribution as the uniform distribution over the constrained/ball domain. Samples are generated by the Euler algorithm in 1,000 steps. For the reflected/projected methods, additional reflection/projection is performed after each step.

E.5 Mirror Map *vs* Gauge Map on Simplex Domain

We compare our GFM model against several baseline methods: Vanilla Flow Matching, DDPM, and Mirror Diffusion. We consider the standard simplex in d dimensions:

$$S_d = \{x \in \mathbb{R}^d \mid x_i \geq 0, \sum_{i=1}^{d} x_i \leq 1\}. \tag{63}$$

Data preparation: We generate training data from a mixture of multivariate normal distributions, with each component centered near a distinct geometric feature of the simplex. The mixture includes d vertex-centered modes, one origin-centered mode, and one face-centered mode. The location parameters are defined as:

$$\begin{aligned}
\mu_i &= (\underbrace{0.1/d, \dots, 0.1/d}_{i}, 0.9, \underbrace{0.1/d, \dots, 0.1/d}_{d-i-1}), \quad i = 1, 2, \dots, d, \\
\mu_o &= (0.1/d, \dots, 0.1/d), \\
\mu_f &= (0.9/d, \dots, 0.9/d).
\end{aligned}$$

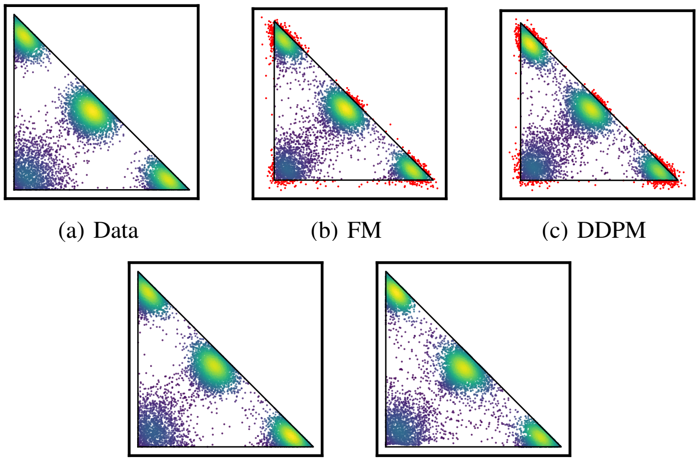
Here, μ_i places most of its mass at the i-th vertex, μ_o is centered near the origin, and μ_f distributes mass in the simplex face. We generate a total of 10,000 training samples from this mixture.

Model settings: For illustrative 2-dimensional cases, we modeled the time-variant velocity field using 3 layers with 256 units each and ELU activation functions. For 50-dimensional cases, we modeled the time-variant velocity field with 4 hidden layers of 512 hidden units, incorporating residual connections and a bottleneck structure (Liu et al., 2024a).

As shown in Table 10, in the synthetic simplex domain example, our GFM method achieves comparable performance to the vanilla flow matching baseline and other constrained generation baselines in low-dimensional settings ($d = 2$), while achieving significantly faster inference speed. In high-dimensional settings ($d = 50$), GFM attains the best fidelity to the data distribution while maintaining faster inference speed. It is worth noting that the Mirror Map model (Liu et al., 2024b) exhibits fast performance in both training and inference phases. This is because a low-complexity, closed-form mirror map exists for this simple simplex domain. However, the mirror map approach fails for more complex constraints, such as the PSD cone constraint.

Table 10: Results for synthetic generation tasks over the simplex domain.

	Method	FM	DDPM	Mirror Map	**GFM**
$d = 2$	Feasibility (%)	94.5	93.8	100	**100**
	MMD ($\times 10^{-3}$)	6.26	14.02	8.34	6.53
	Training (s)	0.09	0.09	0.13	0.13
	Inference (s)	0.40	0.94	0.46	0.97
$d = 50$	Feasibility (%)	11.7	4.35	100	**100**
	MMD ($\times 10^{-2}$)	9.08	1.82	6.00	2.56
	Training (s)	0.14	0.13	0.14	0.15
	Inference (s)	0.93	1.20	0.93	1.38



(a) Data (b) FM (c) DDPM

(d) Mirror Map (e) GFM

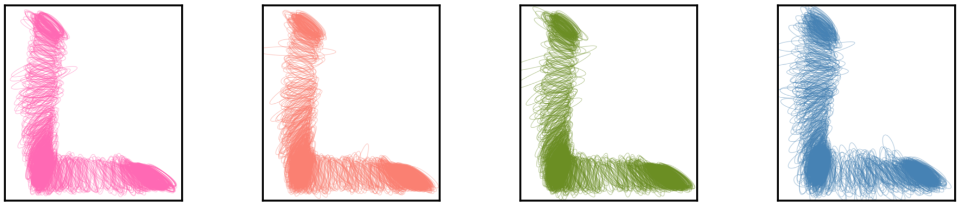
E.6 Configuration Generation in Robotic Control Benchmark

We follow the procedure reported by (Jaquier et al., 2021) to learn the trajectories of the manipulability ellipse. In planar letter drawing problems, the manipulability ellipses are modeled by SPD matrices $M = \{X \in \mathbb{S}^{2\times 2} \mid X \succeq 0,\ \mathrm{tr}(X) \le C\}$. In our formulation, we transform it into the equivalent linear matrix inequality formulation:

$$M = \{x \in \mathbb{R}^3 \mid x_1 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + x_3 \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \succeq 0, \quad \mathrm{tr}(M) = x_1 + x_3 \le C\} \tag{64}$$

Table 11: Results for robotic manipulability ellipse generation task.

Methods	FM	DDPM	GFM
Feasibility (%)	99.92	99.94	**100**
MMD ($\times 10^{-2}$)	5.39	5.58	9.85
Training (s)	3.43	3.64	3.55
Inference (s)	0.50	0.86	0.83



(f) Data (g) FM (h) DDPM (i) GFM

Figure 11: Visual comparison of 1000 samples between data distribution, DDPM model, Flow Matching model, and our GFM model on robotic manipulability ellipse generation task.

We additionally learn to sample the 2-dimensional trajectories. Therefore, our models are parameterized to generate samples in $\mathbb{R}^2 \times M$

Model settings: We model the time-variant velocity field with 3 hidden layers with 256 units each and ELU activation functions. We train the models for 10,000 epochs with a batch size of 256, and prior distribution as the uniform distribution over the constrained/ball domain.

Remark: The feasibility ratios reported in Table 11 are also high for methods that do not force feasibility, e.g., vanilla flow-matching and diffusion models. The possible reason for that phenomenon is that the training data is highly concentrated in the interior of the domain, and only a few data points are close to the boundary. Therefore, a few generated samples are infeasible.

E.7 Solution Generation for Relaxed Combinatorial Problems

We further evaluate our approach on high-dimensional solution generation for relaxed combinatorial problems (Kook & Vempala, 2024). The distribution we consider is particularly interesting as it recovers, as special cases, the sampling problems associated with Max-Cut SDP relaxations and minimum volume ellipsoid problems, defined as:

$$X \sim \exp\left(-(\langle A, X\rangle + \|X - B\|_F^2 + \|X - C\|_F - \log\det X)\right) \tag{65}$$

$$\text{s.t. } X \succeq 0, \langle D_i, X\rangle \geq c_i, \quad \forall i \in [m]. \tag{66}$$

The constraints ensure that X is a positive semidefinite $n \times n$ matrix satisfying m linear inequalities. Each term in the energy function captures different structural aspects of the optimization landscape, including linear objectives, quadratic penalties, proximity constraints, and determinant-based regularization.

By developing generative models for this class of problems, we aim to provide tools that not only sample feasible solutions but also help explore the solution space in ways that could reveal insights about the underlying combinatorial structure. This capability is particularly valuable for understanding the landscape of near-optimal solutions and for generating diverse candidate solutions that might be refined by downstream processes.

Data preparation: We sample from the target distribution using the Hit-and-Run (Bélisle et al., 1993) algorithm. The Hit-and-Run sampling algorithm is a Markov Chain Monte Carlo (MCMC) method that generates random samples from a high-dimensional distribution by iteratively selecting a random direction and moving to a new point along that direction according to the target distribution. To generate training samples, we use the Hit-and-Run sampler with 1000 burn-in, and sample new points using 1-dimensional Metropolis sampling with 100 burn-in.

Model settings: We model the time-variant velocity field with 3 hidden layers with $n^2/2$ hidden units with residual connections and bottleneck structure (Liu et al., 2024a). We train the models for 1000 epochs with a batch size of 256, and prior distribution as the uniform distribution over the constrained/ball domain.

E.8 Constrained Conditional Time Series Generation

We follow the experimental setup of (Narasimhan et al., 2024) to generate time-series data subject to physical and statistical constraints:

$$\underbrace{x_{\min} \le x_t \le x_{\max},}_{\text{capacity bounds}} \quad \underbrace{\mu_{\min} \le \operatorname{ave}(x_{1:T}) \le \mu_{\max},}_{\text{mean volume bounds}} \quad \underbrace{\operatorname{var}(x_{1:T}) \le \sigma^2_{\max}}_{\text{variance bounds}}. \tag{67}$$

These constraint bounds are derived from empirical statistics of the historical dataset. The time series is generated via conditional forecasting, using past observations as model input.

Data preparation: We use the PEMS-BAY traffic dataset (Yoon et al., 2019; Li et al., 2018), partitioned by days. The models learn to generate time series for one day conditioned on data from the previous day. Days with incomplete data are excluded from the dataset.

Model settings: We model the conditional time-variant velocity field using a 4-layer neural network with 64 hidden units per layer, incorporating residual connections and a bottleneck structure (Liu et al., 2024a). All models are trained for 4,000 epochs with a batch size of 180. For the prior distribution, we use a standard Gaussian for vanilla Flow Matching and projection-based methods, and a uniform distribution over the ball domain for GFM.

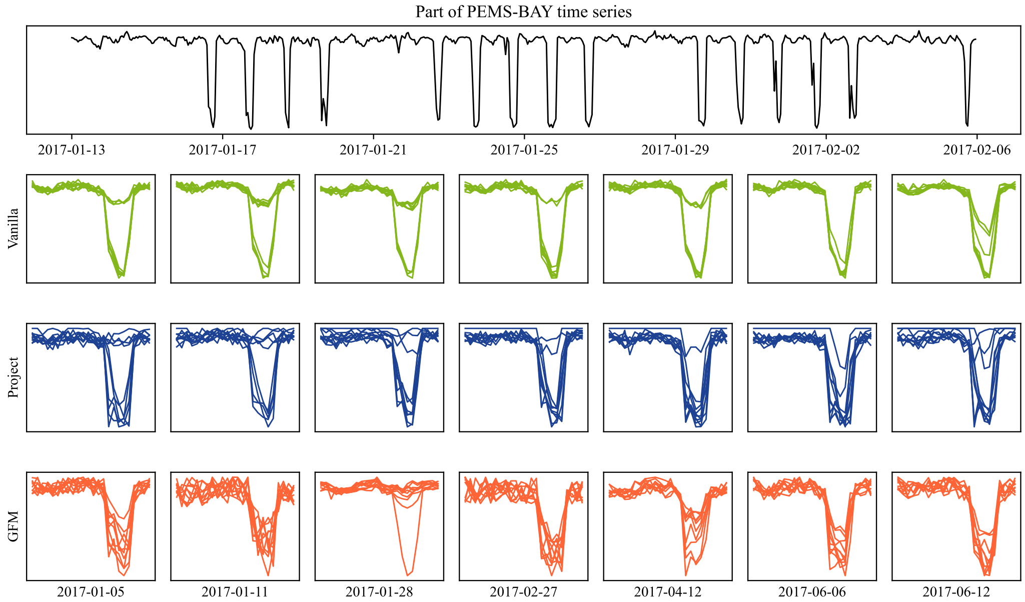
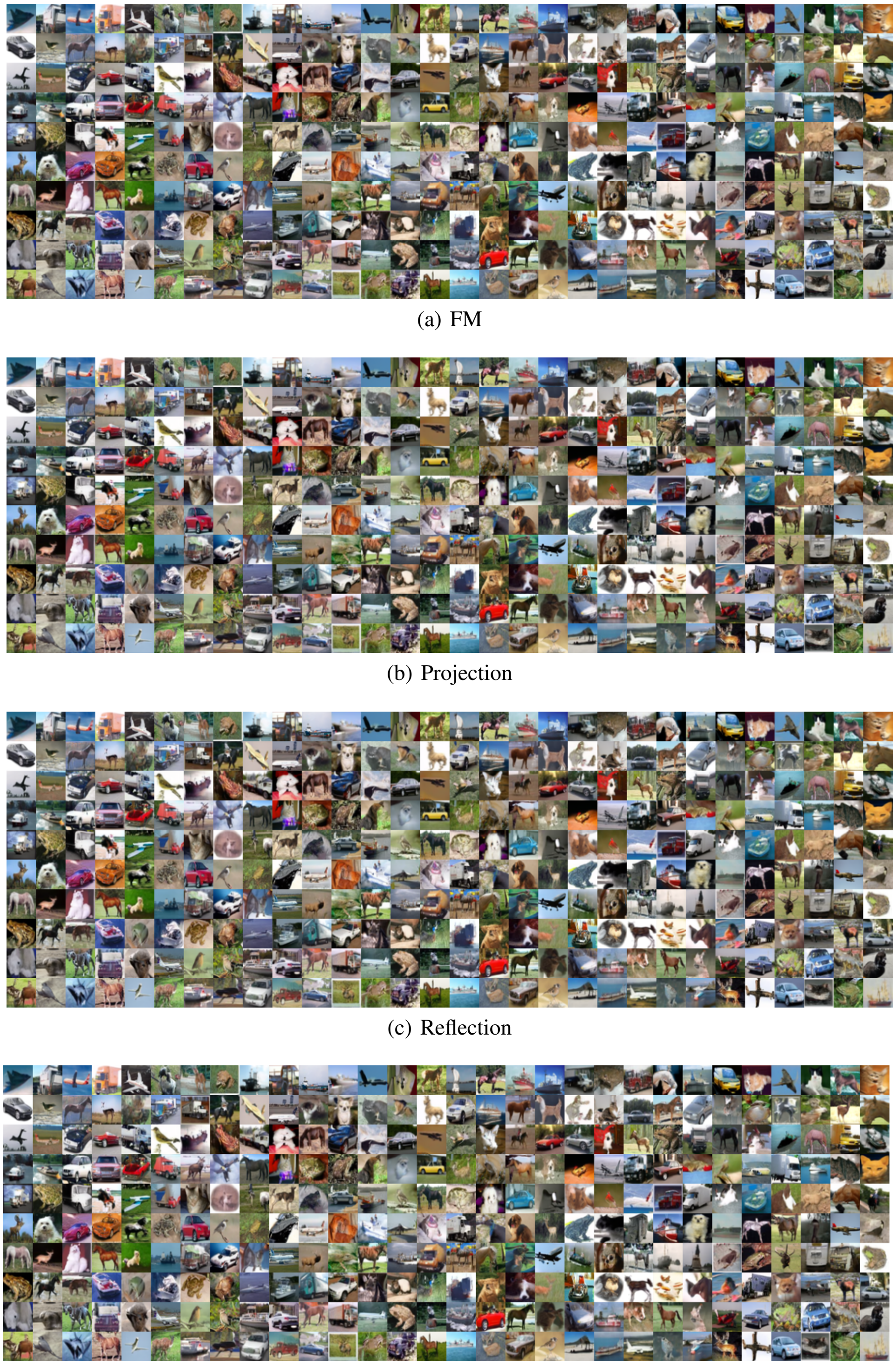


Figure 12: Visual comparison of 24 time series strides between data distribution, Vanilla Flow Matching model, Projection Flow Matching model, and our GFM model on constrained traffic time series generation task.

E.9 Watermarked Image Generation Tasks



(a) FM

(b) Projection

(c) Reflection

(d) GFM

Figure 13: CIFAR-10 samples with embedded watermarks generated by various methods using 100-step Euler sampling.

We evaluate GFM on constrained image generation with embedded watermarks using CIFAR-10, which consists of 32 × 32 RGB images. Following the watermarking methods in (Liu et al., 2024b), we partition pixels $\mathbf{x} \in [0,1]^d$ into unconstrained public pixels $\mathbf{x}_1$ and watermark pixels $\mathbf{x}_2 \in \mathcal{C}$, where $\mathcal{C}$ is a randomly sampled polytope for watermarking.

Model settings: All baselines are fine-tuned from a publicly available checkpoint of flow matching models (Tong et al., 2024), which share the same U-Net architecture (34M parameters). All models use identical fine-tuning hyperparameters (200 epochs) and sampling settings (100-step Euler method) on CIFAR-10 for fair comparison.

Data preparation: We select 32 pixels as watermarks constrained via a random polytope $\mathcal{C} = \{\mathbf{x}_2 : A\mathbf{x}_2 \leq b\}$ with $b > 0$. The watermarked training dataset is created by projecting the selected pixels in the original CIFAR-10 dataset onto the polytope $\mathcal{C}$, while other public pixels remain unchanged.

E.10 Setting of Scalability Tests of Gauge Mapping

To evaluate the scalability of gauge mapping, we conducted experiments with randomly generated inequality constraints. We tested scalability across three dimensions: problem dimension, batch size, and number of constraints.

We established a baseline configuration with 500-dimensional variables ($n = 500$), constraint dimension $k = 500$, $m = 8$ constraints, and batch size 256. For each scalability test, we varied one parameter while keeping the others fixed at their baseline values:

- **Problem dimension scalability:** We varied $n \in [10, 3000]$ while fixing $k = 500$, $m = 8$, and batch size 256. The specific dimensions tested follow Table 8.
- **Batch size scalability:** We varied the batch size from 10 to 100,000 while fixing $n = 500$, $k = 500$, and $m = 8$.
- **Constraint scalability:** We varied the number of constraints $m \in [8, 512]$ while fixing $n = 500$, $k = 500$, and batch size 256.

For each configuration, we measured the computation time of the gauge function evaluation. The results are illustrated in Fig. 8.

To demonstrate the efficiency of the bisection algorithm (Algorithm 3), we tested gauge computation on convex high-dimensional polynomials generated using the sum-of-squares (SOS) method.

A sum-of-squares polynomial of n variables and degree d is defined as

$$p(x_1, x_2, \dots, x_n) = \left(\text{mono}_d(x_1, x_2, \dots, x_n)\right)^T Q, \text{mono}_d(x_1, x_2, \dots, x_n), \tag{68}$$

where $\text{mono}_d(x_1, x_2, \dots, x_n)$ is a vector of all monomials up to degree d, and Q is a positive semi-definite matrix. Each monomial of degree k takes the form $x_1^{c_1} x_2^{c_2} \cdots x_n^{c_n}$ where $c_1, c_2, \dots, c_n \in \mathbb{N} \cup 0$ and $c_1 + c_2 + \cdots + c_n = k$.

We performed the bisection-based gauge function, and the results are summarized in Table 12.

Table 12: Gauge function computation time of high-dimensional polynomials using bisection

No. of polynomials	No. of variables	Degree	No. of monomials	Time (s)
10	10	4	4,356	0.025
10	50	4	1,758,276	0.492
10	100	4	26,532,801	5.592

E.11 Ablation Study

We consider the following ablation study to examine key components of GFM, including

- **Impacts of the interior point selection**: Central interior point *vs* Near-boundary interior point

- **Impacts of different generation strategies**: Reflection-based sampling *vs* Projection-based sampling
- **Impacts of the initial set**: ℓ_2-norm ball *vs* ℓ_∞-norm cube

We follow a similar setting in Sec. 7.1.

Table 13: Performances on joint convex set under different settings.

		L_2		L_∞	
		Central	Boundary	Central	Boundary
Reflection	MMD ($\times 10^{-2}$)	0.34	43.50	0.35	43.59
	Training (s)	0.22	0.22	0.21	0.23
	Inference (s)	0.91	0.86	0.86	0.88
Projection	MMD ($\times 10^{-2}$)	3.14	37.82	3.34	38.13
	Training (s)	0.22	0.22	0.21	0.23
	Inference (s)	0.48	0.50	0.47	0.50

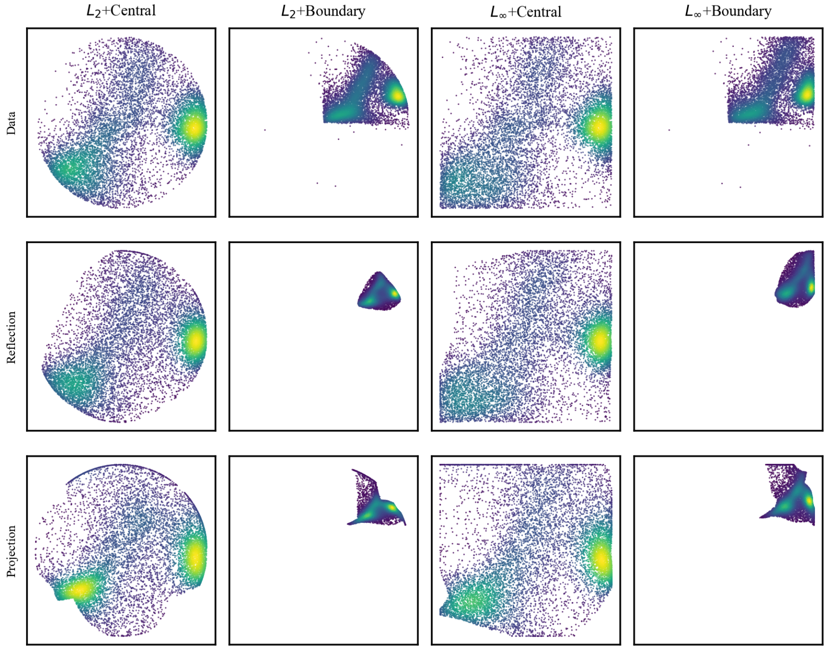


Figure 14: Visual comparison in the transformed space over L_2-norm **ball** and L_∞-norm **cube**: We compare gauge mappings using **central** versus **near-boundary** interior points, and evaluate samples generated via projection versus reflection-based strategies. Results show that gauge mapping with central interior points reduces distortion in the transformed space, facilitating more effective GFM training. Reflection-based sampling achieves superior generation quality compared to projection methods, while the choice of initial set shows negligible impact on generation quality.