



Homeomorphism Methods for Efficient Decision-Making with Hard Constraints

Enming Liang

Research Assistant Professor

Department of Data Science
City University of Hong Kong

2025-12-30

Decision-Making with **Hard** Constraints



Autonomous driving with
mixed traffic flow **constraints**



Power grid with physical
and operational **constraints**



4G/5G parameter optimization
with load balance **constraints**

Constrained Optimization

$$\min_x f(x, \theta)$$

$$\text{s.t. } h(x, \theta) = 0$$

$$g(x, \theta) \leq 0$$

Parameter		θ
Decision		x
Objective		f
Constraints		h, g

Goals: safe, economic, real-time decisions

A Concrete Example in Power Grid Operation

- Minimizing generation cost to serve the load, with an accurate AC model

$$\min \sum_{i \in \mathcal{N}_g} (\lambda_{i,2} P_{gi}^2 + \lambda_{i,1} P_{gi} + \lambda_{i,0}) \quad \xleftarrow{\text{generation cost}}$$

$$\text{s.t. } P_{gi} - P_{di} = \sum_{j:(i,j) \in \mathcal{E}} V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}), \quad \forall i \in \mathcal{N}, \quad \xleftarrow{\text{AC power flow equations}}$$

$$Q_{gi} - Q_{di} = \sum_{j:(i,j) \in \mathcal{E}} V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}), \quad \forall i \in \mathcal{N}, \quad \xleftarrow{\text{AC power flow equations}}$$

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max}, \quad \forall i \in \mathcal{N}_g,$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}, \quad \forall i \in \mathcal{N}_g,$$

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \quad \forall i \in \mathcal{N}_g,$$

$$\theta_{ij}^{\min} \leq \theta_{ij} \leq \theta_{ij}^{\max}, \quad \forall (i, j) \in \mathcal{E},$$

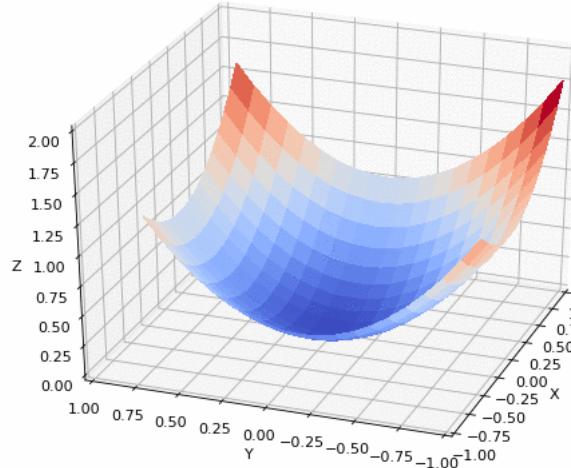
$$0 \leq P_{ij}^2 + Q_{ij}^2 \leq (S_{ij}^{\max})^2, \quad \forall (i, j) \in \mathcal{E},$$

$$P_{ij} = g_{ij} (V_i^2 - V_i V_j \cos \theta_{ij}) - b_{ij} V_i V_j \sin \theta_{ij}, \quad \forall (i, j) \in \mathcal{E}, \quad \xleftarrow{\text{Branch flow limit constraints}}$$

$$Q_{ij} = b_{ij} (-V_i^2 + V_i V_j \cos \theta_{ij}) - g_{ij} V_i V_j \sin \theta_{ij}, \quad \forall (i, j) \in \mathcal{E},$$

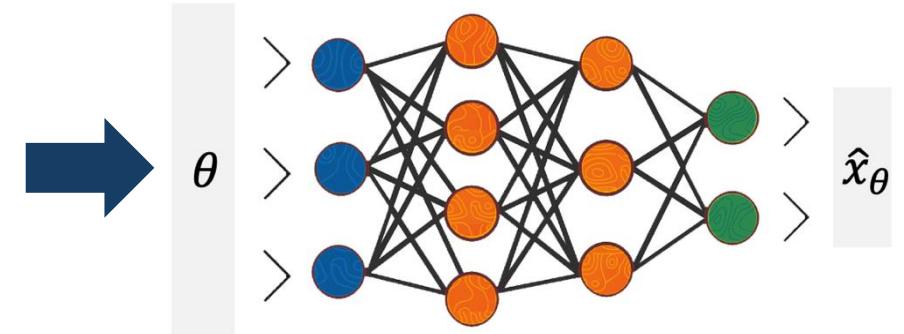
$$\text{var. } P_{gi}, Q_{gi}, \forall i \in \mathcal{N}_g; V_i, \theta_i, \forall i \in \mathcal{N}.$$

Iterative Algorithm vs Machine Learning



$$\begin{aligned} & \min_x f(x, \theta) \\ \text{s.t.. } & h(x, \theta) = 0 \\ & g(x, \theta) \leq 0 \end{aligned}$$

θ : parameter | x : decisions



Given θ , update iteratively to pursue optimal solution x_θ^*

- $x_\theta^{n+1} = x_\theta^n - g(x_\theta^n)$
- E.g., interior point methods, Gurobi, ...

Feasible, optimal, but **slow**

Training neural network to predict solutions end-to-end

- $\hat{x}_\theta = F_{nn}(\theta)$
- E.g., supervised training $\min_{F_{nn}} \sum_i \|F_{nn}(\theta_i) - x_i^*\|$

Fast, near-optimal, but **non-feasible**

Homeomorphism Methods

Motivation: transform **complex** constrained problem into **simple** constraint domain (e.g., ball)



Homeomorphism

- *Bijective & Bi-continuous* mapping
- Preserve topological structures

Our contributions

Accelerate iterative algorithms [1]

- The **first** work achieving **optimal convergence** rates without expensive oracles over **general convex sets**



Ensure neural network feasibility [2,3]

- The **first** work ensuring NN solution feasibility with bounded opt. gap and **low-complexity** over a class of **non-convex sets**



Goals: safe, economic, rea-time decisions

[1] C. Liu, E. Liang*, M. Chen*, “Fast Projection-Free Algorithm (without Optimization Oracles) for Optimization over General Convex Set”. NeurIPS 2025. **Spotlight**.

[2] E. Liang, M. Chen, S. Low, “Low Complexity Homeomorphic Projection to Ensure NN Solution Feasibility for Optimization over (Non-)Convex Set”, ICML. 2023.

[3] E. Liang, M. Chen, S. Low, “Homeomorphic Projection to Ensure NN Solution Feasibility for Constrained Optimization”. JMLR. 2024.

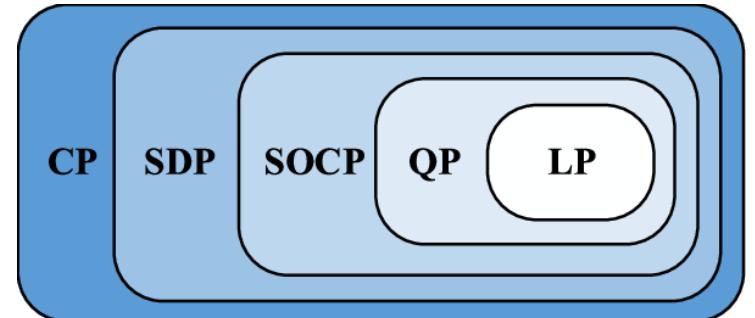
Homeomorphism Methods: Part I

Hom-PGD to Accelerate Iterative Algorithms

Key Metrics for Iterative Algorithms

Applicable Scenarios:

- Objective: linear, quadratic, convex,...
- Constraints: polytope, convex cones,...

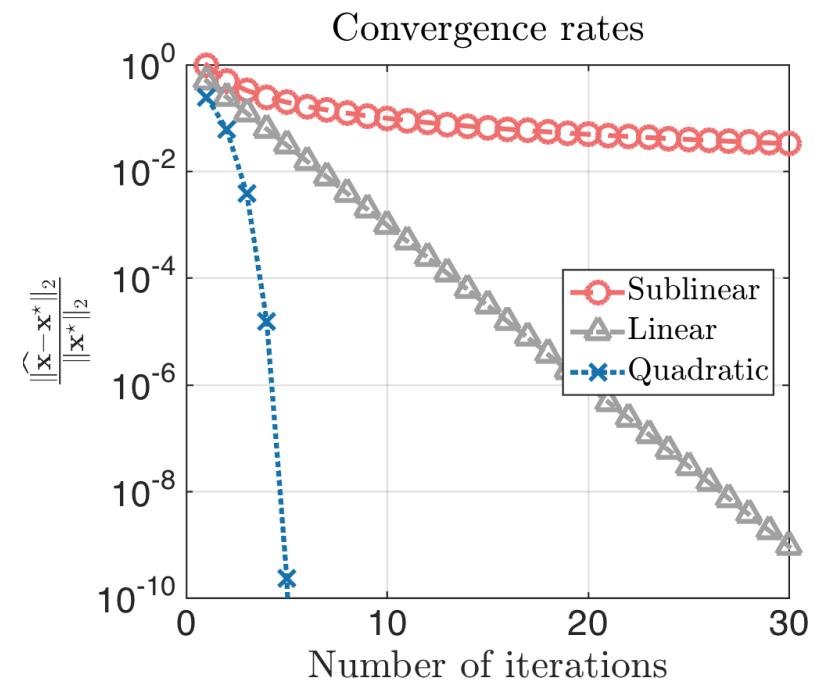


Convergence rate: num. of steps to optimum

- Sublinear: $N = O(1/\varepsilon)$
- Linear: $N = O(\log 1/\varepsilon)$

Per-step cost: $x_\theta^{n+1} = x_\theta^n - g(x_\theta^n)$

- Memory cost: store medium variables.
- Computational cost: $O(n^2), O(n^3)$



Existing Iterative Algorithms

Second-order methods (*Hessian*)

- Primal-Dual Interior Points (in MOSEK)

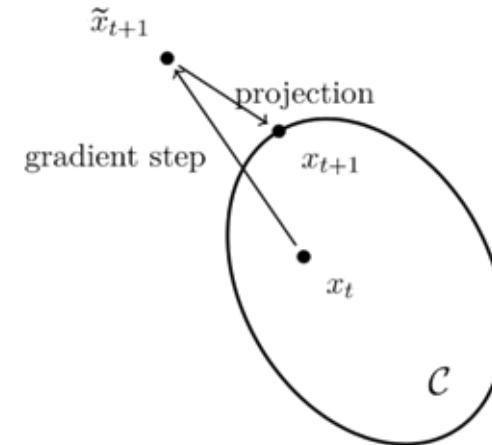
High memory & per-step cost

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \text{ and } \nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

First-order methods (*Gradient*)

- Projection Gradient Descent (PGD)
- Projection-free Frank-Wolfe (FW)
- Augmented Lagrangian (e.g., ADMM)

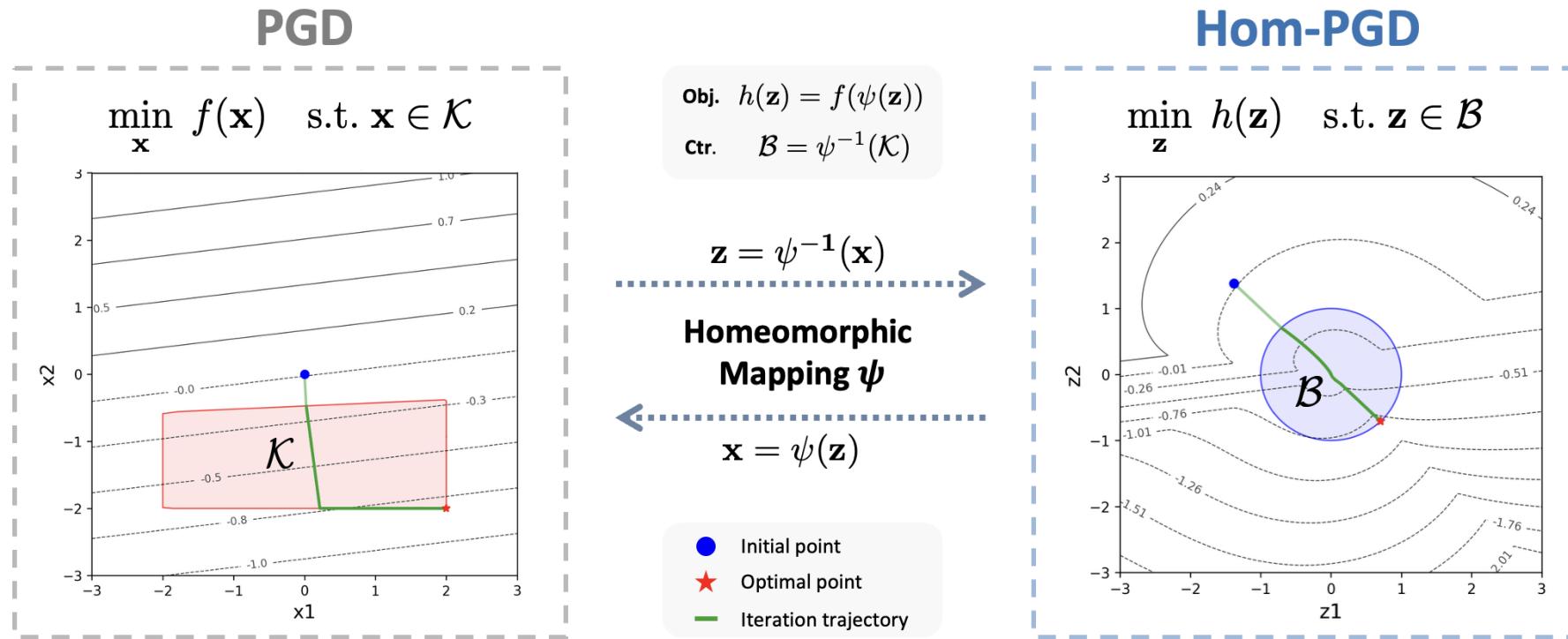
Non-trivial sub-problems



$$x_{t+1} = \Pi_C(\tilde{x}_{t+1}) \\ := \arg \min_{x \in C} \frac{1}{2} \|x - \tilde{x}_{t+1}\|_2^2$$

Research Gap: design an algorithm that with **low per-iter costs**, matching **optimal convergence**, working for **general convex set**

Hom-PGD to Accelerate Iterative Algorithms

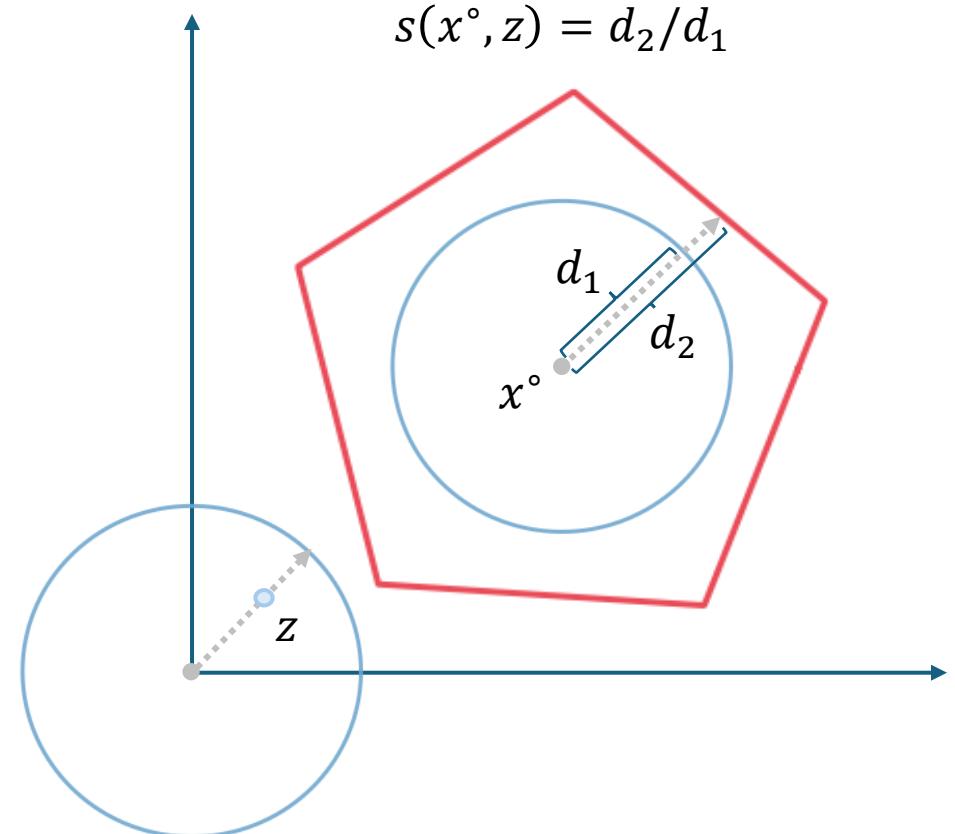


1. Construct homeomorphism for general convex set
2. Transform the problem into ball-constrained space
3. Projection gradient descent over ball is easy

Gauge Mapping: Homeomorphism for Convex Set

$$\Phi(z) = x^\circ + s(x^\circ, z) \cdot z$$

- Translating by $x^\circ \in \text{int}(\mathcal{C})$
- Scaling by $s(x^\circ, z) \in \mathbb{R}_+$
- Properties of Φ :
 - Continuous + Invertible \rightarrow Homeomorphism
 - Bi-Lipschitz \rightarrow Beneficial for algorithm [4]
- Computation of Φ :
 - **Closed-form** for common convex sets [1-2]
 - ✓ Linear, quadratic, SOC, LMI.
 - **Bisection** for general compact convex set [3]



[1] Tabas, D., & Zhang, B. Computationally efficient safe reinforcement learning for power systems. IEEE ACC 2022.

[2] Tordesillas, J., How, J. P., & Hutter, M. Rayen: Imposition of hard convex constraints on neural networks. arXiv 2023.

[3] Mhammedi, Z. Efficient projection-free online convex optimization with membership oracle. COLT 2022.

[4] C. Liu, E. Liang*, M. Chen*, "Fast Projection-Free Algorithm (without Optimization Oracles) for Optimization over General Convex Set". NeurIPS 2025. **Spotlight**.

Computation of Gauge Mapping

Constraints	Formulation	Inverse Distance Function
Intersections	$\{g_1(x) \leq 0, \dots, g_m(x) \leq 0\}$	$\kappa_g(x^\circ, v) = \max_{1 \leq i \leq m} \{\kappa_{g_i}(x^\circ, v)\}$
Linear	$g_L(x) = a^\top x - b \leq 0$	$\kappa_{g_L}(x^\circ, v) = \left\{ \frac{a^\top v}{b - a^\top x^\circ} \right\}^+$
Quadratic	$g_Q(x) = x^\top Qx + a^\top x - b \leq 0$	$\kappa_{g_Q}(x^\circ, v) = \{1/\text{root}(A_Q, B_Q, C_Q)\}^+$
Second Order Cone	$g_S(x) = \ A^\top x + p\ _2 - (a^\top x + b) \leq 0$	$\kappa_{g_S}(x^\circ, v) = \{1/\text{root}(A_S, B_S, C_S)\}^+$
Linear Matrix Inequality	$g_M(x) = \sum_{i=1}^n x_i \cdot F_i + F_0 \succeq 0$	$\kappa_{g_M}(x^\circ, v) = \{\text{eig}(L^\top(-S)L)\}^+$

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

² $(\cdot)^+ = \max(\cdot, 0)$

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

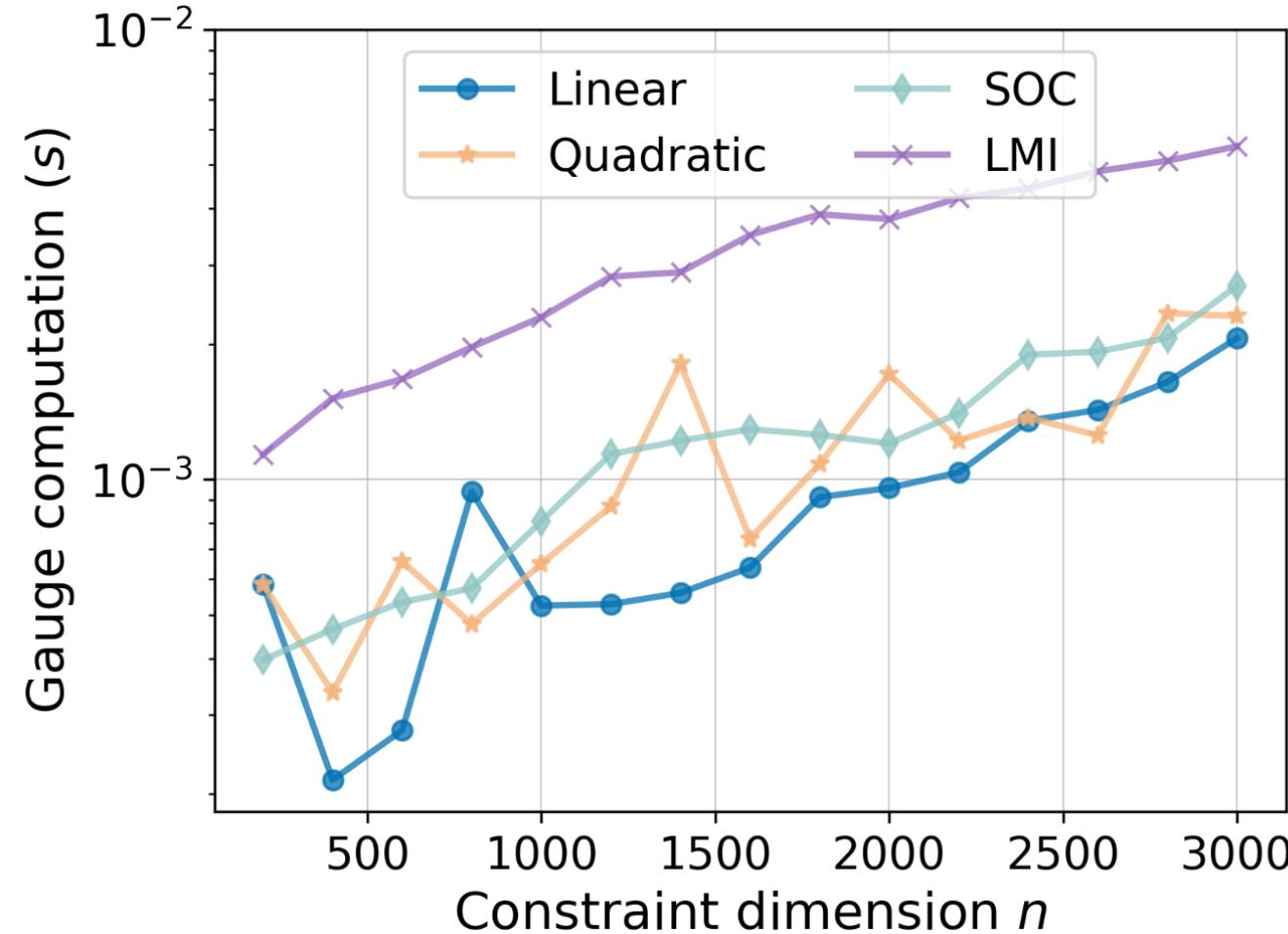
⁴ $\text{eig}(X) = \lambda_1, \dots, \lambda_n$ denotes the eigenvalues satisfying $\det(X - \lambda I) = 0$

⁵ $A_Q = v^\top Qv, B_Q = 2x^\circ \top Qv + a^\top v, C_Q = x^\circ \top Qx^\circ + a^\top x^\circ - b$

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

Computation of Gauge Mapping



Landscape Analysis and Algorithm Design

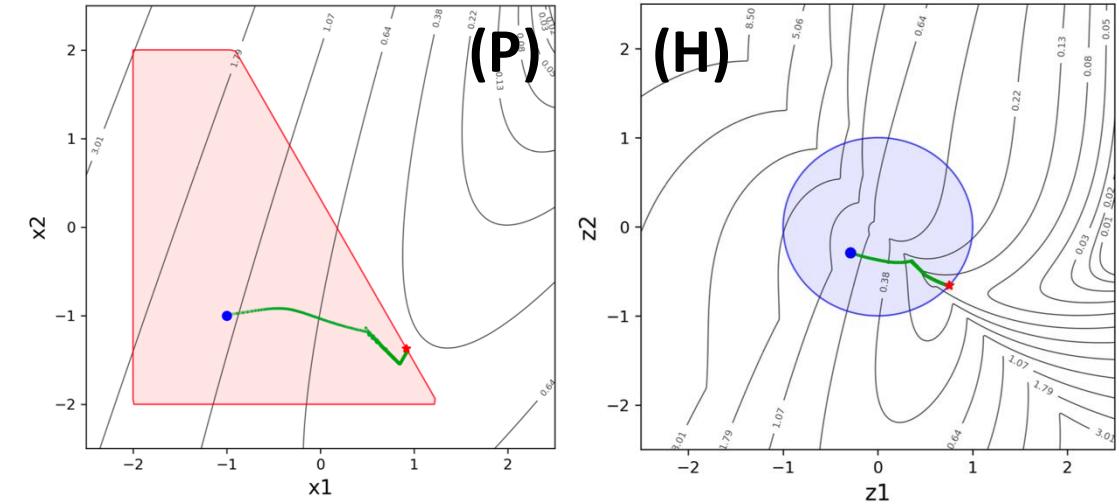
- Original Problem: **(P)**

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{K}$$

$$\mathcal{B} = \psi^{-1}(\mathcal{K})$$
$$h(\mathbf{z}) = f(\psi(\mathbf{z}))$$

- Transformed Problem: **(H)**

$$\min_{\mathbf{z}} h(\mathbf{z}) \quad \text{s.t. } \mathbf{z} \in \mathcal{B}$$



Equivalence between **(P)** and **(H)**

Theorem 1: For convex **(P)** with global optimum x^* , under mild constraint qualification conditions, any stationary point \mathbf{z}^* for **(H)** is

- a global optimum of **(H)**: $h(\mathbf{z}^*) \leq h(\mathbf{z})$
- a global optimum of **(P)**: $x^* = \psi(\mathbf{z}^*)$

Algorithm 1 Hom-PGD

Input: initial point \mathbf{z}_0 , problem **H** with ψ and maximum iteration number K
for $k = 0$ **to** K **do**
 Compute stepsize α_k
 Update: $\mathbf{z}_{k+1} = \Pi_{\mathcal{B}} (\mathbf{z}_k - \alpha_k \nabla h(\mathbf{z}_k))$
end for
Output: $\mathbf{x}_K = \psi(\mathbf{z}_K)$

Convergence Rates Analysis : Part I

- Convergence rates [1]:

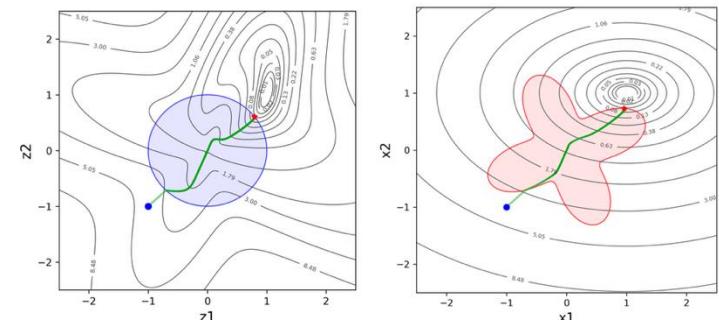
Theorem 2: Under mild regularity conditions, for compact convex constraint, **Hom-PGD** over **(H)** match the same convergence rates of **PGD** over **(P)**:

Objective of (P)		Strongly Convex		Convex		Non-Convex
Convergence Rate		$\mathcal{O}(\log \epsilon^{-1})$		$\mathcal{O}(\epsilon^{-1})$		$\mathcal{O}(\epsilon^{-2})$

- Reach the **optimal** convergence rates under unaccelerated settings [2]
- Per-iteration cost of $\mathcal{O}(n^2)$, without expensive inner optimization

- **Extension** to structured non-convex settings

- Proximal-PL objective [3]: $\mathcal{O}(\log \epsilon^{-1})$ rate (match PGD)
- Star-convex constraints [4]: $\mathcal{O}(\epsilon^{-2})$ rate (beyond PGD)



[1] C. Liu, E. Liang*, M. Chen*, “Fast Projection-Free Algorithm (without Optimization Oracles) for Optimization over General Convex Set”. NeurIPS 2025. **Spotlight**.

[2] Altschuler JM, Parrilo PA. Acceleration by stepsize hedging: Multi-step descent and the silver stepsize schedule. Journal of the ACM. 2025.

[3] Karimi H, Nutini J, Schmidt M. Linear convergence of gradient and proximal-gradient methods under the polyak-Łojasiewicz condition. Springer. 2016.

[4] Lee JM. Introduction to topological manifolds. New York, NY: Springer New York; 2000 May 25.

Convergence Rates Analysis: Part II

- Remarks on convergence rates [1]:
 - Under mild conditions, Hom-PGD with **any homeomorphism** can reach the **same order** of convergence rates
 - The **Bi-Lipschitz constants** of homeomorphism impact the **hidden constants** in the convergence rates
- Bi-Lipschitz constants of Gauge Mapping

Proposition 3.3 (Bi-Lipschitz Constants of the Gauge Mapping). *Let $\mathcal{K} \subset \mathbb{R}^n$ be a compact convex set and let $\mathbf{x}^\circ \in \text{int}(\mathcal{K})$ be an interior point. Define the inner and outer radii with respect to \mathbf{x}° as*

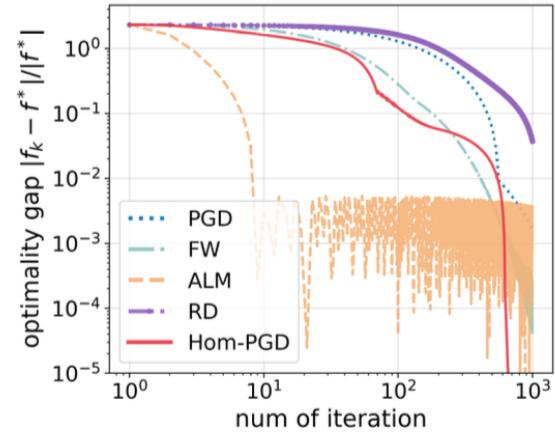
$$r_i := \sup\{r \geq 0 : \mathcal{B}(\mathbf{x}^\circ, r) \subseteq \mathcal{K}\}, \quad r_o := \inf\{r \geq 0 : \mathcal{K} \subseteq \mathcal{B}(\mathbf{x}^\circ, r)\},$$

such that $\mathcal{B}(\mathbf{x}^\circ, r_i) \subseteq \mathcal{K} \subseteq \mathcal{B}(\mathbf{x}^\circ, r_o)$. Then the Lipschitz constant (denoted as $L(\cdot)$) of gauge mapping ψ associated with \mathcal{K} satisfies the following bounds:

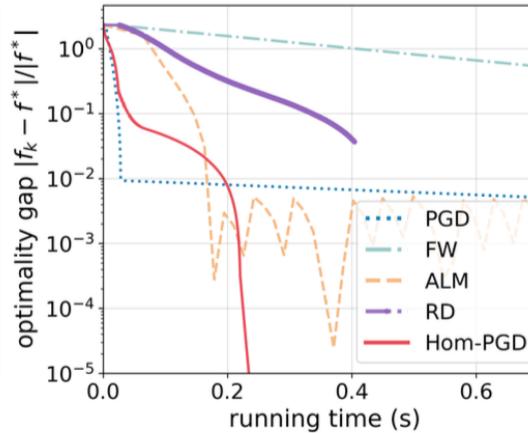
$$\text{Forward Lipschitz: } \kappa_2 := L(\psi) \leq 2r_o + r_o^2/r_i, \quad \text{Inverse Lipschitz: } \frac{1}{\kappa_1} := L(\psi^{-1}) \leq 2/r_i.$$

Central interior point → **Smaller** Lipschitz → **Faster** Convergence

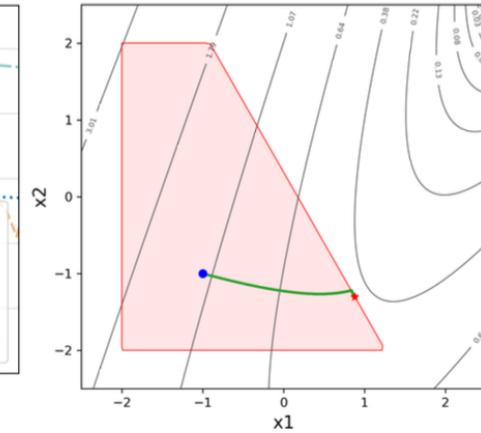
Illustrative Examples: Convex & Star-convex Cases



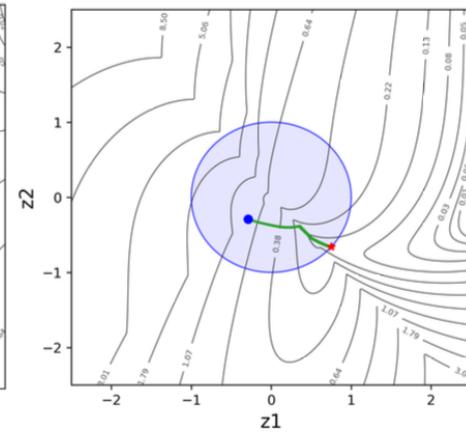
(a) convergence rate



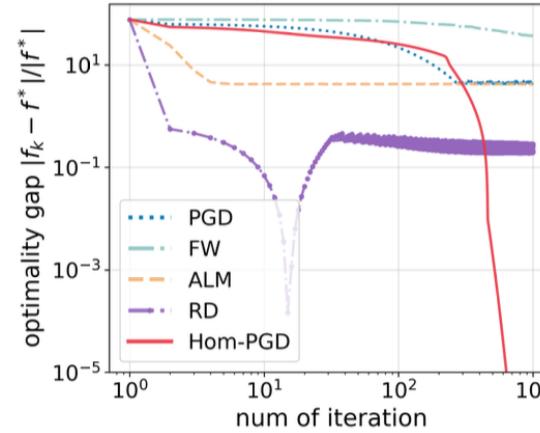
(b) running time (s)



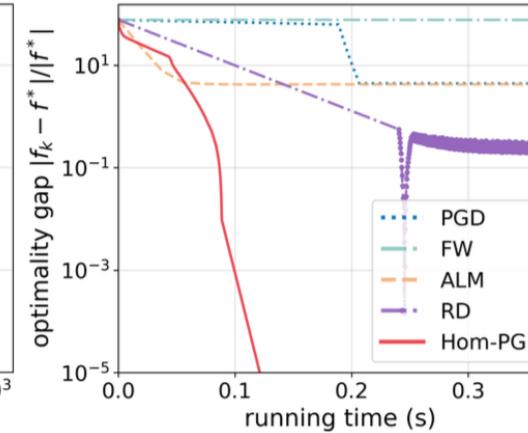
(c) PGD iteration



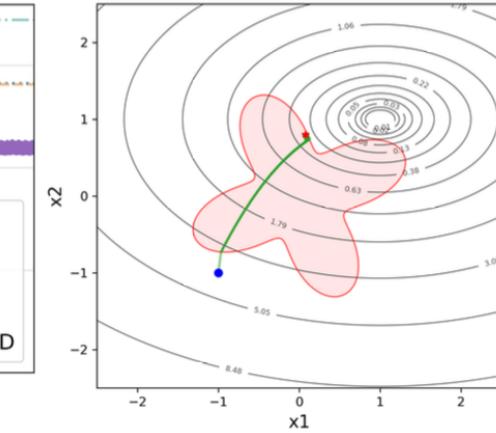
(d) Hom-PGD iteration



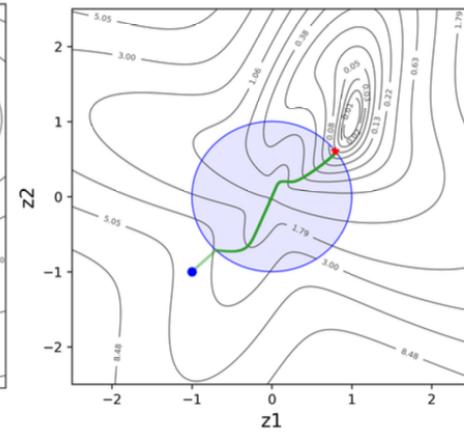
(e) convergence rate



(f) running time (s)



(g) PGD iteration

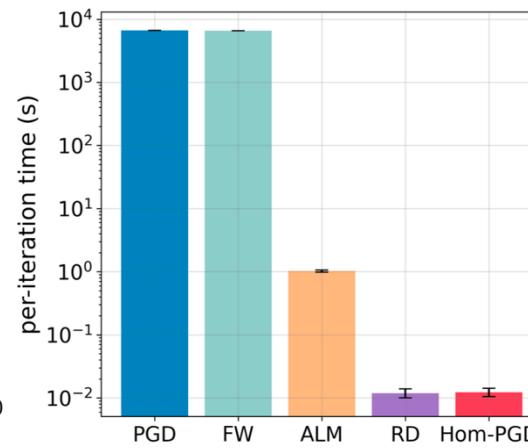
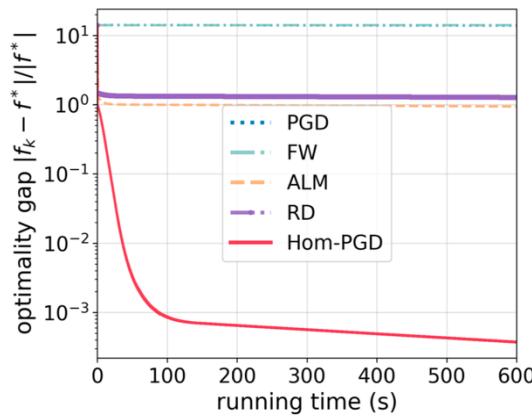


(h) Hom-PGD iteration

1000+ Dim. Convex Programs: SOCP & SDP

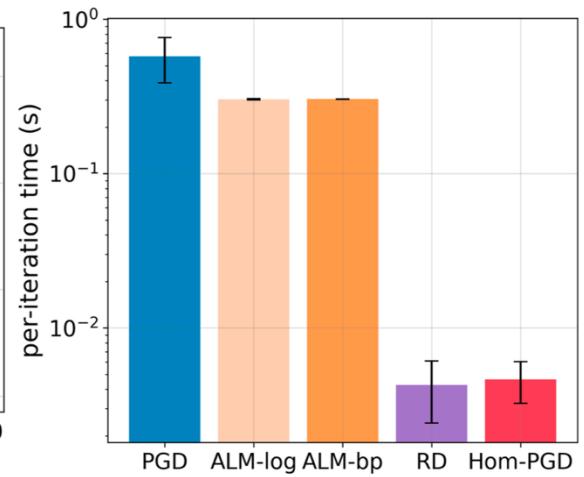
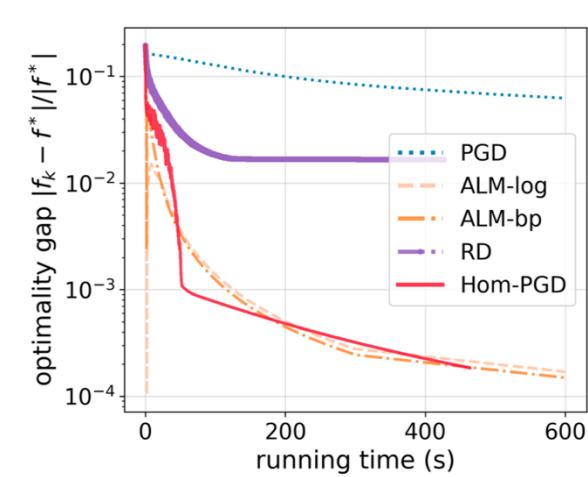
SOCP

$$\begin{aligned} \min_{\mathbf{L} \leq \mathbf{x} \leq \mathbf{U}} \quad & \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{p}^\top \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{G}_i \mathbf{x} + \mathbf{h}_i\|_2 \leq \mathbf{c}_i^\top \mathbf{x} + d_i, \quad i \in [n_{\text{soc}}] \end{aligned}$$



Max-Cut SDP

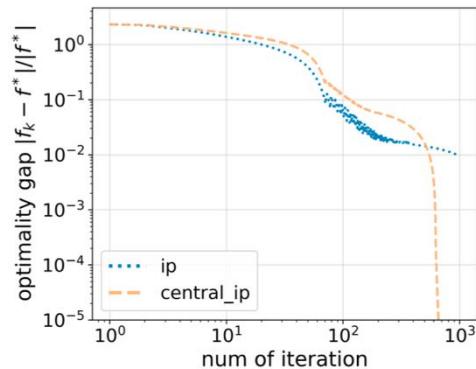
$$\begin{aligned} \max_{-\mathbf{1} \leq \mathbf{X} \leq \mathbf{1}} \quad & \sum_{(i,j) \in \mathcal{E}} (1 - x_{ij})/2 \\ \text{s.t.} \quad & x_{ii} = 1, \quad i \in [n] \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$



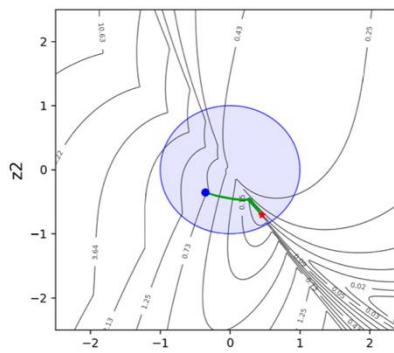
2-4 order reduction on per-iter. cost, **faster** convergence to optimum,
per-step **feasibility** guarantees

Ablation Study

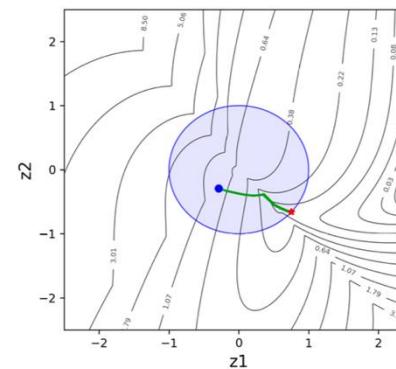
- Impacts of gauge mapping: $\Phi(z) = x^\circ + s(x^\circ, z) \cdot z$



(a) Convergence comparison.



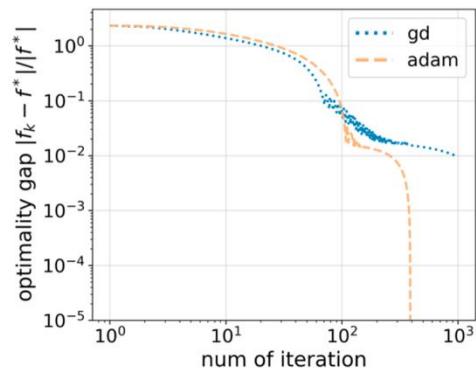
(b) Near-Boundary IP



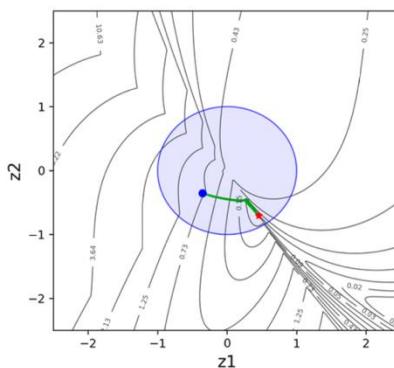
(c) "Central" IP

Central interior point →
Smaller Lipschitz →
Faster Convergence

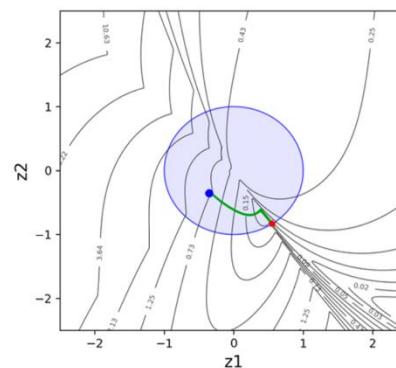
- Impacts of gradient methods: GD vs Adam



(a) Convergence comparison.



(b) Gradient descent



(c) Adam [KB14]

Adaptive/Accelerated
methods excel in non-
convex landscapes

Takeaways: Hom-PGD

- **Idea:**

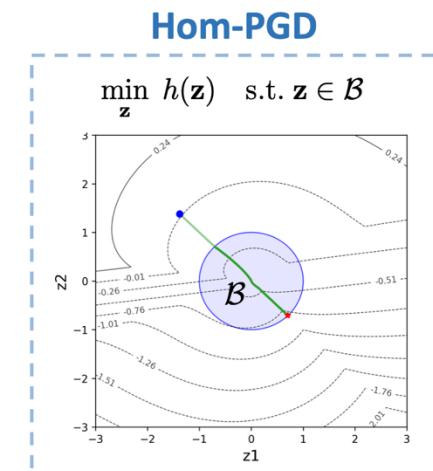
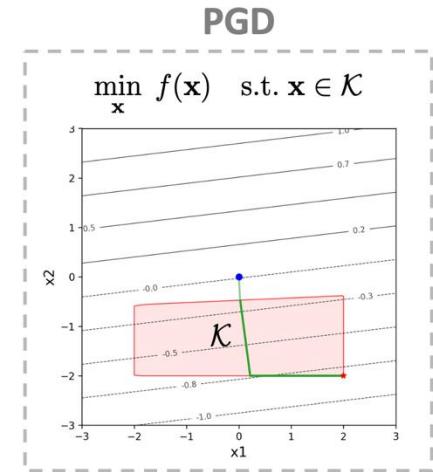
- Transform complex constrained problems into simple ones via homeomorphism.

- **Contributions:**

- Construct **explicit-form homeomorphism** for any compact convex set via **gauge mapping**
- Hom-PGD as an algorithm that with **low per-iter costs**, matching **optimal convergence**

- **Future works:**

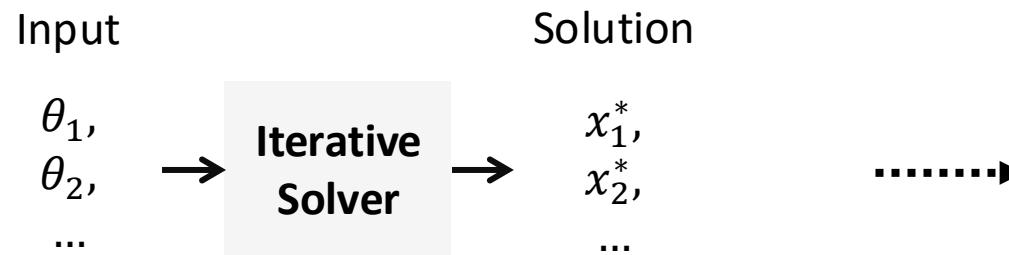
- Advanced gradient methods (Nesterov, momentum, Adam)
- Extend to more general non-convex constraints [2]



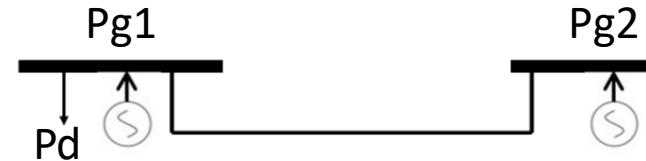
Homeomorphism Methods: Part II

Hom-Proj. to Ensure Neural Network Feasibility

Input-Solution Mapping for Optimization Problems

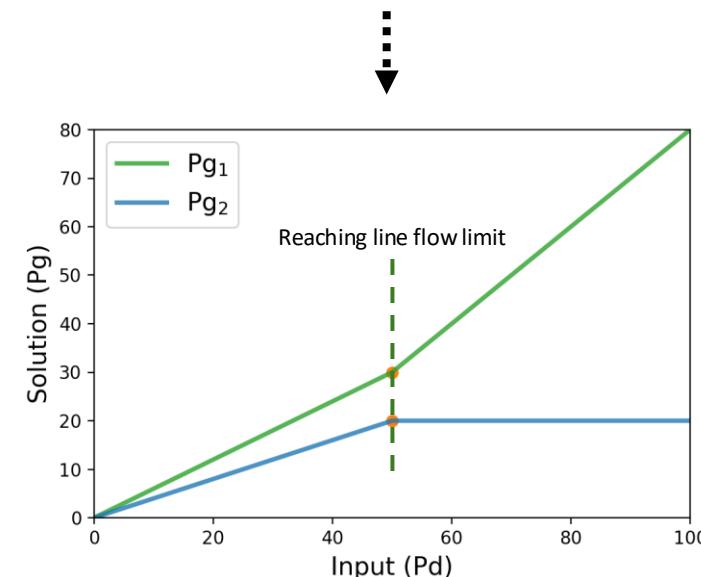


Optimal power flow problem



Pd: demand; Pg: generation

- Learn the mapping between input and solution
 - **Continuous** (a.e.) mapping for optimization with continuous objective and constraints [1-4]



[1] Dontchev, A. L., & Rockafellar, R. T. *Implicit functions and solution mappings* (Vol. 543). New York: Springer. 2009

[2] P. Tøndel, T. A. Johansen and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 2003

[3] X. Pan, T. Zhao, M. Chen, and S. Zhang. DeepOPF: A deep neural network approach for security-constrained dc optimal power flow. *IEEE Trans. Pow. Sys.* 2019

[4] X. Pan, M. Chen, T. Zhao, S. H. Low. DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. *IEEE Sys. J.* 2022

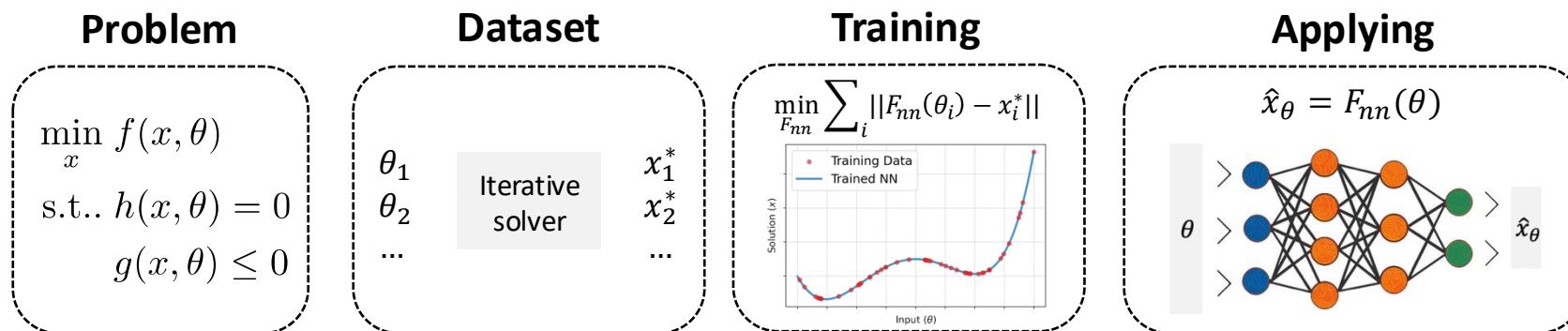
Machine Learning Approach

- Use NN to learn the input-to-solution mapping

➤ **Why:** Universal Approximation Theorem [1-5]

NNs (e.g., FCNN, CNN, ResNet, Transformer, ...) can approximate “any” continuous mapping arbitrarily well with sufficient parameters.

➤ **How:** e.g., supervised learning



- NN can predict near-optimal solutions in real-time

[1] K. Hornik, “Approximation capabilities of multilayer feed forward networks,” *Neural networks*, vol.4, no.2, pp. 251–257, 1991.

[2] G. Cybenko, “Approximation by superpositions of a sigmoidal function”, *Math. Control Signals Systems*, 2(4):303–314, 1989.

[3] Pinkus, Allan. "Approximation theory of the MLP model in neural networks." *Acta numerica*, 8 :143-195, 1999.

[4] Zhou, D. X. Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48(2), 787-794. 2020.

[5] Lin, H., & Jegelka, S. Resnet with one-neuron hidden layers is a universal approximator. *NeurIPS* 2018.

Progress and Significant Results

- Algorithmic design

- Unsupervised, self-supervised, GNN[1-3]

- Theoretical advance

- Approximation capability of deep NN & ResNet [4-6]
 - Universal approximation of GNN [7]

- Real-world applications

- Power grid [8-9], wireless network [10], PDE solvers [11],...

[1] Park, S., & Van Hentenryck, P. Self-supervised primal-dual learning for constrained optimization. AAAI. 2023

[2] Huang, W., Chen, M., & Low, S. H. Unsupervised Learning for Solving AC Optimal Power Flows: Design, Analysis, and Experiment. IEEE Trans. Power Syst. 2024

[3] Owerko, D., Gama, F., & Ribeiro, A. Optimal power flow using graph neural networks. IEEE ICASSP. 2020.

[4] Yarotsky, D. Error bounds for approximations with deep ReLU networks. Neural networks, 94, 103-114. 2017

[5] Liang, S., & Srikant, R. Why deep neural networks for function approximation. ICLR. 2017

[6] Liu, C., Liang, E., & Chen, M. Characterizing ResNet's Universal Approximation Capability. ICML. 2024

[7] Chen Z, Liu J, Wang X, Lu J, Yin W. On representing linear programs by graph neural networks. ICLR 2023.

[8] X. Pan, T. Zhao, M. Chen, and S. Zhang. DeepOPF: A deep neural network approach for security-constrained dc optimal power flow. IEEE Trans. Pow. Sys. 2019

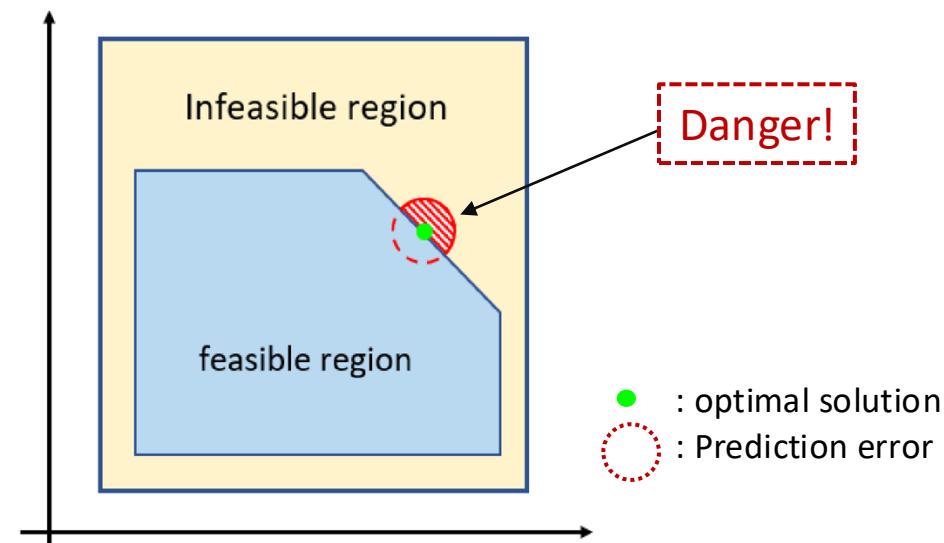
[9] Pan, X., Chen, M., Zhao, T., & Low, S. H. DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. IEEE Syst. J. 2022

[10] Shen, Y., Zhang, J., Song, S. H., & Letaief, K. B. Graph neural networks for wireless communications: From theory to practice. IEEE Trans. Wireless Commun. 2022

[11] Greenfeld D, Galun M, Basri R, Yavneh I, Kimmel R. Learning to optimize multigrid PDE solvers. ICML 2019.

Challenge: Ensuring Neural Network Feasibility

- Feasible solution is crucial for safety-critical systems
 - E.g., in power grid operation, violating line capacity limits can cause grid failure

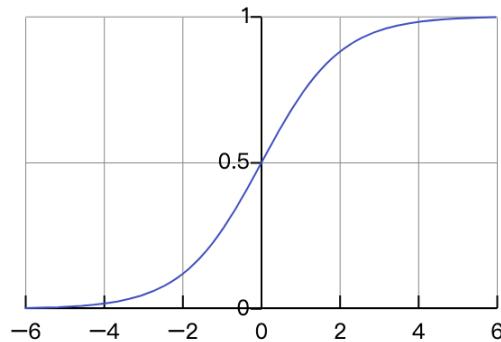


- Hard to guarantee NN feasibility due to prediction errors

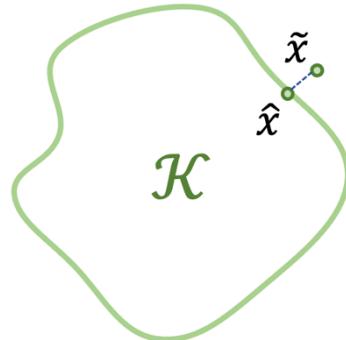
Existing Works on Neural Network Feasibility

Existing Work	Constraint Setting	Feasibility Ensuring	Optimality Bound	Low Run-time
Activation layer	Simplex/Box	✓	✓	✓
Penalty/Lagrangian	General	✗	✗	✓
Orthogonal projection	General	✓	✓	✗
Sampling approach	General	✓	✓	✗
Preventive learning	Linear	✓	✗	✓
Gauge/RAYEN	Convex	✓	✓	✓

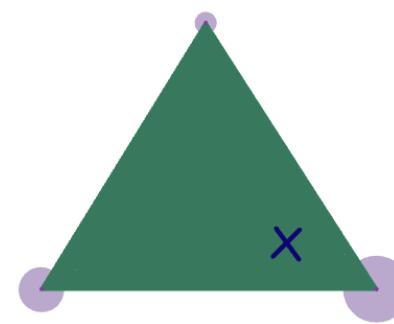
Sigmoid activation



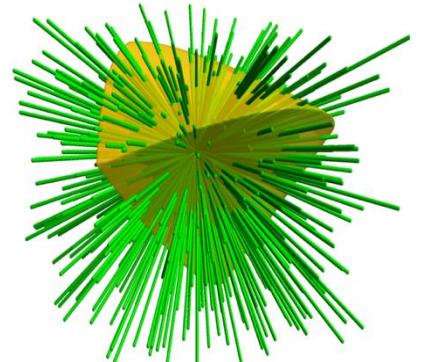
l_2 projection



Sampling-based



RAYEN



Previous works either **lack** performance guarantees, or applicable to **limited** constraint sets, or **slow** in run-time

Motivation and Ball-Homeomorphism

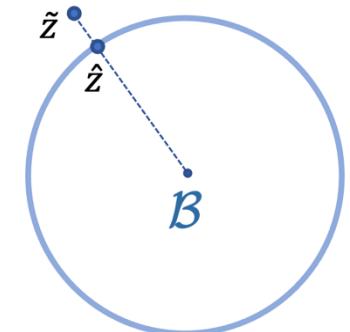
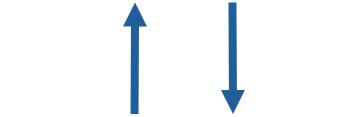
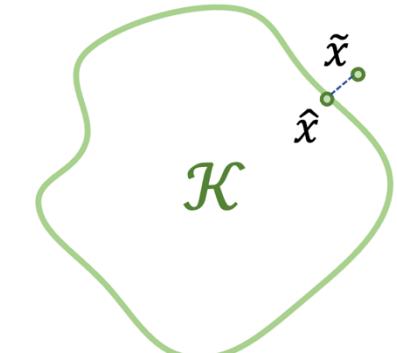
- NN predicts **near-optimal** but **infeasible** solutions
- Solve projection for feasibility
 - over a non-convex set: **hard**
 - over a ball: **easy**

$$\begin{aligned}x_{t+1} &= \Pi_{\mathcal{K}}(\tilde{x}_{t+1}) \\&:= \arg \min_{x \in \mathcal{K}} \frac{1}{2} \|x - \tilde{x}_{t+1}\|_2^2\end{aligned}$$



Transform the hard projection to easy projection over ball

- Homeomorphism:
 - *Bijective & Bi-continuous* mapping
 - Preserve topological structures
- **Ball-homeomorphic** constraint set ?

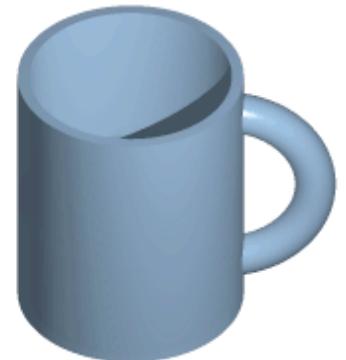


Ball-Homeomorphic Constraint Set

- All compact (star-)convex sets [1]
- For compact and contractible manifold [2]
 - in 6 (or higher)-dim space and its boundary is **simply connected**
 - in 5-dim space and its boundary is **diffeomorphic** to a **4-dim sphere**
- All open simply-connected sets in 2-dimension space [3]



Ball-Homeomorphic set a **general** class of **non-convex** set beyond convex and star-convex set

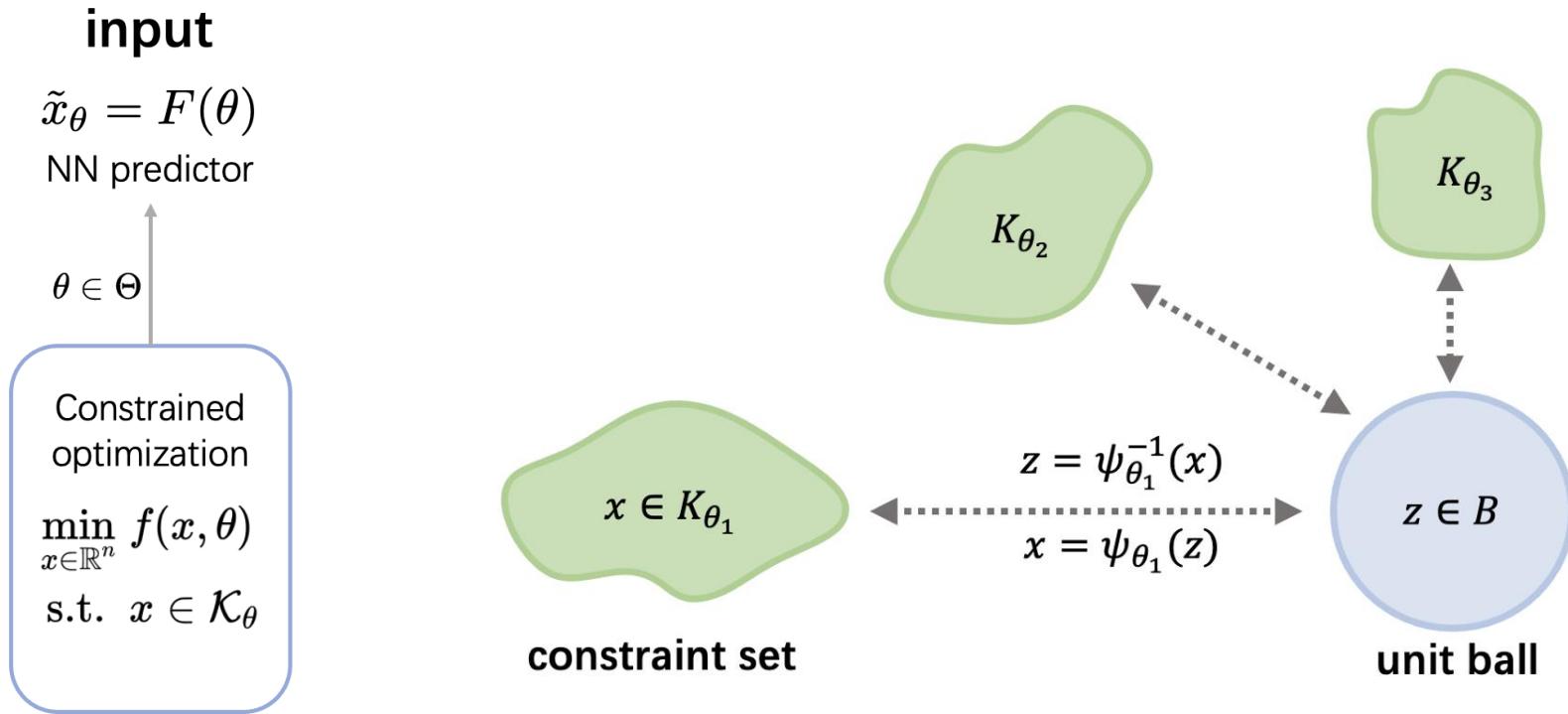


[1] Geschke, S. (2012). Convex open subsets of R^n are homeomorphic to n -dimensional open balls. Hausdorff Center for Mathematic.

[2] Smale, S. (1962). On the structure of manifolds. American Journal of Mathematics, 84(3), 387-399 (**Theorem 5.1**)

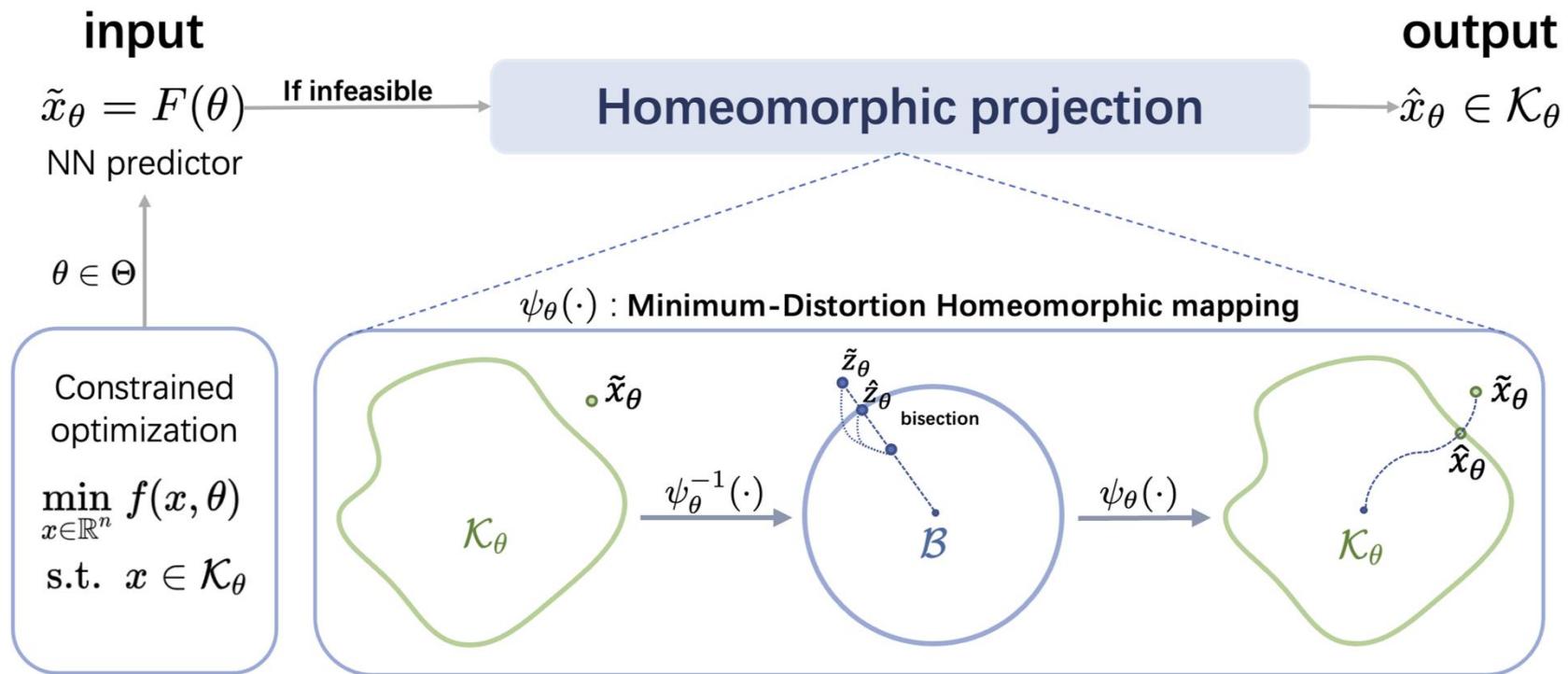
[3] Riemann Mapping Theorem

Homeomorphic Projection



Setting: recover NN solution feasibility over ball-homeomorphic sets

Homeomorphic Projection



1. Learn homeomorphism for a class of non-convex set
2. Transform the hard projection problem into ball space
3. Perform bisection over the ball for feasible solution

Distortion and Homeomorphism

Distortion: $D(\psi) = k_2/k_1 \geq 1$

$$k_2 = \sup_{z_1, z_2} \left\{ \frac{\|\psi(z_1) - \psi(z_2)\|}{\|z_1 - z_2\|} \right\}, \quad k_1 = \inf_{z_1, z_2} \left\{ \frac{\|\psi(z_1) - \psi(z_2)\|}{\|z_1 - z_2\|} \right\}$$

- Ratio of max & min distance **variations** by ψ
- **Multiple** homeomorphic mappings between two sets

Prop. 1. Let ψ be a homeomorphic mapping between \mathcal{B} and \mathcal{K} , the homeomorphic projection as:

$$\text{HP}_{\mathcal{K}}(x) = \psi(\Pi_{\mathcal{B}}(\psi^{-1}(x)))$$

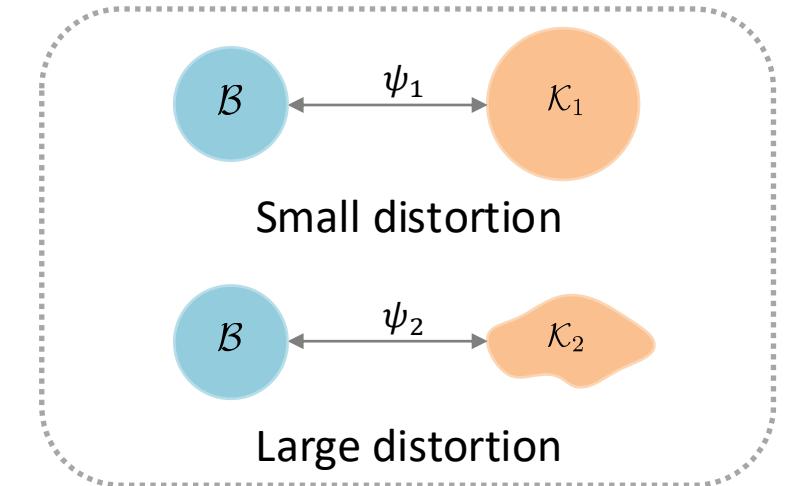
Then the projection distance is bounded as:

$$\|\Pi_{\mathcal{K}}(x) - x\| \leq \|\text{HP}_{\mathcal{K}}(x) - x\| \leq D(\psi) \cdot \|\Pi_{\mathcal{K}}(x) - x\|$$

Orthogonal
projection distance

Homeomorphic
projection distance

Distortion of
Homeomorphism



**Minimum Distortion
Homeomorphism (MDH)**

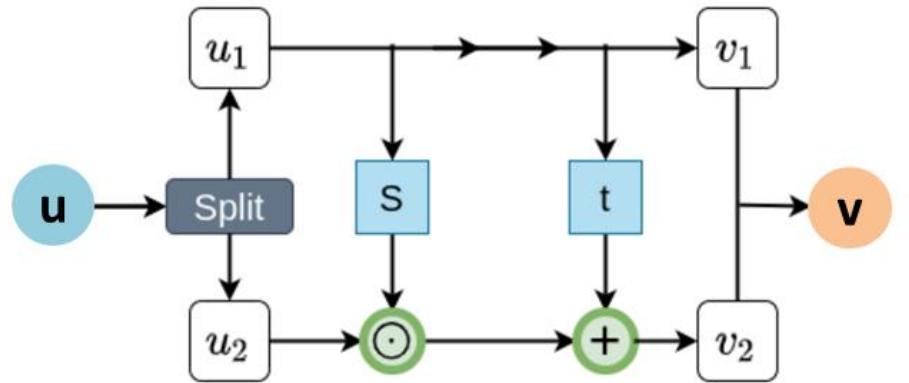
$$\min_{\psi_\theta \in \mathcal{H}^n} \log D(\psi_\theta)$$

s.t. $\mathcal{K}_\theta = \psi_\theta(\mathcal{B})$

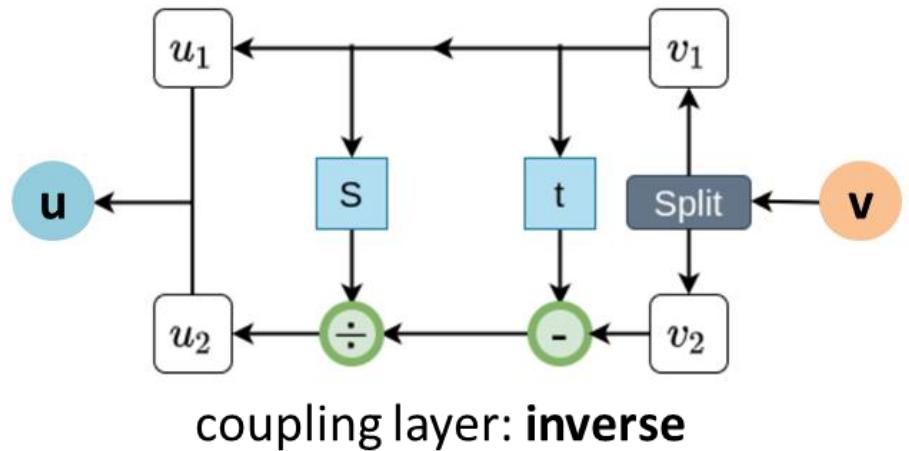
Invertible NN Can Approximate MDH

- Invertible Neural Network Φ :
 - NN with invertibility
 - E.g., coupling layers [1]
- Properties of Φ :
 - Continuous + Invertible \rightarrow Homeomorphism
- Universal approximation of INN
 - INN can approximate “any” homeomorphic mapping **arbitrarily well** with sufficient layers [2]

Challenge: How to design the loss function?



coupling layer: **forward**



coupling layer: **inverse**

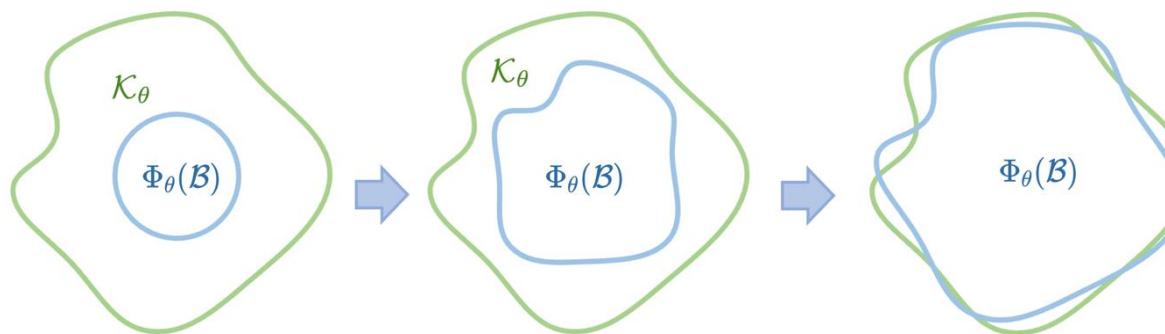
Training INN to Approximate MDH

- Training INN for homeomorphism between \mathcal{B} and \mathcal{K}

➤ Loss design: $\mathcal{L}(\Phi_\theta) = \widehat{V}(\Phi_\theta(\mathcal{B})) - \lambda_1 P(\Phi_\theta(\mathcal{B})) - \lambda_2 \widehat{D}(\Phi_\theta)$

Volume maximization Penalty for $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$ Distortion regularization

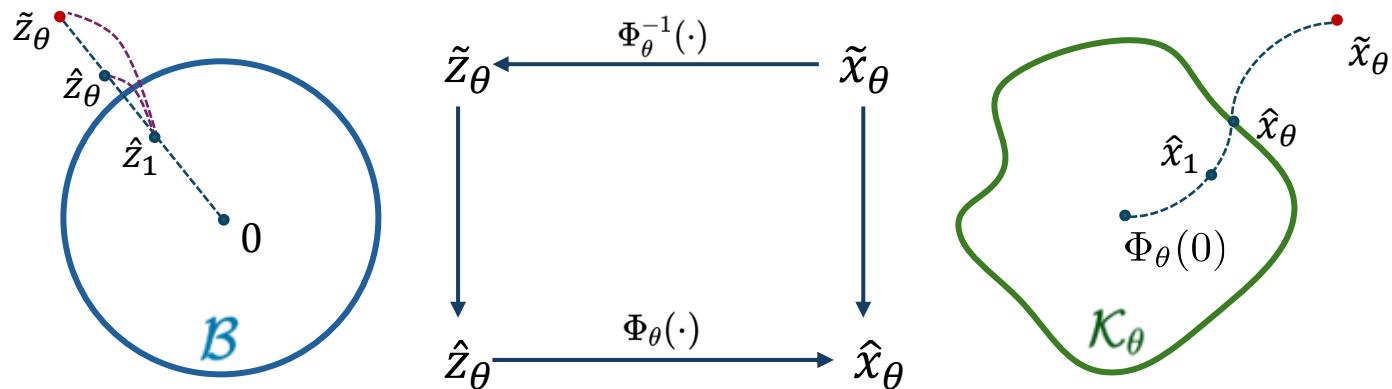
➤ Training dynamics:



➤ θ -dependent constraint $\mathcal{K}_\theta \Rightarrow$ optimize total loss $\mathbb{E}_\theta[L(\Phi_\theta)]$

- Training requirement: a **valid** trained INN, i.e., $\Phi_\theta(0) \in \mathcal{K}_\theta$
➤ Mapping the center of ball to a feasible point

Bisection Algorithm



- Given a **valid** INN and an infeasible solution

➤ **Step 1:** map it to ball space:

$$\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$$

➤ **Step 2:** bisection for α

$$\alpha^* = \sup_{\alpha \in [0,1]} \{\Phi_\theta(\alpha \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta\}$$

➤ **Step 3:** map it back

$$\hat{x}_\theta = \Phi_\theta(\alpha^* \cdot \tilde{z}_\theta)$$

Feasibility, Optimality, and Run-time

Theorem 1. For a ball-homeomorphic set, given a valid m -layer INN and an infeasible n -dim solution, the k -step bisection will return a solution with

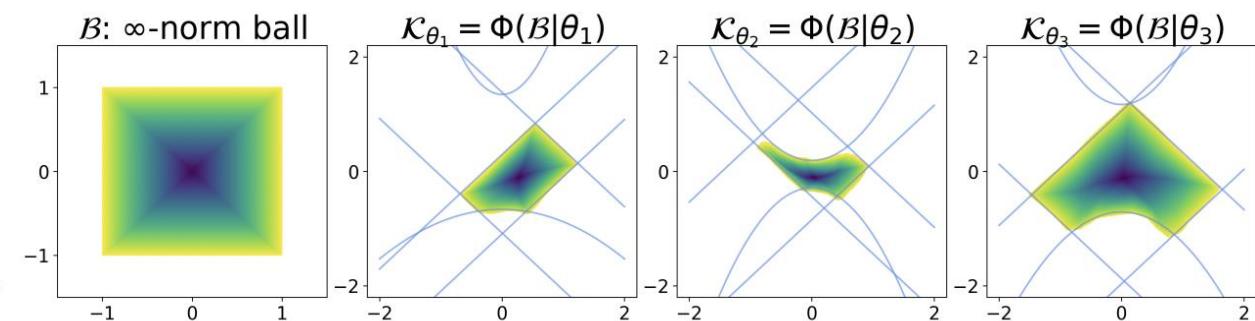
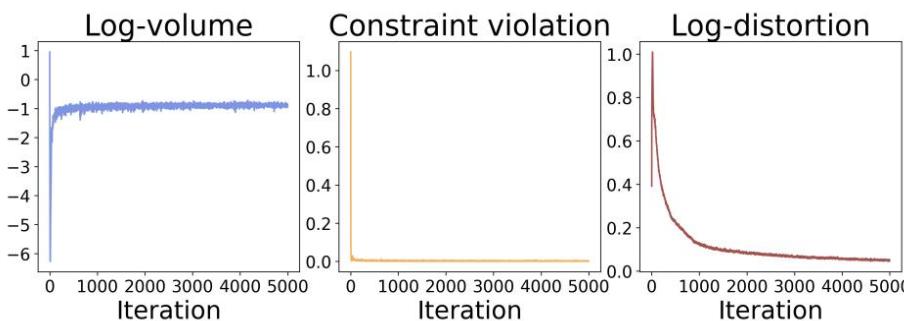
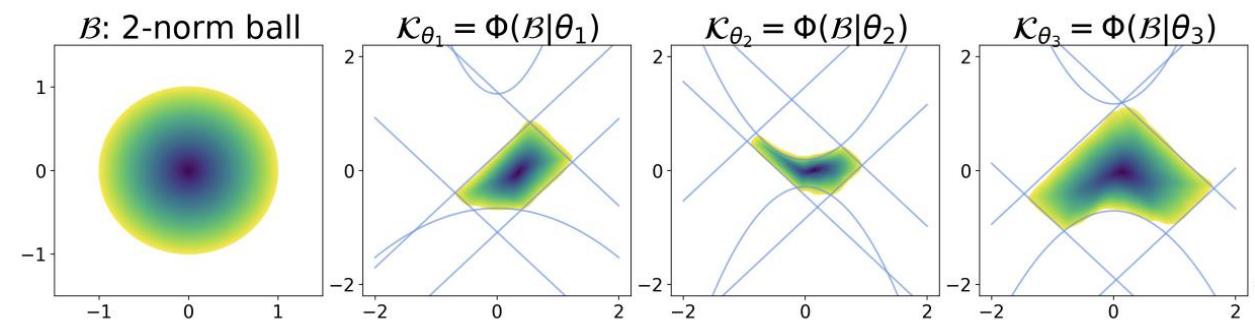
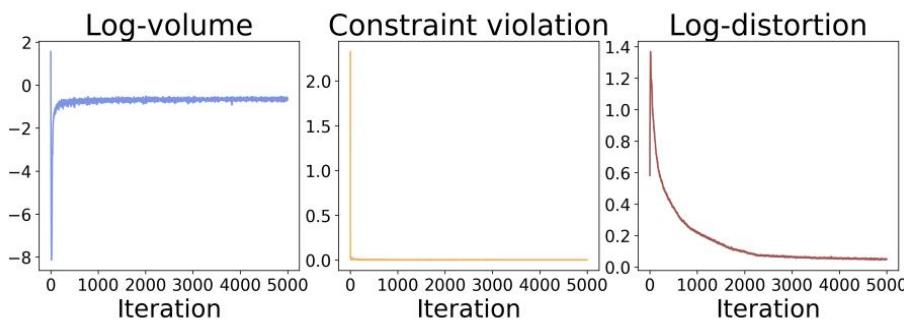
- **Feasibility guarantee**
- **Bounded optimality loss:** $\epsilon_{\text{pre}} + \epsilon_{\text{bis}} + \epsilon_{\text{hom}}$
 - ϵ_{pre} : NN prediction error
 - $\epsilon_{\text{bis}} = O(2^{-k})$: bisection-induced optimal loss
 - $\epsilon_{\text{hom}} \leq D(\Phi_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pre}})$: homeomorphism-induced optimality loss
- **Run-time complexity:** $O(kmn^2)$

Performance guarantees over ball-homeomorphic
constraints beyond convex ones

INN Learns Homeomorphism

- Learning the MDH mapping between a unit ball and a non-convex quadratic constraint set (with different input θ)

$$\mathcal{K}_\theta = \{x \in \mathbb{R}^2 \mid x^\top Qx + q^\top x + b \leq 0, \quad \theta = [Q, q, b]\}\}$$



Recovering Feasibility for Constrained Problems

- NN solutions for QCQP, SOCP, and non-convex AC-OPF
 - **100% feasibility, 0.2% extra optimality loss, 2-4 order speedup**

Method		Feasibility			Optimality				Speedup	
NN Predictor	Post Processing	feas. rate % (↑)	ineq. vio. 1-norm (↓)	eq. vio. 1-norm (↓)	solution error ave. % (↓)	objective error ave. % (↓)	cor. % (↓)	ave. × (↑)	cor. × (↑)	
Convex QCQP: $n = 200, d = 100, n_{\text{eq}} = 100, n_{\text{ineq}} = 100$										
NN	—	93.95	0.047	0	4.16	4.36	1.45	1.42	10^6	—
NN	WS	100	0	0	3.9	0	1.37	0	12.8	0.8
NN	Proj	100	0	0	4.16	4.36	1.45	1.43	21.3	1.3
NN	D-Proj	94.14	0.015	0	4.16	4.36	1.45	1.42	805	48.8
NN	H-Proj	100	0	0	4.17	4.52	1.47	1.69	8353	511
SOCP: $n = 200, d = 100, n_{\text{eq}} = 100, n_{\text{ineq}} = 100$										
NN	—	88.96	0.192	0	4.8	5.27	1.35	0.99	10^6	—
NN	WS	100	0	0	4.22	0	1.24	0	12.9	1.4
NN	Proj	100	0	0	4.8	5.26	1.37	1.14	13.6	1.5
NN	D-Proj	93.85	0.007	0	4.84	5.56	1.38	1.22	308	34
NN	H-Proj	100	0	0	4.83	5.47	1.41	1.56	6724	749
118-node AC-OPF: $n = 344, d = 236, n_{\text{eq}} = 236, n_{\text{ineq}} = 452$										
NN	—	94.92	0.002	0	9.05	9.08	0.69	0.59	10^4	—
NN	WS	100	0	0	8.59	0	0.66	0	29	1.5
NN	Proj	100	0	0	9.13	10.75	0.69	0.59	33.1	1.7
NN	D-Proj	95.41	0.002	0	9.05	9.08	0.69	0.59	24.6	1.3
NN	H-Proj	100	0	0	9.36	15.3	0.78	2.44	370	22.9

Takeaways: Hom-Proj.

- **Idea:**

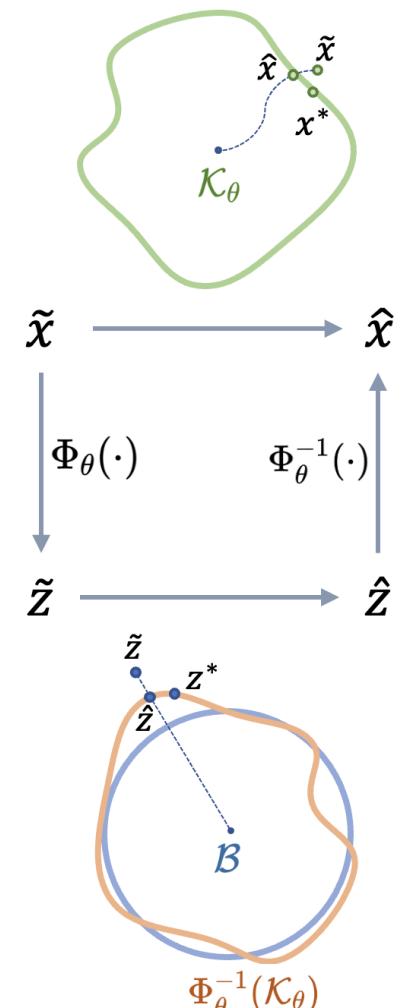
- Transform complex projection problems into easy ones via homeomorphism.

- **Contributions:**

- Learn **homeomorphism** for non-convex set via invertible neural network (with universal approximation)
- Hom-Proj to ensure NN output **feasibility** with **bounded optimality gaps** and **low complexity**

- **Future works:**

- More general non-convex constraint set [1]
- Extend to stochastic constraints [2-3]



[1] E. Liang, M. Chen, "Efficient Bisection Projection to Ensure NN Solution Feasibility over General Set", ICML 2025.

[2] M. Zhou, E. Liang*, M. Chen*, S. Low. "Partially Permutation-Invariant Neural Network for Solving Two-Stage Stochastic AC-OPF Problem.", IEEE Trans. on Power System. 2025.

[3] E. Liang*, M. Zhou, J. Zhao, M. Chen* "Solving Chance-Constrained AC-OPF Problem by Neural Network with Bisection-based Projection", ACM E-energy. 2025.

Conclusion

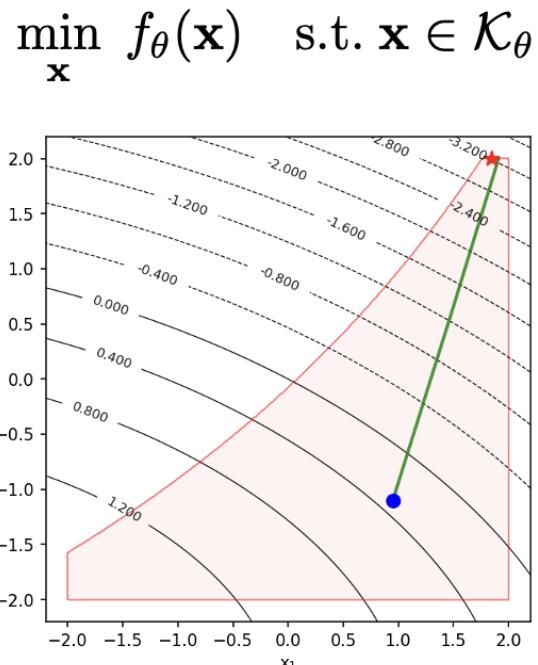
- **Homeomorphism** methods for decision-making with hard constraints
 - **Idea:** transform complex problem into simple domains
 - **Hom-PGD** to accelerate iterative algorithms [1,2]
 - **Hom-Proj.** to ensure neural network feasibility [3,4]
 - **Goals:** safe, economic, real-time decisions
- Leverage ML, optimization, and topology for smarter decision making
 - Leverage symmetry/low-dim. to further reduce complexity [5-6]
 - Generative models with hard constraints [7-8]
 - Combinatorial & Discrete Problems



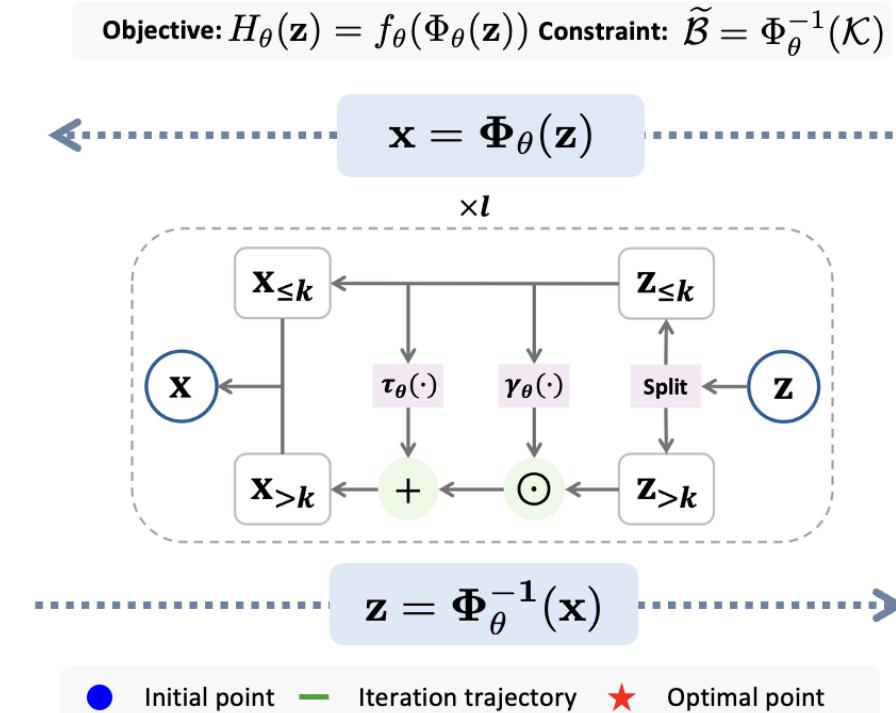
- [1] C. Liu, E. Liang*, M. Chen*, "Fast Projection-Free Algorithm (without Optimization Oracles) for Optimization over General Convex Set". **NeurIPS 2025. Spotlight.**
- [2] C. Liu, E. Liang*, M. Chen*, "Hom-PGD: Homeomorphic Reformulation for Efficient Optimization over Non-convex Sets". Under review
- [3] E. Liang, M. Chen, S. Low, "Low Complexity Homeomorphic Projection to Ensure NN Solution Feasibility for Optimization over (Non-)Convex Set", **ICML**, 2023.
- [4] E. Liang, M. Chen, S. Low, "Homeomorphic Projection to Ensure NN Solution Feasibility for Constrained Optimization". **JMLR** 2024.
- [5] E. Liang, M. Chen, "Efficient Bisection Projection to Ensure NN Solution Feasibility over General Set", **ICML** 2025.
- [6] M. Zhou, E. Liang*, M. Chen*, S. Low. "Partially Permutation-Invariant NN for Solving Two-Stage Stochastic AC-OPF Problem.", **IEEE Trans. on Power System**.
- [7] X. Li*, E. Liang*, M. Chen, "Gauge Flow Matching for Efficient Constrained Generative Modelling over General Convex Set.", **ICLR 2025 Delta Workshop. Outstanding Short Paper Award**
- [8] E. Liang, M. Chen, "Generative Learning for Solving Non-Convex Problem with Multi-Valued Input-Solution Mapping", **ICLR 2024**.

Extension I: Hom-PGD for Non-convex Constraints

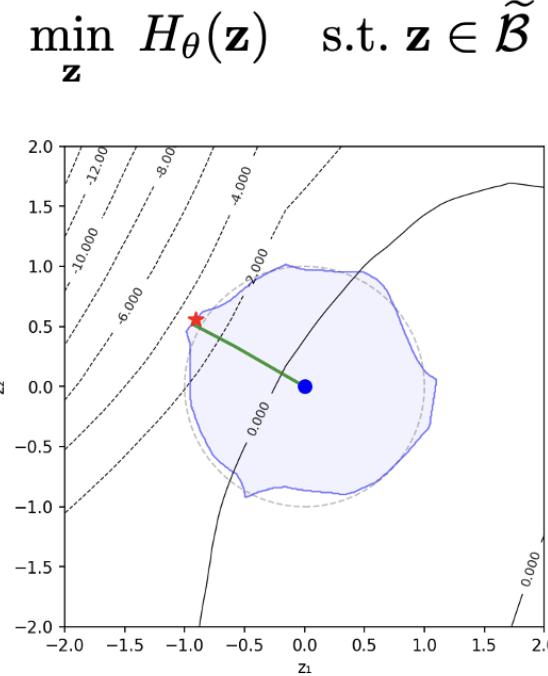
Original Space



Invertible Neural Network Φ_{θ}

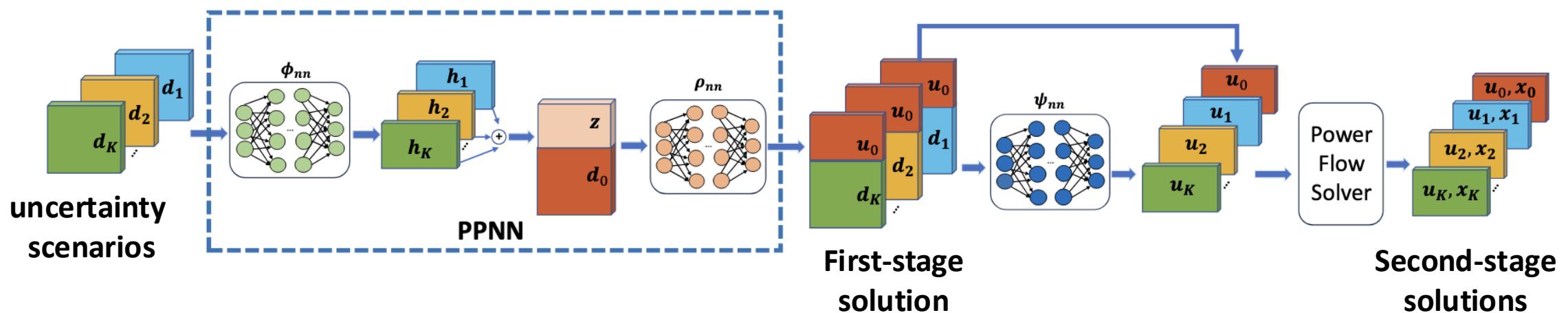


Hom. Space



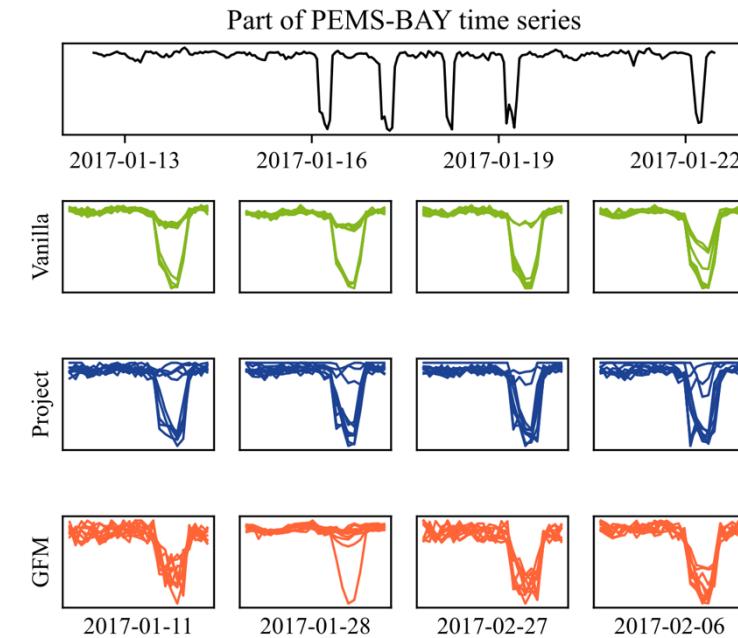
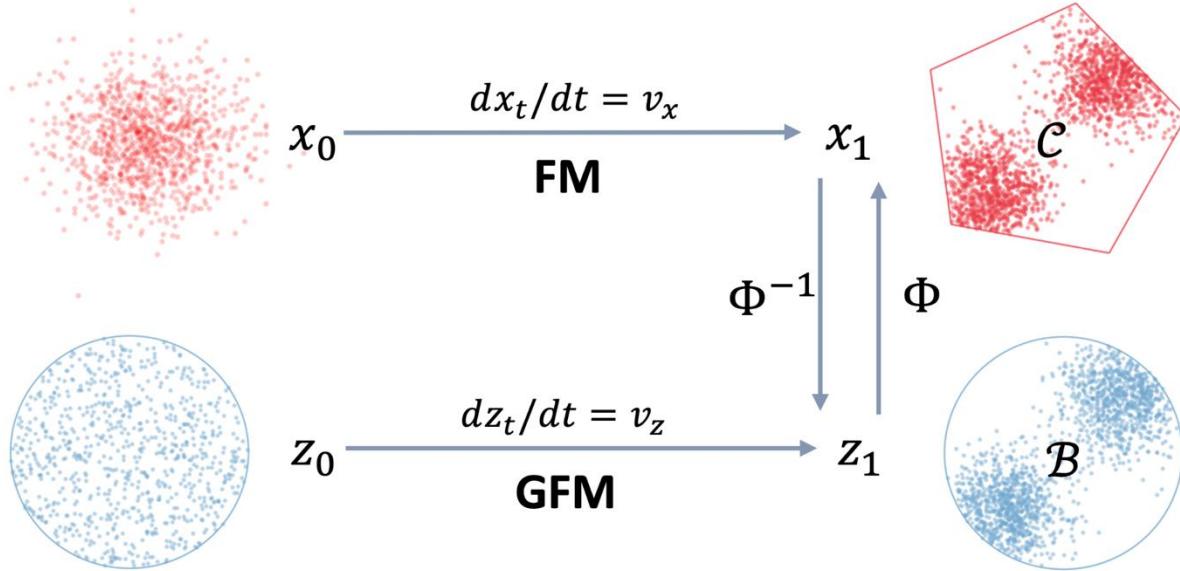
Extension II: Extension to Stochastic Problems

- Two-stage Stochastic Programs
 - *Curse of dimensionality* with increasing number of scenarios
 - Partially permutation-invariant NN to predicting solutions [1]



- Apply Hom-Proj. or Bis-Proj. to ensure solution feasibility [2]
- **2-order speedup, 0.95% optimality gap over 793-bus grid**

Extension III: Hard-Constrained Generative Models



(a) FM



(b) Projection



(c) Reflection



(d) GFM

[1] X. Li*, E. Liang*, M. Chen, "Gauge Flow Matching for Efficient Constrained Generative Modelling over General Convex Set.", ICLR 2025 Delta Workshop. Outstanding Short Paper Award

[2] X. Li, E. Liang*, M. Chen*, "Gauge Flow Matching: Efficient Constrained Generative Modelling over General Convex Set and Beyond.", ICLR 2026

Acknowledgements



Dr. Tianyu Zhao
(CUHK; HKPC-FLAIR)



Min Zhou
(CityU)



Jiawei Zhao
(CityU)



Chenghao Liu
(CityU)



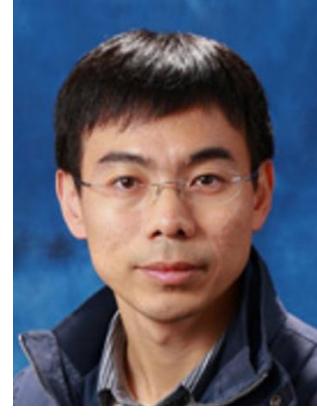
Xinpeng Li
(CityU)



Prof. Xiang Pan
(Linnan U)



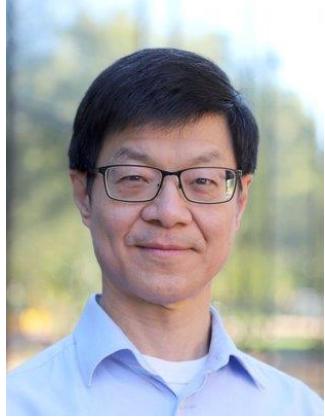
Prof. Wanjun Huang
(Beihang U)



Prof. Shengyu Zhang
(Tencent)

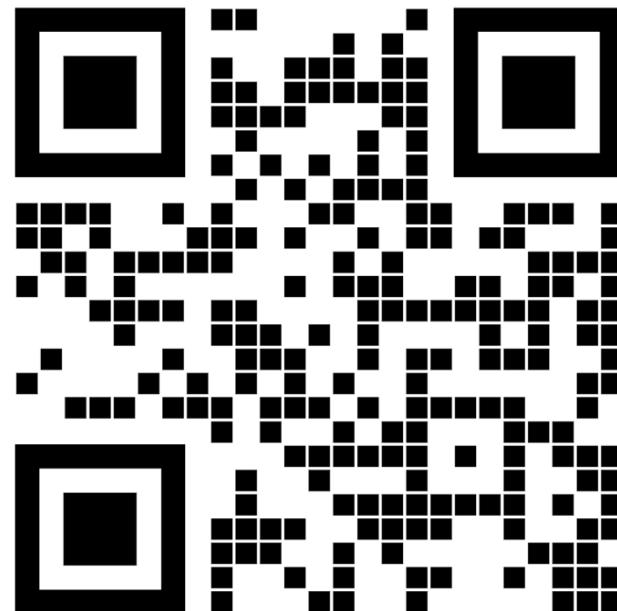


Prof. Minghua Chen
(CUHK-SZ | CityU)



Prof. Steven Low
(Caltech)

Thanks! Q&A



Personal webpage

Homeomorphism Methods: Part III

Gauge Flow Matching for Efficient Constrained Generative Modelling

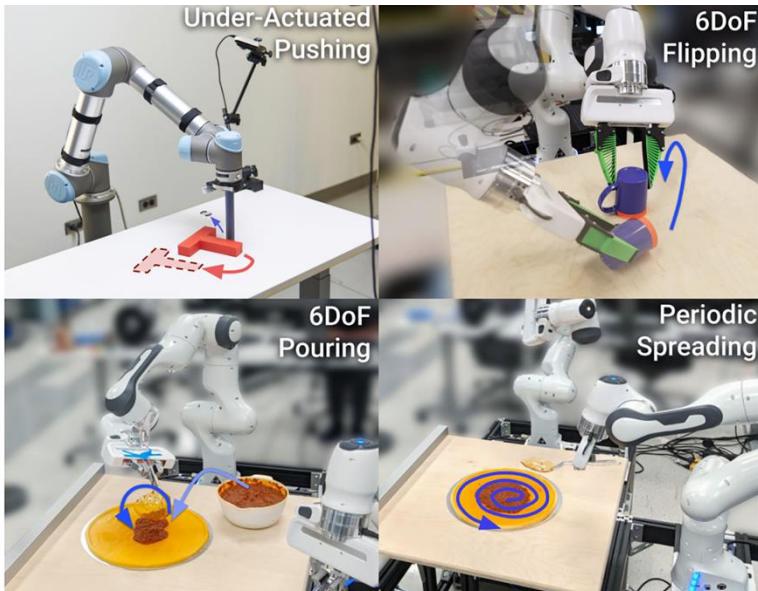
Success of DM/FM-based Generative Models

Picture/Video



[1,2]

Robotic planning



[3]

Protein/Material



[4,5]

[1] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv . 2022

[2] Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., ... & Ramesh, A. Improving image generation with better captions. OpenAI. 2023

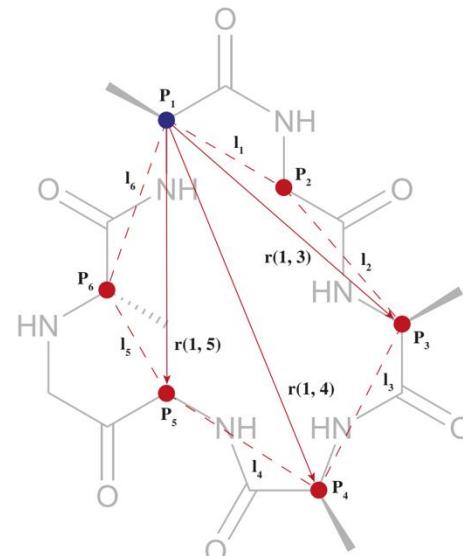
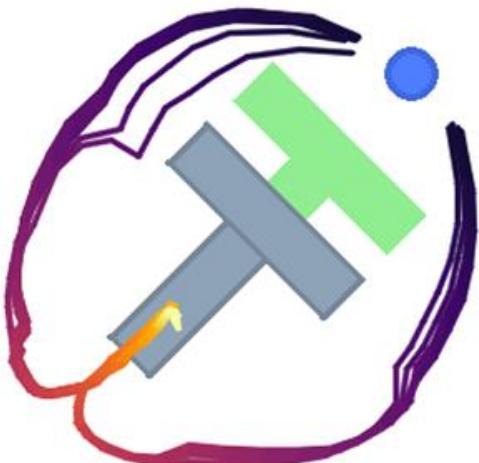
[3] Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., ... & Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. ICRR 2023.

[4] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., ... & Jumper, J. M. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature. 2024

[5] Zeni, C., Pinsler, R., Zügner, D., Fowler, A., Horton, M., Fu, X., ... & Xie, T. A generative model for inorganic materials design. Nature. 2025.

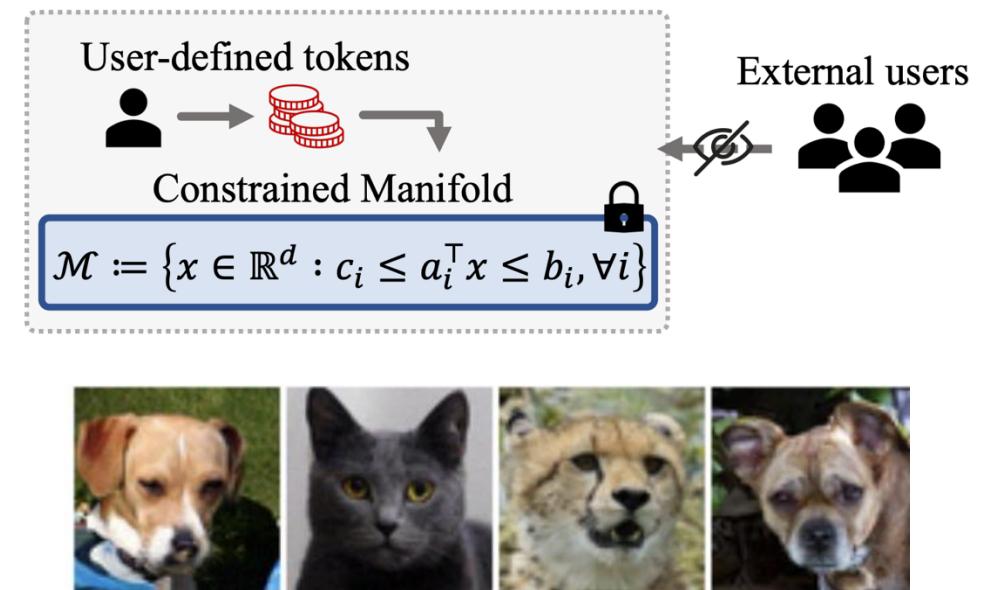
Constraint Matters in Many Applications

Physical Constraints



- Safety-critical **constraints** for robotics [1]
- Structure **constraints** in protein/material [2]

Watermarked Generation



- Embed invisible information by **constraining** user-defined tokens [3]

[1] Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., ... & Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. ICRR 2023.

[2] Fishman, N., Klarner, L., De Bortoli, V., Mathieu, E., & Hutchinson, M. J. Diffusion Models for Constrained Domains. TMLR 2023.

[3] Liu, G. H., Chen, T., Theodorou, E., & Tao, M. Mirror diffusion models for constrained and watermarked generation. NeurIPS. 2023.

DM/FM Suffers from Feasibility Issues

DM/FM Generation



Error Propagation [1]

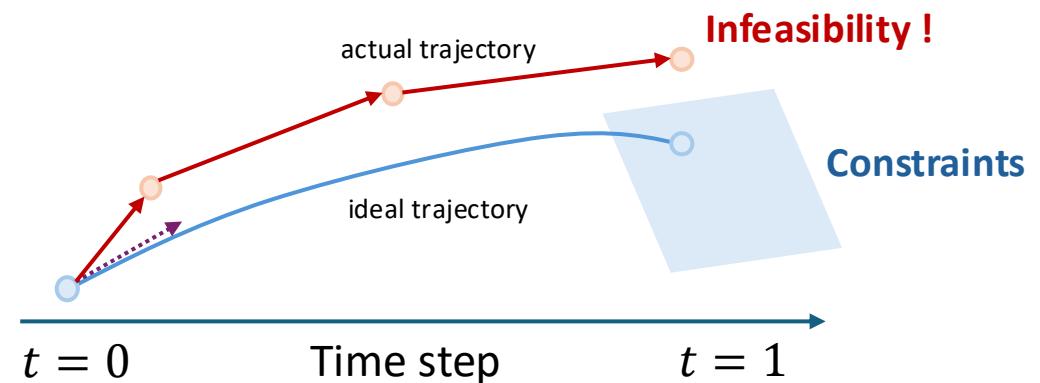
$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) + \sigma(\mathbf{x}, t)\omega_t$$

Drift

Diffusion

Error sources:

- NN **approximation** error (score function / vector field)
- **Discretized** SDE/ODE integration error



- The error bound “typically” has **exponential** dependency on the Lipschitz constants of the score functions [2,3] / vector fields [4].

[1] Li, Y., & van der Schaar, M. On Error Propagation of Diffusion Models. ICLR. 2024.

[2] Kwon, D., Fan, Y., & Lee, K. Score-based generative modeling secretly minimizes the wasserstein distance. NeurIPS 2022.

[3] Chen, S., Daras, G., & Dimakis, A. Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers. ICML 2023

[4] Benton, J., Deligiannidis, G., & Doucet, A. Error bounds for flow matching methods. TMLR. 2024.

Existing Studies on Constrained DM/FM

Limited constraint sets / lack guarantees / high complexity

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

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

Gauge Flow Matching

Dataset:

- $x_1 \sim p_{\text{data}}$ over a compact convex set \mathcal{C}

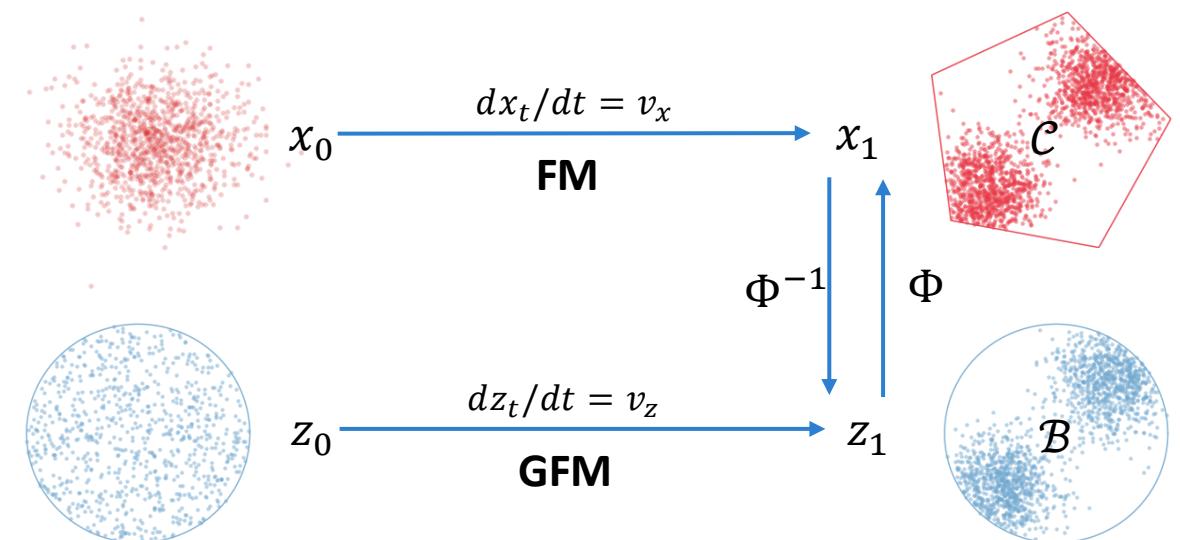
Training:

- Invers mapping: $z_1 = \Phi^{-1}(x_1)$
- Regular flow matching over ball [1]

Inference:

- Reflected generation over ball [2]
- Forward mapping: $x_1 = \Phi(z_1)$

Framework

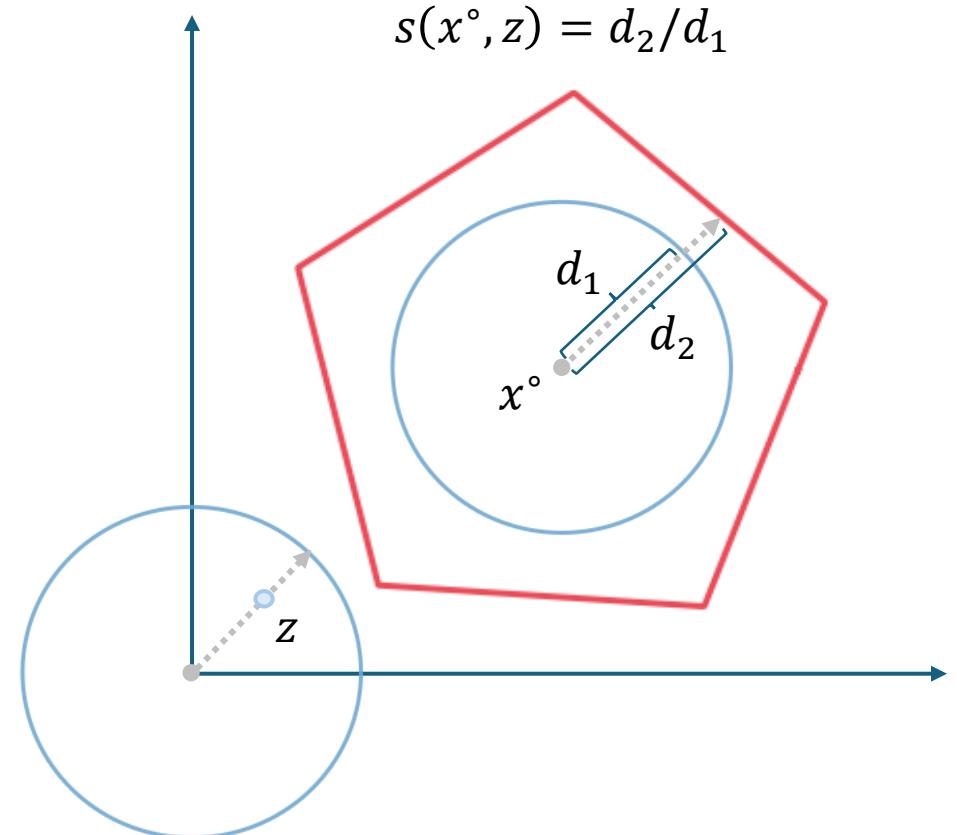


How to find invertible Φ for general convex set with low-complexity ?

Gauge Mapping: Homeomorphism for Convex Set

$$\Phi(z) = x^\circ + s(x^\circ, z) \cdot z$$

- Translating by $x^\circ \in \text{int}(\mathcal{C})$
- Scaling by $s(x^\circ, z) \in \mathbb{R}_+$
- Properties of Φ :
 - Continuous + Invertible \rightarrow Homeomorphism
 - Bi-Lipschitz \rightarrow Beneficial for algorithm [4]
- Computation of Φ :
 - **Closed-form** for common convex sets [1-2]
 - ✓ Linear, quadratic, SOC, LMI.
 - **Bisection** for general compact convex set [3]



[1] Tabas, D., & Zhang, B. Computationally efficient safe reinforcement learning for power systems. IEEE ACC 2022.

[2] Tordesillas, J., How, J. P., & Hutter, M. Rayen: Imposition of hard convex constraints on neural networks. arXiv 2023.

[3] Mhammedi, Z. Efficient projection-free online convex optimization with membership oracle. COLT 2022.

[4] X. Li*, E. Liang*, M. Chen, "Gauge Flow Matching for Efficient Constrained Generative Modelling over General Convex Set.", ICLR 2025 Delta Workshop. Outstanding Short Paper Award

Theoretical Analysis of GFM

Approximation Error

$$\mathcal{W}_2(p_{\text{data}}, p_{\theta}) \leq L_{\Phi} \cdot e^{0.5+L_{\theta}} \cdot \epsilon_{\theta}$$


RFM error [1]

Inference Complexity

$$\mathcal{O}(\text{NFE} \cdot n^2 + m \cdot C)$$


Regular FM complexity

- L_{θ} : Lipschitz of NN-based vector field
- ϵ_{θ} : l_2 flow matching loss over ball
- **L_{Φ} : Lipschitz of gauge mapping**
 - Reduced by selecting a “central” interior point $x^{\circ} \in \text{int}(\mathcal{C})$

- NFE: num of function evaluation
- n : dimension of data
- **$m \cdot C$: gauge mapping calculation**
 - m : num of constraints $\{g_i(x) \leq 0\}_{i=1}^m$
 - C : same order of complexity to calculate $g_i(x)$

Bounded approximation error + minor additional complexity

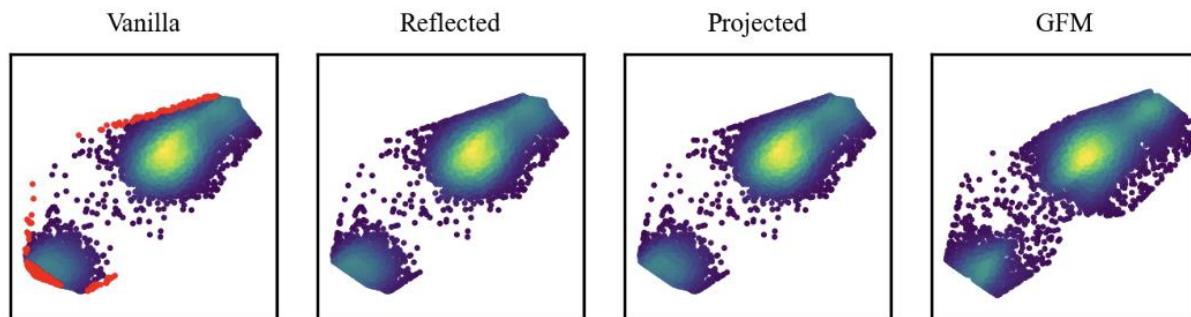
Empirical Study

Low-dim Toy Example

Simple constraints

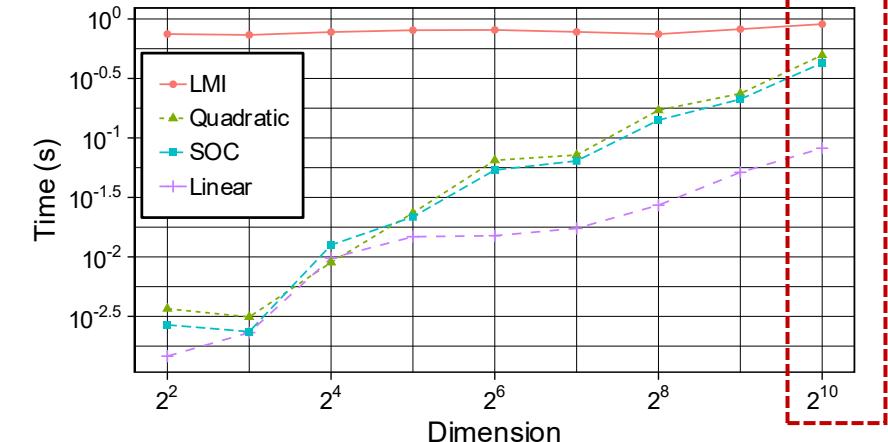
Constraint	Metrics	Vanilla FM	Reflected FM	Projected FM	GFM
Polytope ($n = 2$)	MMD (\downarrow)	0.06209	0.06155	0.06165	0.04154
	Feasibility Rate (%)	95.69	100	100	100
	Inference Time (s)	3.412	5.616	4.776	3.746
Quadratic set ($n = 3$)	MMD (\downarrow)	0.06311	0.06313	-	0.05866
	Feasibility Rate (%)	89.98	100	-	100
	Inference Time (s)	3.679	10.22	>3600	3.675

Joint linear + quadratic constraints

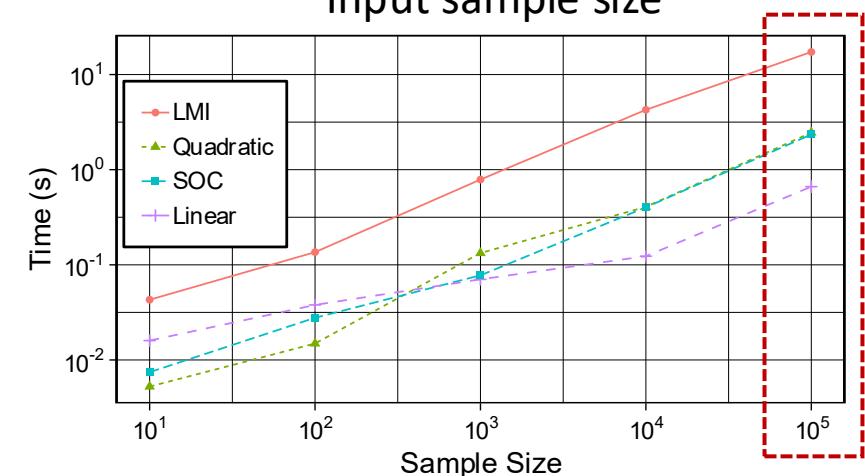


Scalability of Gauge Mapping

Constraints dimension

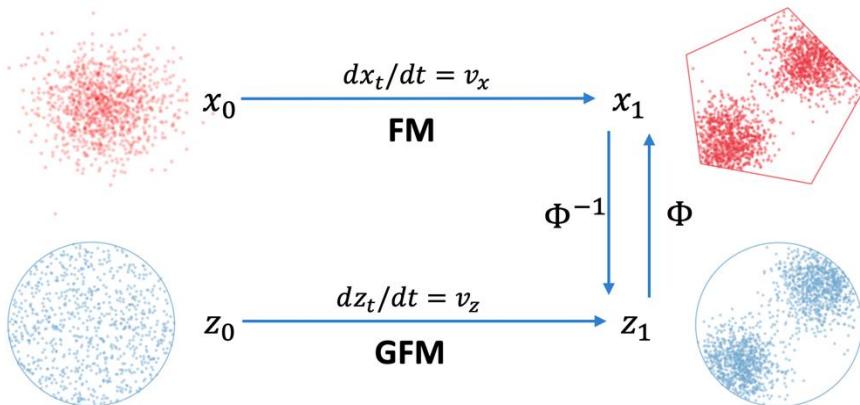


Input sample size



Summary

Takeaway messages



- Bijective gauge mapping for **general** compact convex set
- Enable **low-complexity** constrained generation

Future works

- Extend to non-convex set
 - Star/geodesic-convex set [1]
 - Ball-homeomorphic set [2]



- One-step generation
 - Constrained distillation/consistency
- More real-world applications

Supplementary Slides

Property of Input-Solution Mapping

□ Convex optimization

- Strongly convex (e.g., QP)
 - Continuous solution mapping
- General convex (e.g., LP)
 - Continuous (a.e.) solution mapping

□ Continuous optimization

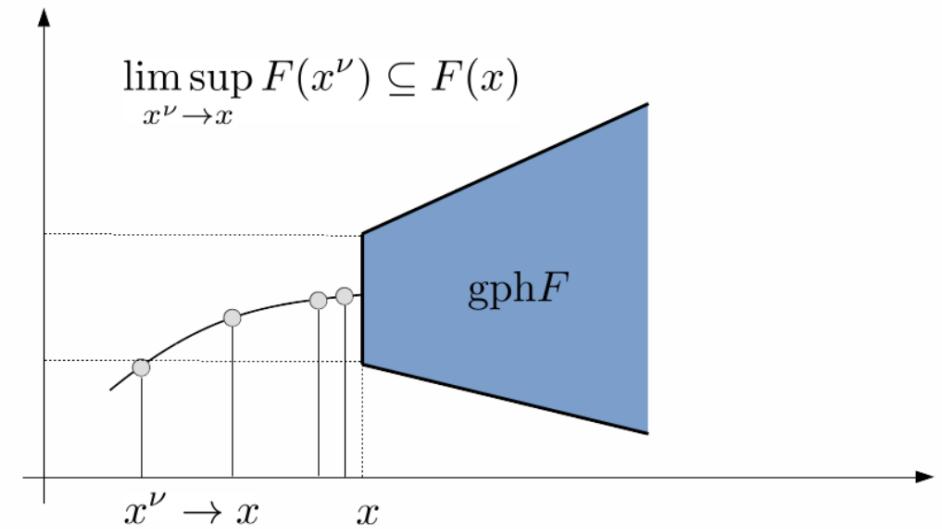
- Unique solution
 - Continuous solution mapping
- General continuous
 - “Continuous” solution mapping
 - Set-valued mapping

Theorem 3B.5 (basic continuity properties of solution mappings in optimization). In the preceding notation, let $\bar{p} \in P$ be fixed with the feasible set $S_{\text{feas}}(\bar{p})$ nonempty and bounded, and suppose that:

(a) the mapping S_{feas} is Pompeiu–Hausdorff continuous at \bar{p} relative to P , or equivalently, S_{feas} is continuous at \bar{p} relative to P with $S_{\text{feas}}(Q \cap P)$ bounded for some neighborhood Q of \bar{p} ,

(b) the function f_0 is continuous relative to $P \times \mathbb{R}^n$ at (\bar{p}, \bar{x}) for every $\bar{x} \in S_{\text{feas}}(\bar{p})$.

Then the optimal value mapping S_{val} is continuous at \bar{p} relative to P , whereas the optimal set mapping S_{opt} is osc at \bar{p} relative to P .



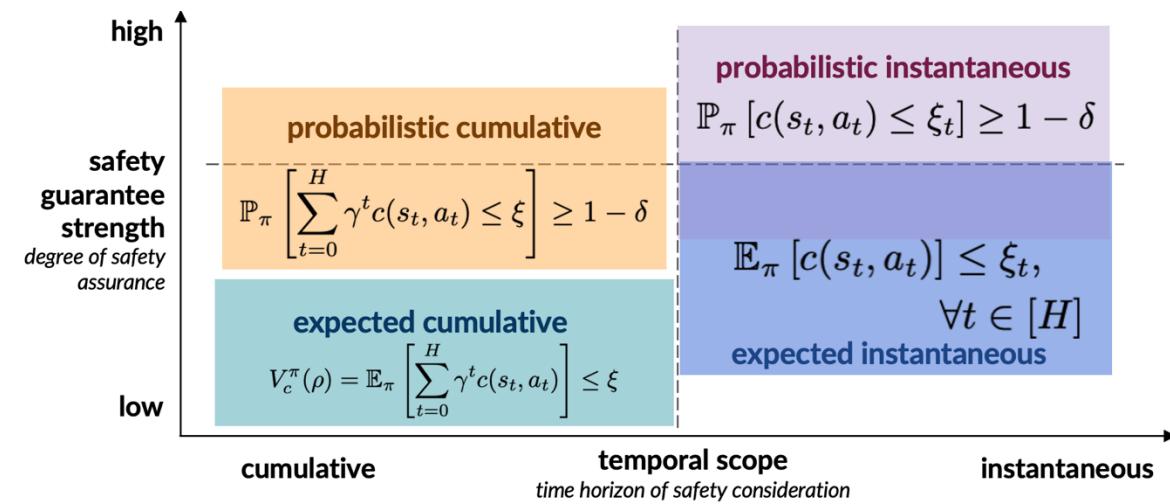
Feasibility of Sequential Problems

□ Explicit-form constraints

- Deterministic sequential problems (e.g., MPC)
 - Unroll it as a large CO / utilize problem structure
- Stochastic sequential problems (e.g., MSP)
 - Sampling \Rightarrow det. problem \Rightarrow feasibility with prob.
- Infinite horizon problem (e.g., Stability)
 - Construct explicit safe state/policy set

□ Unknown, but can sampling

- Constrained MDP
 - Zero-duality gap
 - Primal-dual algorithm
 - Feasibility with prob.



Discrete & Combinatorial Problems

□ Solving continuous sub-problems

- Continuous relaxation + B&B/cutting-plane
 - ML to predict continuous problems [3]
 - ML to accelerate B&B/cutting-plane [4-5]

□ Equivalent continuous formulation

- Motzkin-Straus formulation of Max-Clique problem
 - Constrained indefinite quadratic program [1]

□ Relaxation + Recovering

- SDP relaxation + randomized rounding for Max-Cut problem
 - 0.879 approximation ratio [2]

[1] Gibbons, L. E., Hearn, D. W., Pardalos, P. M., & Ramana, M. V. Continuous characterizations of the maximum clique problem. *MATH OPER* 1997.

[2] M. X. Goemans and D. P. Williamson. .879-approximation algorithms for max cut and max 2sat. *STOC* 1994.

[3] Kool, W., van Hoof, H., & Welling, M. Attention, Learn to Solve Routing Problems!. *ICLR* 2019

[4] Khalil, E., Le Bodic, P., Song, L., Nemhauser, G., & Dilkina, B. Learning to branch in mixed integer programming. *AAAI* 2016

[5] Balcan, M. F., Dick, T., Sandholm, T., & Vitercik, E. Learning to branch. *ICML* 2018.

Equality and Inequality Constraint

- Constraint set: $\mathcal{K}_\theta = \{x | g(x, \theta) \leq 0, h(x, \theta) = 0\}$

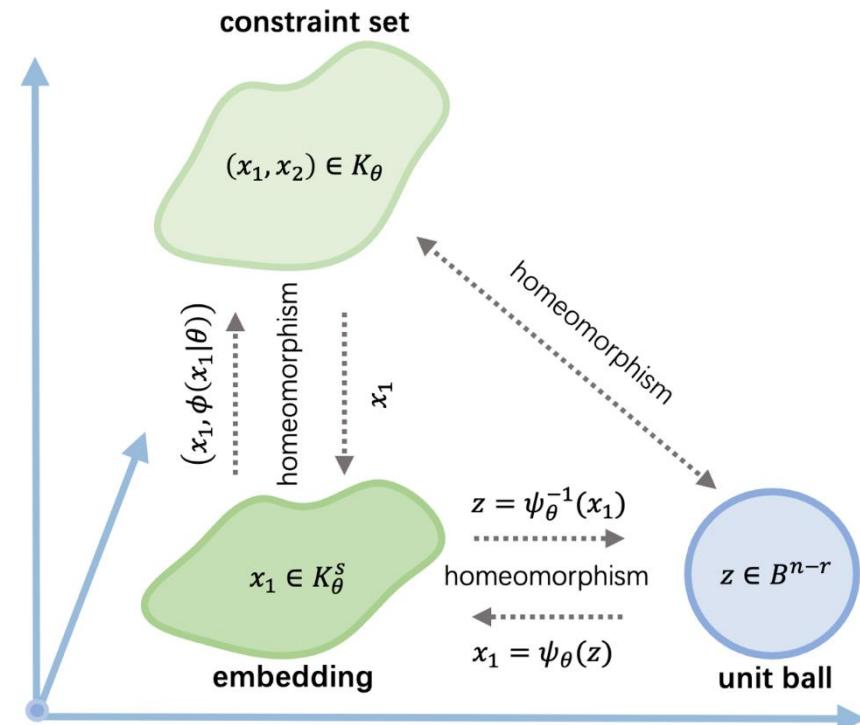
- **Algorithmic view:**

- **predict-then-reconstruct** [1-2]

- $x = [x_1, x_2]$
 - $h(x_1, \phi(x_1)) = 0$
 - Linear \Rightarrow linear mapping ϕ
 - Non-linear \Rightarrow implicit mapping ϕ
 - $\mathcal{K}_\theta^s = \{x | g(x_1, \phi(x_1)) \leq 0\}$

- **Topological view:**

- **Constant-Level-Set Theorem** [3]
 - $\mathcal{K}_\theta \cong \mathcal{K}_\theta^s \cong \mathcal{B}$



[1] Pan, X., Chen, M., Zhao, T., & Low, S. H. DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. IEEE SJ

[2] Donti, P. L., Rolnick, D., & Kolter, J. Z. DC3: A learning method for optimization with hard constraints. ICLR 2021

[3] Lee, J. M., & Lee, J. M. (2012). Smooth manifolds (pp. 1-31). Springer New York.

Distortion and Homeomorphic Mapping

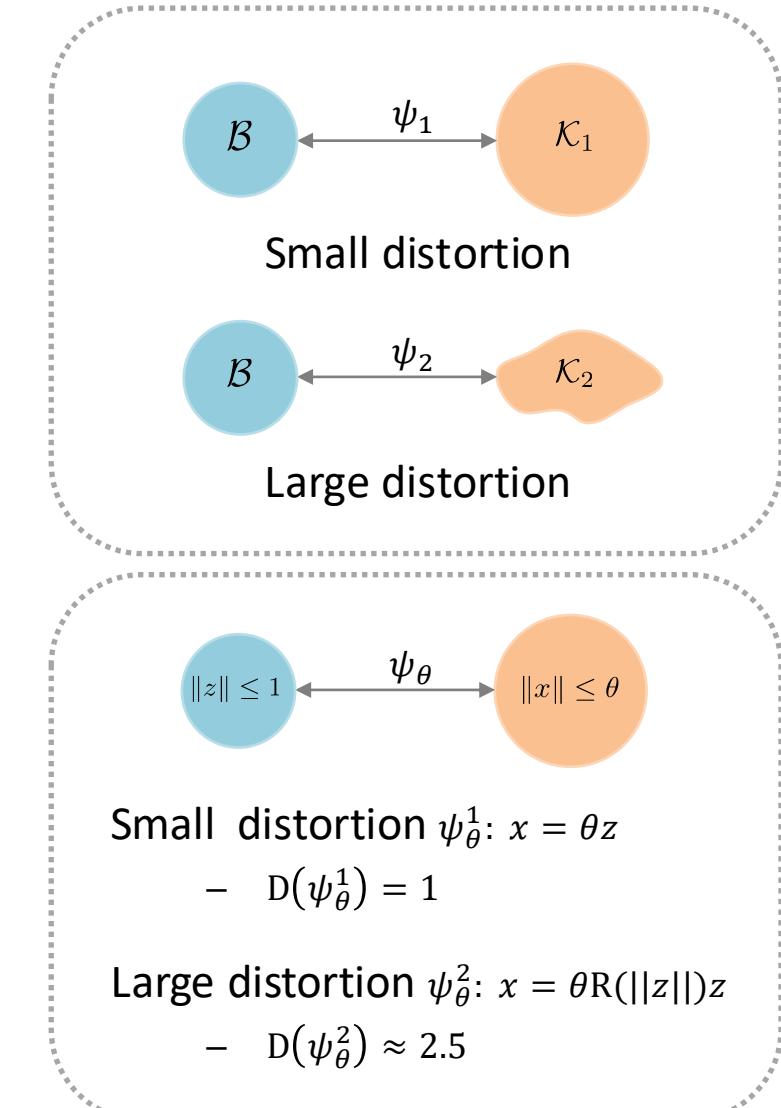
- Distortion: $D(\psi) = k_2/k_1 \geq 1$

$$k_2 = \sup_{z_1, z_2} \left\{ \frac{\|\psi(z_1) - \psi(z_2)\|}{\|z_1 - z_2\|} \right\}, \quad k_1 = \inf_{z_1, z_2} \left\{ \frac{\|\psi(z_1) - \psi(z_2)\|}{\|z_1 - z_2\|} \right\}$$

- Ratio of max & min distance **variations** by ψ
- Exist **multiple** homeomorphic mappings between two sets
- **Bi-Lipschitz** Constants: $D(\psi) = \text{Lip}(\psi) \cdot \text{Lip}(\psi^{-1})$

- Minimum Distortion Homeomorphism (MDH)

- In Hom-PGD:
 - Small distortion \Rightarrow faster convergence speed
- In Hom-Proj:
 - Small distortion \Rightarrow small opt. loss by HP



Calculation of Gauge Mapping II

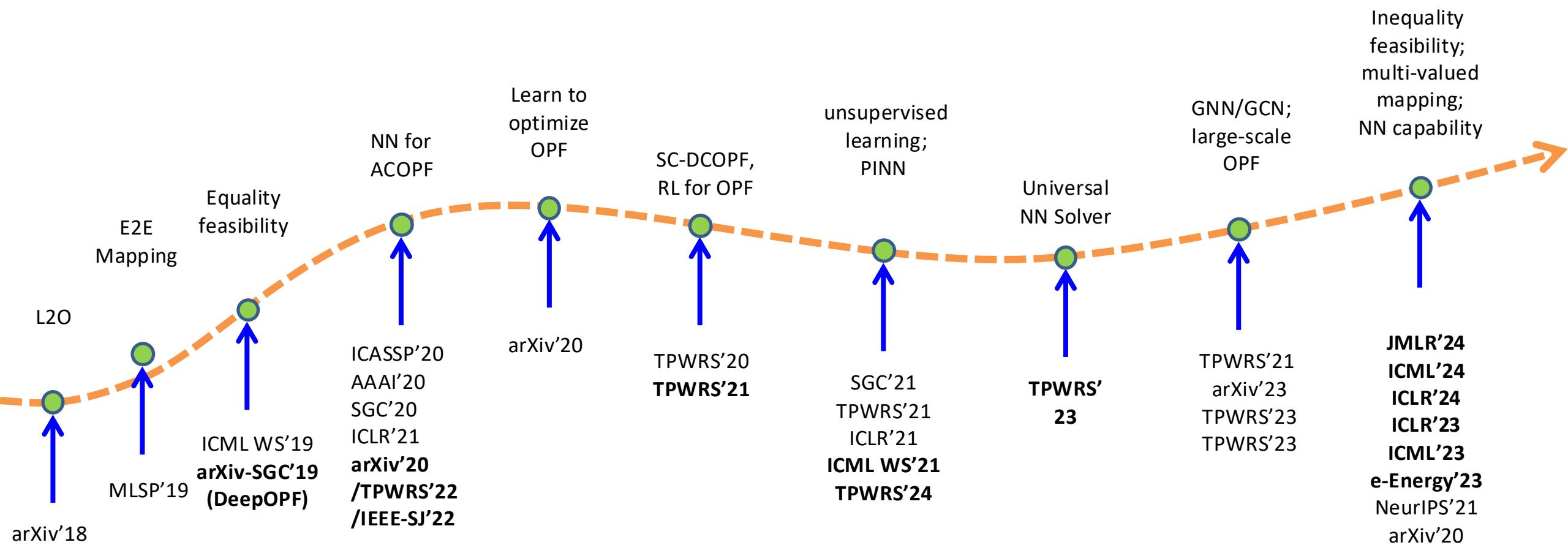
Algorithm 1 Bisection Algorithm for Point-to-Boundary Distance

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

```
1: Initialize:  $\alpha_l = 0$  and  $\alpha_u = 1$ 
2: while  $|\alpha_l - \alpha_u| \geq \epsilon$  do
3:   if  $x^\circ + \alpha_u \cdot v \in \mathcal{C}$  then
4:     increase lower bound:  $\alpha_l \leftarrow \alpha_u$ 
5:     double upper bound:  $\alpha_u \leftarrow 2 \cdot \alpha_m$ 
6:   else
7:     bisection:  $\alpha_m = (\alpha_l + \alpha_u)/2$ 
8:     if  $x^\circ + \alpha_m \cdot v \in \mathcal{C}$  then
9:       increase lower bound:  $\alpha_l \leftarrow \alpha_m$ 
10:    else
11:      decrease upper bound:  $\alpha_u \leftarrow \alpha_m$ 
12:    end if
13:  end if
14: end while
```

Output: $d_{\mathcal{C}}(x^\circ, v) \approx \alpha_m$

Research Landscape



- Our works in **bold font**
- [Wiki: `https://energy.hosting.acm.org/wiki/index.php/ML_OPF_wiki`](https://energy.hosting.acm.org/wiki/index.php/ML_OPF_wiki)