GAUGE FLOW MATCHING FOR EFFICIENT CONSTRAINED GENERATIVE MODELING OVER GENERAL CONVEX SET

Xinpeng Li*, **Enming Liang***, **& Minghua Chen**†
Department of Data Science, City University of Hong Kong

ABSTRACT

Generative models, particularly diffusion and flow-matching approaches, have achieved remarkable success in various domains including image synthesis and robotic planning. However, a fundamental challenge remains: ensuring generated samples strictly satisfy problem-specific constraints—a crucial requirement for safety-critical applications and watermark embedding. Existing approaches, such as mirror maps and reflection methods, either support limited 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 that introduces a bijective gauge mapping to transform generation over arbitrary compact convex sets into an equivalent process over the unit ball. Our GFM framework guarantees strict constraint satisfaction with low computational complexity and bounded distribution approximation errors. Extensive numerical 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 synthesizing 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.

However, many real-world applications necessitate generation under specific constraints. Image generation might require watermark placement or physical consistency, inverse problems often involve physical constraints, and robotic manipulation must respect joint limits and obstacle avoidance. These constraints are fundamental to the problem domain and must be strictly satisfied for the generated samples to be meaningful and useful.

Existing approaches to constrained generative modeling face significant limitations. While some methods effectively handle specific constraint types, such as linear or simplex constraints, they lack generality across different constraint classes. Other approaches employ penalty-based methods that provide only approximate feasibility without guaranteed constraint satisfaction. A general, efficient framework for constrained generation over arbitrary compact convex sets remains an open challenge.

This work proposes Gauge Flow Matching, addressing these challenges with following contributions:

▷ We develop a novel gauge mapping that transforms generative modeling over arbitrary compact convex sets into an equivalent but simpler problem over the unit ball, where existing approaches such as reflection or projection can be efficiently applied.

▷ We provide comprehensive theoretical analysis and extensive empirical validation demonstrating our framework's effectiveness in terms of feasibility, approximation capability, and computational efficiency compared to state-of-the-art methods in constrained generative modeling.

*Equal contribution.
†Corresponding author (minghua.chen@cityu.edu.hk).

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

Methods	Constraint setting	Feasibility guarantee	Approximation bound	Training complexity	Inference complexity
RDM[a] (FKDB+23)	Convex	✓	✗	+++	+++
RDM[b] (LE23)	Cube/Simplex	✓	✗	++	+++
RSB (DCY+24)	Smooth + Bounded	✓	✓	+++	+++
RFM (XZY+24)	Convex	✓	✓	+	+++
Metropolis sampling (FKM+24)	Manifold	✗	✓	+	++
MDM (LCTT24)	Ball/Simplex	✓	✗	+	+
NAMM (FBB24)	(Non)-Convex	✗	✗	+++	+
Projection-based (SKZ+23; CBF24)	Convex	✓	✓	+	+++
Barrier methods (FKDB+23)	Convex	✓	✗	+	+
Penalty training (LDDB24; KDR24)	General	✗	✗	+	+
Gauge Flow Matching	Convex	✓	✓	+	+

[1] Training/inference complexity is compared with the unconstrained versions of those generative models.

2 Related Work

We review existing feasibility enforcement approaches for conventional end-to-end neural networks (Appendix A.1) and recent diffusion/flow-based generative models (Appendix A.2).

For constrained generative modeling, feasibility strategies differ fundamentally between conventional and modern generative models. While traditional VAEs and GANs can directly incorporate regular neural network feasibility methods from Appendix A.1, diffusion and flow-based models present unique challenges despite superior distribution approximation capabilities. These challenges arise from their generation mechanism, which relies on forward integration with neural network-approximated score functions or vector fields, rather than direct outputs. Table 1 summarizes specialized approaches addressing these challenges, with detailed discussions in Appendix A.2.

In summary, existing works either lack performance guarantees or have limited applicable scenarios. 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 its broader applicability, theoretical analysis, and enhanced computational efficiency.

3 Problem Statement

We consider flow matching[1]-based generative modeling for a data distribution p_{data} over a general compact *convex* [2] set $\mathcal{C} \subset \mathbb{R}^n$. The vanilla flow-matching model (LCBH+22; LGL22) is trained by matching the designed conditional flow (e.g., linear flow) as:

$$\min \quad \mathcal{L}(v_\theta) = \int_0^1 \mathbb{E}_{x_0,x_1,t}\left[\|v_\theta(x_t,t) - (x_1 - x_0)\|^2\right] \tag{1}$$

where $x_t = (1-t)x_0 + tx_1$ where $x_0 \sim p_0$, $x_1 \sim p_1$, 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 (typically Gaussian $p_0 = \mathcal{N}(0, I)$) into the target data distribution $p_1 = p_{\text{data}}$. 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.

However, the generated samples often deviate from the constraint set $\mathcal{C}$ due to two sources of error: neural network approximation error and numerical integration discretization error. While existing approaches address this challenge (see Table 1), they suffer from either limited applicability or high computational complexity. To address these limitations, we propose Gauge Flow Matching (GFM) for efficient constrained generative modeling over general compact convex sets in the next section.

[1]While we consider the flow-based generative models in this work, the proposed methodology can also be applied for diffusion-based models as discussed in Appendix B.

[2]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 (THH23; DRK20; LCL23; DWDS23), see Appendix C for details. For unbounded constraints, we may add additional box constraints to enforce physically meaningful limits.

4 GAUGE FLOW MATCHING (GFM)

We introduce our GFM framework. It employs gauge mapping—an explicit bijective mapping between two convex sets—to transform complex constrained generative modeling into 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 and transforms them back to the original space through forward gauge mapping.

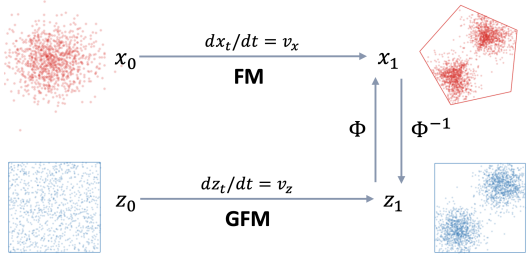


Figure 1: Gauge flow matching framework.

Gauge Mapping between Convex Sets: We first introduce a bijective mapping between two compact convex sets, termed gauge mapping:

Definition 4.1 (Gauge mapping (TZ22b))**.** Let $\varphi_{\mathcal{C}}(x, x^\circ) = \inf\{\lambda \geq 0 \mid x \in \lambda(\mathcal{C} - x^\circ)\}$ be the gauge function (BM08) given an interior point $x^\circ \in \mathcal{C}$. The gauge mapping $\Phi : \mathcal{B} \to \mathcal{C}$ can be defined between a unit p-norm ball and a compact convex set:

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

As shown in Figure 1, gauge mapping $\Phi(\cdot)$ establishes a bijective correspondence between any compact convex set and a unit p-norm ball: $\mathcal{C} = \Phi(\mathcal{B})$ and $\mathcal{B} = \Phi^{-1}(\mathcal{C})$. The gauge function $\varphi_{\mathcal{C}}(x, x^\circ)$ has *closed-form* expressions for common convex sets and can be efficiently computed via *bisection* methods for general convex constraints. Details are provided in Appendix C.

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

$$\min \quad \mathcal{L}(v_\theta) = \int_0^1 \mathbb{E}_{z_0, z_1, t} \left[\|v_\theta(z_t, t) - (z_1 - z_0)\|^2 \right], \tag{3}$$

where the initial samples z_0 are selected as a simple distribution within the unit ball (e.g., uniform), the terminal samples are transformed from the data sample from the target distribution $z_1 = \Phi^{-1}(x_1)$, and the conditional flow is linear as $z_t = (1-t)z_0 + tz_1$. We then leverage the regular flow matching training approach to train a neural network vector field v_θ following (3).

Remark 1. Mirror map-based generative models also employ a bijective mapping to transform constrained distributions to the unconstrained dual space (LCTT24). However, it is only computationally tractable for simple convex sets (ball and simplex), and it maps near-boundary samples to *infinity* in the dual spaces, challenging the transformed generative modeling. In contrast, gauge mapping is computationally efficient for any compact convex set and maintains bounded Lipschitz constants for both sides (see Appendix D for details), crucial for NN training and bounding approximation errors.

Inference Phase of GFM: After training, we generate the samples within the unit ball following the NN-based vector field with an additional reflection term (XZY+24):

$$z_1 = z_0 + \int_0^1 (v_\theta(z_t, t) + L(z_t))dt, \tag{4}$$

where z_0 is uniformly sampled from a unit ball, and $L(z_t)$ is the reflection term when z_t hits the constraint boundary, which has a *closed-form* expression for a simple ball or cube. Finally, we recover the sample to the original space following the forward gauge mapping as $x_1 = \Phi(z_1)$.

Remark 2. The existing reflection-based generative models also utilize an additional reflection term to keep the generation trajectory within the constraint set (LE23; XZY+24). However, they are computationally expensive beyond simple sets (e.g., ball and simplex), thus limiting their potential for more complex sets. In contrast, after transforming the generative modeling over a unit ball through gauge mapping, we can easily implement a closed-form reflection term with $\mathcal{O}(n)$ complexity, thus ensuring efficient sample generation within the ball and strict sample feasibility by gauge mapping.

Performance Analysis: In Appendix D, we provide comprehensive performance analysis for GFM regarding its distribution approximation error, feasibility guarantee, and run-time complexity over general convex sets, and discuss the impacts of key designs in GFM on those metrics. We also discuss its limitations and extensions for more general constraints in Appendix B.

Table 2: Performance comparison on low-dim. convex-constrained generation task.

Constraint	Metrics	Vanilla FM	Reflected FM	Projected FM	GFM
Polytope ($n = 2$)	MMD (↓)	0.03856	0.03869	0.03866	0.05708
	Feasibility rate (%)	97.56	100	100	100
	Inference Time (s)	3.230	5.162	4.924	3.398
Quadratic set ($n = 3$)	MMD (↓)	0.03056	0.03117	-	0.05949
	Feasibility rate (%)	98.29	100	-	100
	Inference Time (s)	3.299	10.19	>3600	3.293

[1] Solving projection onto quadratic constraint set (i.e., convex QCQP) by (DB16) incurs significant computational costs.

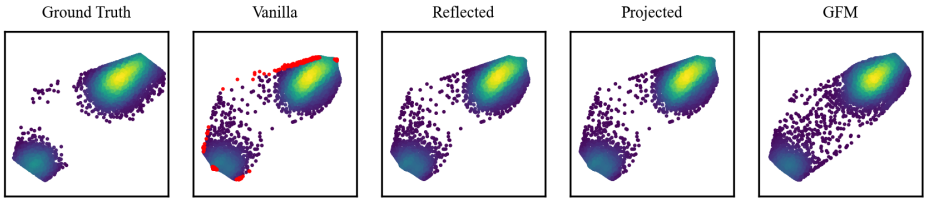


Figure 2: The visualization over 2-dim generative task

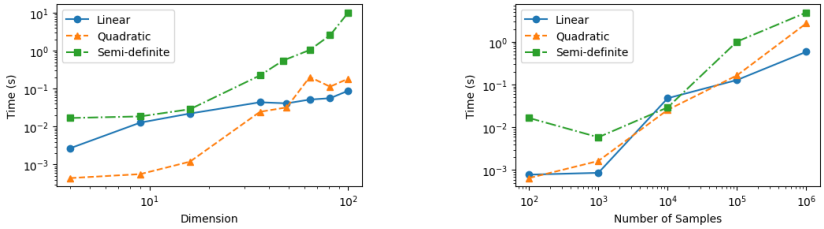


Figure 3: Inference time of gauge mapping over constraint dimension and generated samples.

5 Empirical Study

We conduct extensive simulations to demonstrate the effectiveness of the GFM framework. **Baselines**: we consider the following constrained generative models: (i) vanilla flow matching (LCBH+22); (ii) reflected flow matching (XZY+24); (iii) projected generation (CBF24); and (iv) the proposed gauge flow matching. **Metrics**: we evaluate those baselines based on (i) constraint satisfaction, (ii) distribution approximation quality, and (iii) run-time complexity. Detailed experimental settings are provided in Appendix E.

Low-dim. Toy Example: We first evaluate GFM's efficiency in low-dimensional convex-constrained domains. As shown in Table 2 and Figure 2, for simple polytope constraints, GFM achieves 100% feasibility rate and comparable MMD scores to baselines while reducing computational costs. For more complex quadratic inequality constraints where conventional reflection and projection-based models fail due to lack of closed-form solutions, GFM maintains both computational efficiency and MMD performance comparable to vanilla FM.

High-dim. Scalability Tests: We then evaluate GFM's scalability in high-dimensional convex-constrained spaces by measuring the forward computational time of gauge mapping when transforming the generated samples from a unit ball to the target constraint set. As shown in Figure 3, it remains efficient for various convex constraints up to 100 dimensions while generating 10^6 batched samples, demonstrating its potential for high-dimensional generative tasks.

Constrained Generation for Robotic Control: We apply GFM to constrained generative modeling for robotic control, focusing on manipulability analysis. A robot's manipulability is represented by a symmetric positive-definite (SPD) matrix $M \in \mathbb{S}^d_{++}$ with trace constraints $\mathrm{tr}(M) \leq C$. Using the planar robotic arms benchmark for letter drawing (JRCC21), Figure 5 demonstrates the generated velocity manipulation ellipses $M \in S^2_{++}$ and their corresponding trajectories $x \in \mathbb{R}^2$. GFM successfully models this joint distribution while maintaining zero constraint violations.

Ablation Study: In Appendix E.3, we present a comprehensive ablation study examining key components of GFM, including (i) the effect of norm choice for the unit ball; (ii) the impact of interior point x° selection; and (iii) alternative strategies for unit ball sampling beyond reflection.

REFERENCES

[AAB+19] 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.

[AK17] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

[BDD23] Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *arXiv preprint arXiv:2305.16860*, 2023.

[BM08] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*, volume 78. Springer, 2008.

[CBF24] 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.

[CDB+21] 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*, pages 199–210, 2021.

[COMB19] 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, pages 3387–3395, 2019.

[CRBD18] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[CSRY22] 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.

[DB16] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

[DCY+24] Wei Deng, Yu Chen, Nicole Tianjiao Yang, Hengrong Du, Qi Feng, and Ricky TQ Chen. Reflected schr\" odinger bridge for constrained generative modeling. *arXiv preprint arXiv:2401.03228*, 2024.

[DKP+24] 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.

[DRK20] 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.

[DWDS23] 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.

[FBB24] Berthy T Feng, Ricardo Baptista, and Katherine L Bouman. Neural approximate mirror maps for constrained diffusion models. *arXiv preprint arXiv:2406.12816*, 2024.

[FKDB+23] Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael Hutchinson. Diffusion models for constrained domains. *arXiv preprint arXiv:2304.05364*, 2023.

[FKM+24] 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.

[FL23] Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.

[FNC20] 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*, pages 748–749, 2020.

[JRCC21] 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.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KDR24] Shervin Khalafi, Dongsheng Ding, and Alejandro Ribeiro. Constrained diffusion models via dual training. *arXiv preprint arXiv:2408.15094*, 2024.

[KFL22] Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the wasserstein distance. *arXiv preprint arXiv:2212.06359*, 2022.

[KU23] 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, pages S233–S241. Springer, 2023.

[KZLD21] Anastasis Kratsios, Behnoosh Zamanlooy, Tianlin Liu, and Ivan Dokmanić. Universal approximation under constraints is possible with transformers. In *International Conference on Learning Representations*, 2021.

[LAL+21] Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher Strong, Clark Barrett, Mykel J Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4):244–404, 2021.

[LCBH+22] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[LCL23] 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.

[LCL24] 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.

[LCTT24] 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, 2024.

[LDDB24] 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.

[LE23] Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pages 22675–22701. PMLR, 2023.

[LGL22] 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*, 2022.

[LKM23] Meiyi Li, Soheil Kolouri, and Javad Mohammadi. Learning to solve optimization problems with hard linear constraints. *IEEE Access*, 2023.

[LM23] Meiyi Li and Javad Mohammadi. Toward rapid, optimal, and feasible power dispatch through generalized neural mapping. *arXiv preprint arXiv:2311.04838*, 2023.

[LW23] Xingchao Liu and Lemeng Wu. Learning diffusion bridges on constrained domains. In *international conference on learning representations (ICLR)*, 2023.

[NC21] 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)*, pages 419–424. IEEE, 2021.

[PCZL22] 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*, pages 42–47, 2022.

[PGM+19] 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.

[PZCZ20] 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.

[SDCS23] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[SKZ+23] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. *arXiv preprint arXiv:2307.08123*, 2023.

[THH23] Jesus Tordesillas, Jonathan P How, and Marco Hutter. Rayen: Imposition of hard convex constraints on neural networks. *arXiv preprint arXiv:2307.08336*, 2023.

[TVH24] Mathieu Tanneau and Pascal Van Hentenryck. Dual lagrangian learning for conic optimization. *arXiv preprint arXiv:2402.03086*, 2024.

[TZ22a] Daniel Tabas and Baosen Zhang. Computationally efficient safe reinforcement learning for power systems. In *2022 American Control Conference (ACC)*, pages 3303–3310. IEEE, 2022.

[TZ22b] 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)*, pages 1142–1147. IEEE, 2022.

[uAYKJ22] Zain ul Abdeen, He Yin, Vassilis Kekatos, and Ming Jin. Learning neural networks under input-output specifications. In *2022 American Control Conference (ACC)*, pages 1515–1520. IEEE, 2022.

[UZB+24] 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.

[VQLC20] Andreas Venzke, Guannan Qu, Steven Low, and Spyros Chatzivasileiadis. Learning optimal power flow: Worst-case guarantees for neural networks. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7. IEEE, 2020.

[WZG+23] 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*, pages 36605–36625. PMLR, 2023.

[XZY+24] 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.

[ZPC+20] Tianyu Zhao, Xiang Pan, Minghua Chen, Andreas Venzke, and Steven H Low. Deepopf+: A deep neural network approach for dc optimal power flow for ensuring feasibility. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2020.

[ZPCL23] Tianyu Zhao, Xiang Pan, Minghua Chen, and Steven H Low. Ensuring dnn solution feasibility for optimization problems with convex constraints and its application to dc optimal power flow problems. In *International Conference on Learning Representations*, 2023.

[ZSRZ21] 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*, pages 336–347. PMLR, 2021.

A RELATED WORK

A.1 CONVENTIONAL NEURAL NETWORK 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 (PCZL22; DRK20). Penalty of output constraint violations can be incorporated into loss functions to improve NN feasibility (COMB19; PZCZ20; NC21).

Strict Satisfaction Methods: For exact equality constraint satisfaction, prediction-then-reconstruct or completion techniques can be applies (DRK20; PCZL22; LCL23) For more general constraint enforcement, orthogonal/L2 projection is often employed. However, solving the projection problem either by iterative solver or equivalent optimization layers (AK17; AAB+19; CDB+21; WZG+23) is computationally intensive in real-time. More efficient homeomorphic projection can also be applied at the cost of minor optimality loss (LCL23; LCL24).

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 (FNC20; ZSRZ21). For general compact but fixed constraint sets, probabilistic transformer utilizes feasible samples to ensure feasibility (KZLD21). However, scalability remains a challenge due to the exponential growth in required samples with increasing problem dimensionality.

Preventive learning. a Preventive Learning framework is proposed for ensuring linear constraint feasibility in (ZPC+20; ZPCL23). It first adjusts inequality constraints to account for NN prediction errors. Subsequently, it trains the NN using mixed-integer programming techniques to limit the worst-case prediction error. However, it lacks an optimality guarantee. Additionally, other NN verification techniques can also be applied to assess the worst-case performance (VQLC20; uAYKJ22; LAL+21).

Gauge function. These works utilize gauge functions (BM08) 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 (TZ22a; TZ22b; LKM23). For fixed convex constraints, RAYEN and several works apply analytic expressions for gauge functions to find feasible boundary solutions (THH23; KU23; LM23; TVH24). However, these approaches only work for convex sets.

A.2 CONSTRAINED GENERATIVE MODELING

Recent advances in diffusion/flow-matching based constrained generative modeling have several directions:

Reflected Process: These approaches leverage reflection mechanisms to constrain generation trajectories within feasible regions. Different methods have been proposed for training score functions with reflection terms: RDM^a employs implicit score matching (FKDB+23), RDM^b develops an approximated denoising score matching approach (LE23), and RSB utilizes iterative proportional fitting (DCY+24). RFM extends this framework to flow-based generation over convex sets by incorporating reflected directions into ODEs (XZY+24). While effective, these reflection-based methods incur significant computational overhead during forward integration, requiring complex boundary localization and reflection calculations. A recent Metropolis sampling approach addresses the computational burden of reflection calculations (FKM+24), proving convergence to reflected Brownian motion as the step size approaches zero, though it lacks strict feasibility guarantees.

Bijective Map: These approaches utilize bijective mappings to transform constrained domains. RDM maps simplexes to unit cubes, enabling scalable denoising score matching (LE23). MDM employs mirror maps to transform constrained data into unconstrained dual space (LCTT24), though its applicability is limited to simple convex sets like balls and simplexes. NAMM generalizes this approach using neural networks to approximate mirror maps for arbitrary sets (FBB24), but lacks theoretical guarantees for feasibility and distribution accuracy. This bijective mapping framework has also found applications in continuous embedding of discrete data (DKP+24).

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 (LW23). Log-barrier diffusion models maintain feasibility through logarithmic barrier functions (FKDB+23). PDM enforces constraints via iterative projection (CBF24), a technique that has been successfully adapted for manifold constraints (CSRY22; SDCS23).

Penalty Training/Fine-tuning: Constraint violation penalties can be directly incorporated into diffusion model training objectives (LDDB24). Recent work has introduced Lagrangian-based training with dual variable updates for handling KL-constraints (KDR24). Post-training fine-tuning with custom loss functions, including constraint penalties, offers another pathway for improving constraint satisfaction (FL23; UZB+24).

B LIMITATIONS AND FUTURE DIRECTIONS

While our GFM framework demonstrates promising results, several important limitations and future research directions warrant discussion:

Our work primarily focuses on flow-matching-based generative modeling, though we note that the gauge mapping can also enable diffusion-based generative modeling over a unit ball, which can then be transformed back to the constraint set to enforce feasibility. In the inference phase, while we generate samples within a ball through a reflection-based scheme, we note that projection-based generation could also be applied to enforce feasibility within a ball with low computational complexity. Theoretical characterization of these design choices remains an important direction for future work.

The current GFM framework primarily addresses compact convex constraint sets, encompassing linear equality constraints and convex inequality constraints. Extending the current framework to more general constraint sets presents interesting future directions. For manifold constraints, one could extend the gauge mapping for convex constraints over manifolds to handle more general generative modeling over manifolds. For non-convex inequality constraints, we can extend the gauge mapping to star-shaped non-convex sets by identifying proper interior points. More general non-convex sets may require developing more sophisticated bijective mappings.

Given the current trend toward distillation and consistency training for one-step generation, incorporating constraints during these processes could lead to better feasibility performance. A theoretical understanding of the trade-off between generation speed and sample fidelity, particularly regarding feasibility and optimality, warrants further investigation.

The current framework assumes fixed constraints during training and inference. However, many practical applications require handling dynamic, input-dependent constraints or guided generation scenarios. Extending GFM to a conditional version for dynamic constraints remains an open challenge. This could involve developing architectures that jointly learn condition-dependent bijective mappings and generative modeling to handle input-dependent constraint sets.

These limitations present exciting opportunities for future research in constrained generative modeling. Addressing them would significantly expand the practical applicability of gauge flow matching to more complex real-world scenarios.

C GAUGE MAPPING OVER GENERAL CONVEX SETS

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

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid Ax = b,\ g_1(x) \leq 0, \cdots, g_m(x) \leq 0\}, \tag{5}$$

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.

C.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:

$$A_1x_1 + A_2x_2 = b \tag{6}$$

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

$$x_2 = \phi(x_1) = A_2^{-1}(b - A_1x_1) \tag{7}$$

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

$$\mathcal{C}^s = \{x_1 \in \mathbb{R}^{n-r} \mid g([x_1, \phi(x_1)]) \leq 0\} \tag{8}$$

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

C.2 GAUGE MAPPING FOR INEQUALITY CONSTRAINTS

Definition C.1 (Gauge/Minkowski function (BM08))**.** The Gauge/Minkowski function for a compact convex set $\mathcal{C}$ is defined as $\varphi_{\mathcal{C}}(x, x^\circ) = \inf\{\lambda \geq 0 \mid x \in \lambda(\mathcal{C} - x^\circ)\}$, where $x^\circ \in \mathcal{C}$ is an interior point of the compact convex set $\mathcal{C}$.

Table 3: Closed-form Expressions for Inverse Distance Functions (THH23)

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

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

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

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

[4] $\mathrm{eig}(X) = \lambda_1, \cdots, \lambda_n$ denotes the eigenvalues satisfying $\det(X - \lambda I) = 0$

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

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

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

Algorithm 1 Bisection Algorithm for Inverse Point-to-Boundary Distance

Input: A compact convex set $\mathcal{C}$, an interior point $x^\circ \in \mathcal{C}$, and a unit vector v.

```
α_l = 0, and α_u = 1
while |α_l − α_u| ≥ 10^−3 do
    if x° + α_u · v ∈ C then
        increase lower bound: α_l ← α_u
        double upper bound: α_u ← 2 · α_m
    else
        bisection: α_m = (α_l + α_u)/2
        if x° + α_m · v ∈ C then
            increase lower bound: α_l ← α_m
        else
            decrease upper bound: α_u ← α_m
        end if
    end if
end while
```

Output: inverse distance: $\gamma_{\mathcal{C}}(x^\circ, v) = 1/\alpha_m$

The Gauge function generalizes the concept of a norm function. For instance, when $\mathcal{C}$ is a p-norm ball, the gauge function reduces to the p-norm: $\varphi_{\mathcal{B}}(x, 0) = \lVert x \rVert_p$. Like norms, the gauge function exhibits key properties, including non-negativity, positive homogeneity, and sub-additivity.

Building upon this foundation, we define the gauge mapping between two compact convex sets:

Definition C.2 (Gauge mapping (TZ22b))**.** The gauge mapping between two compact convex sets $\mathcal{Z}$ and $\mathcal{X}$ is defined as:

$$\Phi(z) = \frac{\varphi_{\mathcal{Z}}(z - z^\circ, z^\circ)}{\varphi_{\mathcal{X}}(z - z^\circ, x^\circ)}(z - z^\circ) + x^\circ,\ z \in \mathcal{Z} \setminus z^\circ \quad \Phi^{-1}(x) = \frac{\varphi_{\mathcal{X}}(x - x^\circ, x^\circ)}{\varphi_{\mathcal{Z}}(x - x^\circ, z^\circ)}(x - x^\circ) + z^\circ,\ x \in \mathcal{X} \setminus x^\circ \tag{9}$$

with additional definitions as: $\Phi(z^\circ) = x^\circ$ and $\Phi^{-1}(x^\circ) = z^\circ$.

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

$$\Phi(z) = \frac{\lVert z \rVert_p}{\varphi_{\mathcal{C}}(z, x^\circ)} z + x^\circ,\ \forall z \in \mathcal{B}, \qquad \Phi^{-1}(x) = \frac{\varphi_{\mathcal{C}}(x - x^\circ, x^\circ)}{\lVert x - x^\circ \rVert_p}(x - x^\circ),\ \forall x \in \mathcal{C}, \tag{10}$$

For computational purposes, we introduce an inverse distance function (THH23):

$$\gamma_{\mathcal{C}}(x^\circ, v) = \inf\{\lambda^{-1} \mid x^\circ + \lambda v \in \mathcal{C}\} \tag{11}$$

where x° is an interior point and v is a unit vector. For simplicity, we denote the point-to-boundary distance function as $d_{\mathcal{C}}(x^\circ, v) = 1/\gamma_{\mathcal{C}}(x^\circ, v)$

This function relates to the gauge function through:

$$\varphi_{\mathcal{C}}(x, x^\circ) = \gamma_{\mathcal{C}}(x^\circ, x/\|x\|) \cdot \|x\|_2 \tag{12}$$

Table 3 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 online given v. When the inverse distance function lacks an explicit expression, we employ an efficient bisection algorithm detailed in Alg. 1. This algorithm supports batch processing, enabling efficient parallel computation for multiple inputs simultaneously.

D THEORICAL RESULTS

Assumption 1. We made the following assumptions for the error analysis.

▷ The NN-based velocity model v_θ is L_θ-Lipschitz in x uniformly on t

▷ The approximation error of the NN-based velocity model is bounded as: $\epsilon_\theta^2 = \int_0^1 \mathbb{E}_{p_t(x)} \|v_\theta(x,t) - v_t(x)\|^2 \, \mathrm{d}t$

We remark that those assumptions are common for error analysis for the flow/diffusion-based generative models (KFL22; BDD23; XZY$^+$24).

Lemma 1 (Error Bound for Flow Matching (BDD23))**.** *For 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 squared Wasserstein-2 distance between the data distribution $p_{data}(x)$ and the approximated distribution $p_\theta(x)$ is bounded by*

$$W_2^2(p_{data}(x), p_\theta(x)) \leq \mathrm{e}^{2L_\theta} \epsilon_\theta^2 \tag{13}$$

Lemma 2 (Error Bound for Reflected Flow Matching (XZY$^+$24))**.** *For reflected flow matching model: $x_1 = x_0 + \int_0^1 v_\theta(x,t) + L(x_t)\,\mathrm{d}t$, with induced probability distribution p_θ^r at $t = 1$. The squared Wassserstein-2 distance between the data distribution $p_{data}(x)$ and the approximated distribution $p_\theta^r(x)$ is bounded by*

$$W_2^2(p_{data}(x), p_\theta^r(x)) \leq \mathrm{e}^{1+2L_\theta} \epsilon_\theta^2 \tag{14}$$

Proposition D.1 (Error Bound for Gauge Flow Matching)**.** *Consider the gauge flow matching with a gauge mapping Φ and transformed data distribution as $q_{data} = \Phi_\#^{-1} p_{data}$, where $\#$ is the push-forward operator.*

▷ *For GFM model with reflected generation: $x_1 = \Phi(z_0 + \int_0^1 v_\theta(z,t) + L(z_t)\,\mathrm{d}t)$, with induced probability distribution p_θ^{gr} at $t = 1$. The squared Wassserstein-2 distance between the data distribution $p_{data}(x)$ and the approximated distribution $p_\theta^{gr}(x)$ is bounded by*

$$\mathcal{W}_2^2(p_{data}, p_\theta^{gr}) \leq L_\Phi^2 \mathrm{e}^{1+2L_\theta} \epsilon_\theta^2 \tag{15}$$

where L_ϕ is the Lipschitz of the gauge mapping over $\mathcal{B}$.

Proof.

$$\begin{aligned}
\mathcal{W}_2^2(p_{data}, p_\theta^{gr}) &= \inf_{\gamma = \Pi(p_{data}, p_\theta^{gr})} \{\int \|x_1 - x_2\|^2) d\gamma\} && (16)\\
&= \inf_{\gamma = \Pi(q_{data}, q_\theta^{gr})} \{\int \|\Phi(z_1) - \Phi(z_2)\|^2) d\gamma\} && (17)\\
&= L_\Phi^2 \inf_{\gamma = \Pi(q_{data}, q_\theta^{gr})} \{\int \|z_1 - z_2\|^2) d\gamma\} && (18)\\
&\leq L_\Phi^2 \mathcal{W}_2^2(q_{data}, q_\theta^r) && (19)\\
&\leq L_\Phi^2 \mathrm{e}^{1+2L_\theta} \epsilon_\theta^2 && (20)
\end{aligned}$$

□

Proposition D.2 (Lipschitz of Gauge Mapping)**.** *Let a scaling function defined as: $s(x^\circ, z) = d_{\mathcal{C}}(x^\circ, \frac{z}{\|z\|}) \frac{\|z\|_p}{\|z\|_2}$, where $d_{\mathcal{C}}(x^\circ, \frac{z}{\|z\|})$ is the distance function from an interior point $x^\circ \in \mathcal{C}$ along a direction $\frac{z}{\|z\|}$ for a compact convex set $\mathcal{C}$. The Lipschitz of the gauge mapping between a p-norm $\mathcal{B}_p$ and a compact convex set can be bounded as:*

$$L_\Phi \leq \underbrace{\sup_{z \in \mathcal{B}} \{s(x^\circ, z)\}}_{\textit{maximum scaling ratio}} + \underbrace{\sup_{z \in \mathcal{B}} \{\| \frac{\partial s(x^\circ, z)}{\partial z} \|\}}_{\textit{maximum variation of scaling ratio}} \tag{21}$$

Proof. Consider the gauge mapping between p-norm ball $\mathcal{B}_p$ and a compact convex set $\mathcal{C}$:

$$\Phi(z) = \underbrace{d_{\mathcal{C}}(x^\circ, \frac{z}{\|z\|}) \frac{\|z\|_p}{\|z\|_2}}_{\text{scaling function: } s(x^\circ,\, z)} \cdot z + x^\circ,\ \forall z \in \mathcal{B} \tag{22}$$

First, the distance function is sub-differentiable for a convex set defined by a set of differentiable convex inequality. To calculate the Lipschitz of the gauge mapping, we consider the Jacobian of Φ

$$J_\Phi(z) = s(x^\circ, z) \cdot I + \frac{\partial s(x^\circ, z)}{\partial z} z^\top \tag{23}$$

The Lipschitz of a mapping can be expressed as the maximum singular value of its Jacobian over $\mathcal{B}$:

$$L_\Phi = \sup_{z \in \mathcal{B}} \{\sigma_{\max}(J_\Phi(z))\} \tag{24}$$

$$\leq \sup_{z \in \mathcal{B}} \{\sigma_{\max}(s(x^\circ, z) \cdot I) + \sigma_{\max}(\frac{\partial s(x^\circ, z)}{\partial z} z^\top)\} \tag{25}$$

$$\leq \underbrace{\sup_{z \in \mathcal{B}} \{s(x^\circ, z)\}}_{\text{maximum scaling ratio}} + \underbrace{\sup_{z \in \mathcal{B}} \{\| \frac{\partial s(x^\circ, z)}{\partial z} \|\}}_{\text{maximum variation of scaling ratio}} \tag{26}$$

□

The Lipschitz constant of gauge mappings between compact sets remains inherently. This property stands in stark contrast to mirror mapping-based generative models (LCTT24), 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 minimize the Lipschitz constant of the gauge mapping by identifying an interior point x° that serves as the "center" of the constraint set. Ideally, this point should maintain uniform distances to all boundaries of the constraint set. Consider a unit ball $\mathcal{C}$ as an illustrative example: when x° is positioned at its geometric center, the Jacobian matrix reduces to the identity matrix since the distance from x° to any boundary point equals 1, yielding zero directional derivatives.

In practice, we seek such a "central" interior point by solving the following residual minimization problem through convex optimization in the offline phase (THH23):

$$\min_{x^\circ} \eta \tag{27}$$

$$\text{s.t. } g_i(x^\circ) \leq \eta \quad i = 1, \cdots, m \tag{28}$$

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.

For the online inference complexity, we establish the following proposition:

Proposition D.3 (Inference Complexity for Gauge Flow Matching)**.** *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 computational overhead for GFM inference—specifically, forward integration and the gauge mapping calculation—is $\mathcal{O}(\text{NFE} \cdot n^2 + mC_i)$, where C_i denotes the complexity of computing the point-to-boundary distance λ for a unit vector v such that $g_i(x^\circ + \lambda v) = 0$. The complexity C_i varies by constraint type:*

1. *For linear constraints $g_i(x) = a^\top x - b \leq 0$, $C_i \sim \mathcal{O}(n)$.*
2. *For quadratic constraints $g_i(x) = x^\top Q x + a^\top x - b \leq 0$, $C_i \sim \mathcal{O}(n^2)$;*
3. *For second-order cone constraints $g_S(x) = \|A^\top x + p\|_2 - (a^\top x + b) \leq 0$, $C_i \sim \mathcal{O}(n^2)$;*
4. *For general convex constraints g_i, $C_i \sim \mathcal{O}(c_i \log \frac{1}{\epsilon})$ (using bisection), where c_i is the complexity of evaluating g_i given a point.*

The forward integration complexity of our model aligns with standard flow matching approaches, requiring NFE (Number of Function Evaluations) multiplied by the forward computation complexity of the neural network v_θ. The additional reflection or projection operations onto the ball or cube incur negligible overhead with their $\mathcal{O}(n)$ complexity compared to the NN forward calculation. For gauge mapping computation, given that the interior

point is pre-computed offline, the bisection algorithm achieves linear convergence with minimal per-iteration complexity, merely requiring constraint satisfaction verification.

In conclusion, GFM offers three key advantages: (1) it handles general compact convex sets, extending beyond simple set constraints; (2) it maintains bounded Lipschitz constants for bijective gauge mappings between compact sets, with theoretical guarantees on distribution distances; (3) it achieves efficient computation through polynomial-time preprocessing and linear-time online inference for common constraints, with minimal additional cost for general constraints

E EXPERIMENTAL SETTINGS AND RESULTS

In the following subsections, we describe the experimental settings to generate the results reported in section 5. The proposed model is implemented in pytorch (PGM+19), and all models are trained using Adam (KB14) with parameters $\beta_1 = 0.99$, $\beta_2 = 0.999$ and learning rate 10^{-3}. The vanilla interpolating trajectories are solved by (CRBD18) and we utilized (XZY+24) to solve the reflected ODE Equation 4.

E.1 GENERATION UNDER COMMON CONVEX CONSTRAINTS

Baselines: We consider the following baselines:

1. **Vanilla flow matching:** it implements standard flow matching procedures for both training and generation (LGL22; LCBH+22), without any specialized constraint handling mechanisms.
2. **Reflected flow matching:** it extends the basic flow matching framework with reflected generation (XZY+24), leveraging reflection terms for constraint boundaries.
3. **Projected flow matching:** it incorporates projected generation (CBF24), applying orthogonal projections whenever forward trajectories violate the specified constraints.
4. **Vanilla gauge flow matching:** it transforms the data distribution through inverse gauge mapping while maintaining standard integration procedures during generation, without additional constraint processing.
5. **Reflected gauge flow matching:** it combines gauge transformation with reflection terms to ensure trajectories remain within the unit ball through systematic reflection operations, which is the approach adopted in the main body of this work.
6. **Projected gauge flow matching:** it integrates gauge transformation with projection operations to maintain sample trajectories within the unit ball constraint.
7. **Mirror gauge flow matching:** it employs closed-form mirror mapping (LCTT24) during generation to naturally constrain all forward samples within the unit ball.

Problem settings: We test different generative models on some common convex-constrained sets: (1) polytopes $Gx \leq h$ of different dimensions, and (2) quadratic constraints $\frac{1}{2}x^T Qx + p^T x + d \leq 0$. All constrained sets are randomly generated and validated to have a non-empty interior. We also select the interior point for the gauge mapping by solving Equation 28 in Appendix D. Data are sampled from mixed Gaussian distribution, whose location parameter is generated randomly, and covariance is set to $0.3I_d$.

Model settings: We model the velocity field by 2 hidden layers with exponential linear unit (ELU) activation functions. The hidden layers contain 64 units for 2-dimensional and 3-dimensional tasks, 96 units for 6-dimensional tasks and 128 units for 10-dimensional tasks. We train the models for 10 000 epochs with batch size of 32, and prior distribution as the uniform distribution over the constrained domain. Samples are generated by the Euler algorithm in 100 steps, and for the reflected/projected methods, additional reflection/projection is performed after each step.

Table 4: Performance comparison on polytope-constrained generation task.

		Vanilla	Reflected	Projected	Vanilla-GFM	Reflected-GFM	Projected-GFM	Mirror-GFM
$d = 2$	Inference Time (s)	3.230 (0.00%)	5.162 (+59.80%)	4.924 (+52.44%)	3.514 (+8.80%)	**3.398** (+5.20%)	**3.251** (+0.64%)	3.822 (+18.33%)
	KL Divergence (↓)	0.1478	0.1517	0.1685	**0.1371**	**0.1401**	0.1427	2.430
	MMD (↓)	0.03856	0.03869	**0.03866**	0.05700	0.05708	0.05707	**0.3120**
$d = 6$	Inference Time (s)	4.604 (0.00%)	9.992 (+117.03%)	6.922 (+50.35%)	**5.473** (+18.87%)	6.892 (+49.69%)	6.162 (+33.84%)	**4.994** (+8.47%)
	KL Divergence (↓)	0.4523	**0.4670**	**0.4677**	1.1275	1.1275	1.1275	3.5696
	MMD (↓)	0.0324	**0.0323**	**0.0323**	0.0505	0.0505	0.0505	0.1949
$d = 10$	Inference Time (s)	7.756 (0.00%)	12.69 (+63.62%)	8.797 (+13.43%)	**5.493** (−29.17%)	6.854 (−11.63%)	**6.362** (−17.97%)	7.204 (−7.12%)
	KL Divergence (↓)	0.1154	**0.1502**	**0.1539**	0.8872	0.8872	0.8872	11.28
	MMD (↓)	6.505e-3	**6.488e-3**	**6.503e-3**	8.462e-3	8.462e-3	8.462e-3	3.257e-1

[1] We calculate the KL divergences to evaluate the quality of generated samples (XZY+24).
[2] The performance metrics are averaged over 10 runs.
[3] Percentages in the brackets are the relative difference of inference time compared to the vanilla models.

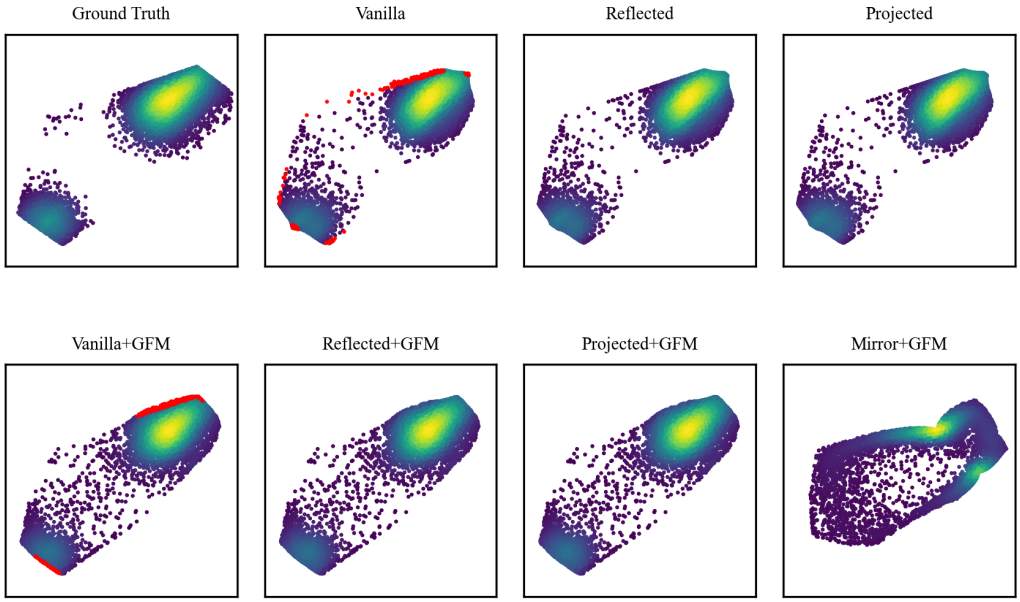


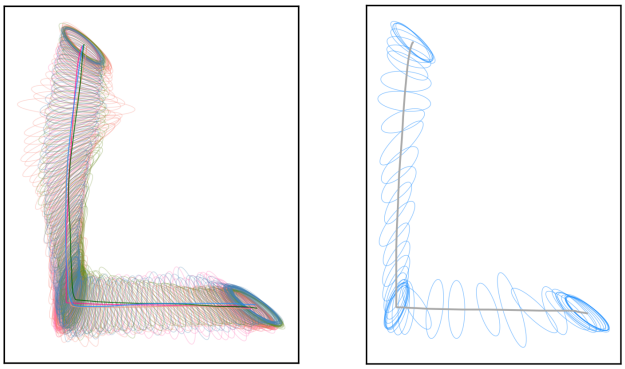
Figure 4: Generated samples by different methods in one 2-dimensional polytope-constrained generation task. From left to right (Data distribution, Vanilla FM, Reflected-GFM, Projected-GFM, and Mirror-GFM.)

Table 5: Performance comparison on 3-dimensional quadratic constrained generation task.

	Vanilla	Reflected	Projected	Vanilla-GFM	Reflected-GFM	Projected-GFM	Mirror-GFM
Inference Time (s)	3.299	10.19	>3600	3.024	3.293	3.083	2.813
KL Divergence (↓)	0.2902	0.3015	-	0.7688	0.7688	0.7688	11.32
MMD (↓)	0.03056	0.03117	-	0.05949	0.05949	0.05949	0.62980

[1] Solving projection onto quadratic constraint set (i.e., convex QCQP) by (DB16) incurs significant computational costs in experiments.

E.2 SPD Matrix Generation



(a) Data (b) GFM

Figure 5: Joint distribution over SPD matrices (demonstrated as manipulation ellipse) and their position for letter L.

Problem settings: We follow the procedure reported by (JRCC21) to learn the trajectories of manipulability ellipse. In planar letter drawing problems, the manipulability ellipse are modeled by SPD matrices $M \in S^2_{++}$,

i.e.,

$$M = x_1 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + x_3 \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \succeq 0, \tag{29}$$

$$\operatorname{tr} M = x_1 + x_3 \leq C. \tag{30}$$

In the original settings, the trajectories are learned by an independent model, which parameterize the trajectory as $p(t) : [0, 1] \to \mathbb{R}^2$. Therefore, we parameterize our models as $\{M \in S^2 \mid M \succeq 0, \operatorname{tr} M \leq C\} \times [0, 1]$.

Model settings: We model the time-variant velocity field by 2 hidden layers with 128 unit each and ELU activation functions. We train the models for 10 000 epochs with batch size of 64, and prior distribution as the uniform distribution over the constrained domain. Samples are generated in 10 steps.

E.3 Ablation Study

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

1. **Choice of norm for unit ball constraint:**
   - ℓ_2-norm ball
   - ℓ_∞-norm cube
2. **Selection of interior point x°:**
   - Central interior point
   - Near-boundary interior point
3. **Unit ball sampling strategies:**
   - Reflection-based sampling
   - Projection-based sampling
   - Mirror map sampling

Model settings: We model the velocity field by 2 hidden layers with 64 units each and ELU activation functions. We train the models for 10 000 epochs with a batch size of 32 and prior distribution as the uniform distribution over the constrained domain. Samples are generated by the Euler algorithm in 100 steps, and for the reflected/projected methods, additional reflection/projection is performed after each step.

Data distribution: For the 2-dimensional example reported in section 5, we set the constrained domain to be

$$\begin{cases} Ax \leq b, \\ \|x - c\|_2 \leq 2.5, \\ x^T Q x + p^T x + d \leq 0, \end{cases} \tag{31}$$

where

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad b = \begin{pmatrix} 2 \\ 1.5 \\ 0 \\ 0 \end{pmatrix}, \qquad c = \begin{pmatrix} 0.3 \\ 0.5 \end{pmatrix},$$

$$Q = \begin{pmatrix} 0.5467 & -0.5600 \\ -0.5600 & 1.3867 \end{pmatrix}, \qquad p = \begin{pmatrix} -0.0427 \\ -0.6880 \end{pmatrix}, \qquad d = -0.8345.$$

Training data are generated from a mix Gaussian distribution

$$p_0 \sim 0.4\mathcal{N}(\mu_1, \Sigma_1) + 0.3\mathcal{N}(\mu_2, \Sigma_2) + 0.4\mathcal{N}(\mu_3, \Sigma_3), \tag{32}$$

where

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \mu_2 = \begin{pmatrix} 1.0 \\ 1.2 \end{pmatrix}, \qquad \mu_3 = \begin{pmatrix} 2.0 \\ 0.6 \end{pmatrix},$$

$$\Sigma_1 = \begin{pmatrix} 0.32 & 0 \\ 0 & 0.08 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 0.15 & 0.30 \\ 0.30 & 0.90 \end{pmatrix}, \qquad \Sigma_3 = \begin{pmatrix} 0.68 & -0.17 \\ -0.17 & 1.19 \end{pmatrix}.$$

We set the interior point for the Gauge map as

$$x_0 = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}. \tag{33}$$

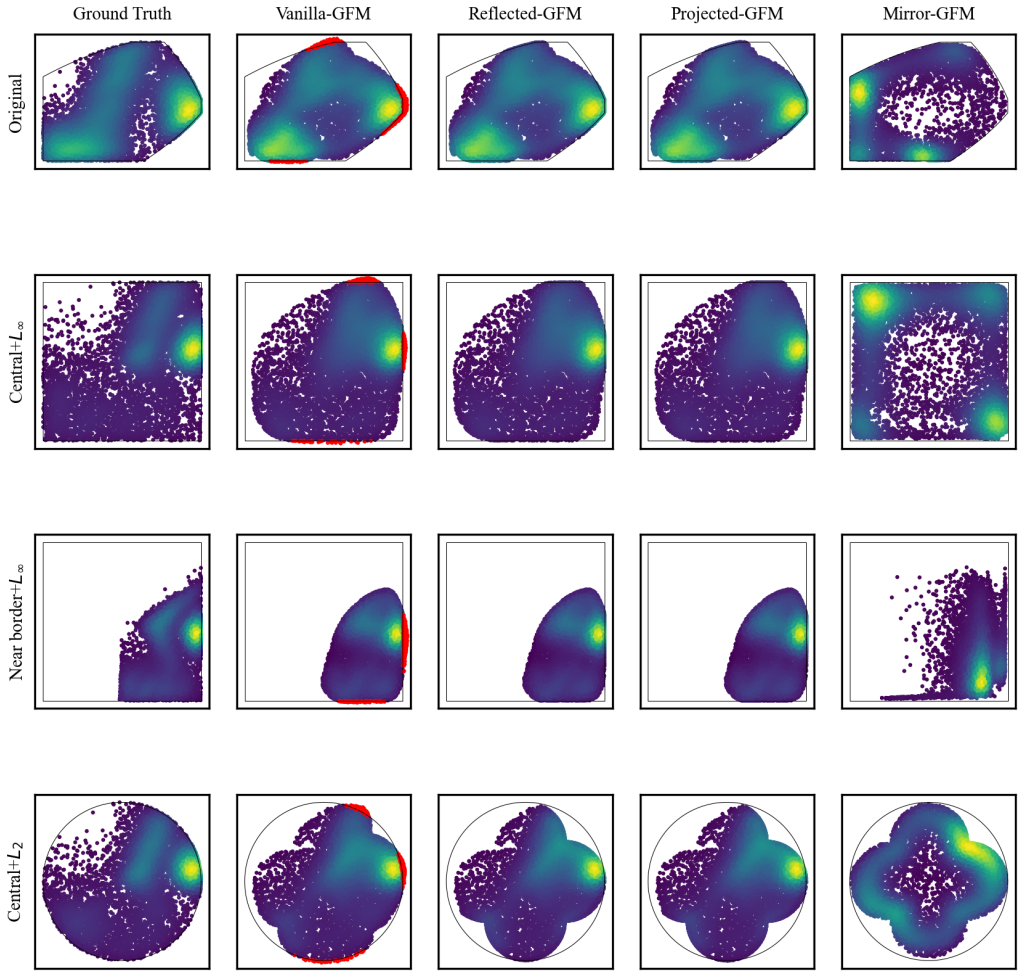


Figure 6: Generated samples by different methods on a 2-dimensional constrained generation task. From left to right (Data distribution, Vanilla-GFM, Reflected-GFM, Projected-GFM, and Mirror-GFM.)

Table 6: Performance comparison on 2-dimensional constrained generation task.

		Vanilla-GFM	Reflected-GFM	Projected-GFM	Mirror-GFM
Central	Inference Time (s)	3.472	3.865	4.116	3.343
+	KL Divergence (↓)	0.1310	0.1307	0.1327	0.8342
L_∞ norm	MMD (↓)	0.02932	0.02936	0.02934	0.09477
Near border	Inference Time (s)	3.694	3.135	4.367	3.594
+	KL Divergence (↓)	0.1849	0.1826	0.1982	1.808
L_∞ norm	MMD (↓)	0.02471	0.02502	0.02496	0.1182
Central	Inference Time (s)	3.754	3.556	2.911	3.465
+	KL Divergence (↓)	0.1310	0.1300	0.1351	1.808
L_2 norm	MMD (↓)	0.02932	0.02942	0.02939	0.1182