Low Complexity Homeomorphic Projection to Ensure Neural-Network Solution Feasibility for Optimization over (Non-)Convex Set

Enming Liang [1]   Minghua Chen [1]   Steven H. Low [2]

Abstract

There has been growing interest in employing neural network (NN) to directly solve constrained optimization problems with low run-time complexity. However, it is non-trivial to ensure NN solutions strictly satisfying problem constraints due to inherent NN prediction errors. Existing feasibility-ensuring methods either are computationally expensive or lack performance guarantee. In this paper, we propose *homeomorphic projection* as a low-complexity scheme to guarantee NN solution feasibility for optimization over a general set homeomorphic to a unit ball, covering all compact convex sets and certain classes of non-convex sets. The idea is to (i) learn a minimum distortion homeomorphic mapping between the constraint set and a unit ball using an invertible NN (INN), and then (ii) perform a simple bisection operation concerning the unit ball so that the INN-mapped final solution is feasible with respect to the constraint set with minor distortion-induced optimality loss. We prove the feasibility guarantee and bound the optimality loss under mild conditions. Simulation results, including those for non-convex AC-OPF problems in power grid operation, show that homeomorphic projection outperforms existing methods in solution feasibility and run-time complexity, while achieving similar optimality loss.

1. Introduction

Constrained Optimization (CO) has tremendous applications in various engineering domains, including supply chain, transportation, power systems, and resource allocation. A large number of iterative algorithms have been developed and incorporated into commercial solvers (e.g., Gurobi and MOSEK) to solve various CO problems exactly or approximately. While widely successful, iterative algorithms can still fail to solve challenging CO problems in real-time, limiting their usefulness in time-sensitive applications, such as solving AC optimal power flow problems in real-time power grid operations and semi-definite programming-based real-time scheduling and coding operations in modern wireless communication systems.

Recently, machine learning (ML) schemes have been developed for solving CO in real-time, including the end-to-end (E2E) solution mapping (Kotary et al., 2021; Amos, 2022) and the learning-to-optimize (L2O) iterative scheme (Khalil et al., 2016; Chen et al., 2021b). Boosted by the universal approximation capacity of neural networks (NN) (Hornik et al., 1989; Leshno et al., 1993), the end-to-end approaches learn the mapping between the input parameters and high-quality solutions of CO. After the training procedure, NN directly outputs the solution in real-time, which is much faster than iterative solvers. For example, researchers have developed NN-based approaches for optimal power flow (OPF) problems, where NN predicts the optimal power generation schemes based on real-time power demand (Pan et al., 2019; Guha et al., 2019; Pan et al., 2020; Fioretto et al., 2020; Zamzam & Baker, 2020; Donti et al., 2021).

However, it is non-trivial to ensure NN solution feasibility with respect to the problem constraints, due to inherent NN prediction errors. Existing feasibility-ensuring methods are either computationally expensive or lacking performance guarantees. See Sec. 2 for detailed discussions.

In this paper, we develop *homeomorphic projection* (HP) as a novel low-complexity approach to take an infeasible NN solution and generate a feasible solution with bounded optimality loss. We make the following contributions:

▷ After presenting the optimization problem over a general ball-homeomorphic set in Sec. 3, we propose an HP framework for ensuring NN solution feasibility in Secs. 4 and 5. The framework includes (i) training an invertible neural network (INN) in an unsupervised manner to approximate a minimum distortion homeomorphic (MDH) mapping between the constraint set and a unit ball, and (ii) performing

---
[1] School of Data Science, City University of Hong Kong. [2] Depts. of EE and CMS, Caltech. Correspondence to: Minghua Chen <minghua.chen@cityu.edu.hk>.

Table 1: Comparison of existing approaches for ensuring NN solution feasibility for constrained optimization problems.

Existing Study (see Sec. 2 for references)	Input-Adaptive Constraint Set	Solution Feasibility Guarantee	Bounded Optimality Loss	Low Run-Time Complexity	Low Training Complexity
Penalty approach	✓	✗	✗	✓	✓
Projection approach	✓	✓	✓	✗	N/A
Sampling approach	✗	✓	✓	✗	✗
Preventive learning	✓(linear)	✓	✗	✓	✗
Gauge mapping	✓(linear)	✓	✗	✓	N/A
Homeomorphic Projection	✓	✓	✓	✓	✓

simple bisection operation with respect to the unit ball so that the corresponding final solution in the constraint set is feasible with minor distortion-induced optimality loss.

▷ In Sec. 6, we prove that the HP framework can take an infeasible solution and recover a feasible solution with bounded overall optimality loss. We also discuss the run-time complexity and training complexity of the framework.

▷ In Sec. 7, we carry out simulations, including for solving non-convex AC-OPF problems in power grid operations, to evaluate the performance of our HP approach. The results show that homeomorphic projection outperforms existing methods in feasibility and run-time complexity, while achieving similar optimality loss.

To the best of our knowledge, this is the first work to guarantee NN solution feasibility for (fairly) general constrained optimization problems, with bounded optimality loss and low run-time complexity. Code is available at HP_Code.

2. Related Work

Machine Learning (ML) driven optimization research can be categorized into two main areas: L2O iterative scheme (Khalil et al., 2016; Chen et al., 2021b) and E2E solution mapping (Kotary et al., 2021; Amos, 2022). For both research lines, guaranteeing the feasibility of the output solution by trained NN under input-dependent constraint is nontrivial. Researchers have developed different approaches to improve the feasibility, and a summary is in Table 1.

Penalty approach. To reduce the constraint violation of predicted solutions, different penalty functions (e.g., quadratic function) are designed and augmented in the loss function (Cheng et al., 2019; Pan et al., 2019; 2020; Zamzam & Baker, 2020; Fioretto et al., 2020). Considering the optimal condition of CO, Karush–Kuhn–Tucker (KKT) conditions (a set of equations) are treated as equality constraints to improve the performance of NN (Nellikkath & Chatzivasileiadis, 2021a;b; Zhang et al., 2021). However, those approaches do not guarantee feasibility over constraints set due to the prediction error of NN.

Projection approach. To ensure feasibility, projection can be applied for infeasible solutions. However, either solv-

ing projection problem by optimization solver (Diamond & Boyd, 2016) or applying equivalent projection layers (Amos & Kolter, 2017; Agrawal et al., 2019; Chen et al., 2021a) is computationally expensive and inefficient in real-time. Differentiable gradient-based methods are proposed to correct infeasible solutions (Donti et al., 2021). L2O-based methods are also proposed to learn the iterative process of projection by different types of NN (Xia & Wang, 2000; Heaton et al., 2021; 2022). However, those projection-equivalent approaches do not guarantee feasibility for general constraints.

Sampling approach. To guarantee feasibility, feasible points are sampled and used to construct the inner approximation of the original constraint set. A convex combination of vertexes and rays is adopted to ensure feasibility under linear constraints (Frerix et al., 2020; Zheng et al., 2021). For a general but input-invariant constraint sets, sampling-based methods are theoretically studied in (Kratsios et al., 2021). These methods work for simple linear constraints or input-invariant constraints, and the number of required feasible samples grows exponentially with the dimension of the decision variable, which limits their potential for complex CO problems.

Preventive learning and gauge mapping. These methods are dedicated to finding a feasibility-guaranteed NN and then improving its optimality. A preventive learning framework is proposed in (Zhao et al., 2023), which calibrates inequality constraints and ensures the feasibility of NN by solving mixed-integer programs at each iteration. Another work utilizes a closed-form gauge mapping to constrain the output within a polytopic set (Tabas & Zhang, 2021; 2022). Those approaches only work for linear constraints, and there lacks a general and computational tractable approach to realizing feasibility over general constraints.

In summary, existing schemes to ensure NN solution feasibility either incur high run-time complexity or lack feasibility/optimality guarantee. In this paper, we propose *homeomorphic projection* as a low run-time complexity scheme to guarantee NN solution feasibility with bounded optimality loss. Our scheme is conceptually related to the projection approach and gauge mapping. But it is uniquely different in its design, applicability to non-linear constraints, and performance guarantee.

3. Settings and Open Issue to Address

We consider a general constrained optimization problem:

$$\min_{x \in \mathbb{R}^n} \ f(x, \theta) \quad \text{s.t.} \ \ x \in \mathcal{K}_\theta, \tag{1}$$

where $x \in \mathbb{R}^n$ is the decision variable and $\theta \in \Theta \subset \mathbb{R}^d$ is the input parameter. The objective function $f(x, \theta)$ is continuous and can be non-convex. The optimal solution of problem in (1), assumed to be unique, is denoted as $x_\theta^* = \arg\min_{x \in \mathcal{K}_\theta} \{f(x, \theta)\}$. The constraint set $\mathcal{K}_\theta$ is compact and specified by n_{ineq} inequalities[1]: $\mathcal{K}_\theta = \{x | g_i(x, \theta) \le 0, i = 1, \ldots, n_{\text{ineq}}\}$, where $g_i(x, \theta)$ is a continuous function.
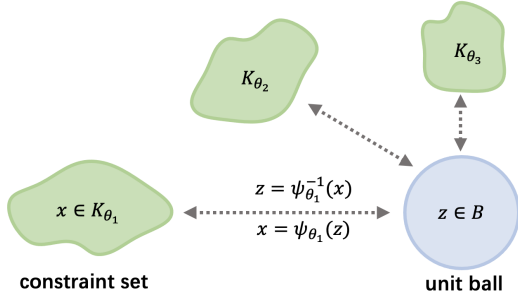

Figure 1: Homeomorphism between constraint sets and a unit ball.

Assumption 1. $\forall \theta \in \Theta$, $\mathcal{K}_\theta$ is homeomorphic to the unit ball[2] in $\mathbb{R}^n$, denoted as $\mathcal{K}_\theta \cong \mathcal{B}$. See Fig. 1 for illustration.

Homeomorphic mapping (or homeomorphism) is a one-to-one mapping between topological spaces (e.g., $\mathcal{K}_\theta$ and $\mathcal{B}$) that is continuous in both directions (Lee, 2013). In other words, Assumption 1 means that the constraint set and unit ball are topologically equivalent, and we can continuously stretch a ball to the constraint set and vice versa.

We note that Assumption 1 is easy to satisfy, e.g., by *any compact convex set* (Geschke, 2012), and *certain classes of compact and simply-connected non-convex set* (e.g., star set and invex set) (Sapkota & Bhattarai, 2021). Thus, the formulation in (1) under Assumption 1 is pretty general and covers many continuous optimization problems in various domains.

3.1. Open Issue: ensuring NN solution feasibility

As discussed in the introduction and related works, there has been NN schemes that learn the input-solution mapping

---

[1] While we do not explicitly consider equality constraints in the formulation, we remark that equality constraints with constant rank (e.g., linear equations) can be exploited and removed without losing optimality as discussed in Appendix A. We thus focus on problems with only inequality constraints.

[2] The p-norm ball is is defined as $\mathcal{B}_p(x_0, r) = \{x \in \mathbb{R}^n | \|x - x_0\|_p \le r\}$, we denote the zero-centered unit 2-norm ball as $\mathcal{B}$.

$F : \mathbb{R}^d \to \mathbb{R}^n$ for a constrained problem and pass inputs through the NN to obtain high-quality solutions instantly; see e.g., (Pan et al., 2020; Donti et al., 2021; Amos, 2022). However, it is non-trivial to ensure NN solution feasibility with respect to the problem constraints due to inherent NN prediction error, defined as $\epsilon_{\text{pre}} = \sup_{\theta \in \Theta} \{\|F(\theta) - x_\theta^*\|\}$. As discussed in Sec. 2, existing feasibility-ensuring methods are either computationally expensive or lacking performance guarantees. To date, it remains largely open to ensure NN solution feasibility to the problem in (1) with bounded optimality loss and low run-time complexity.
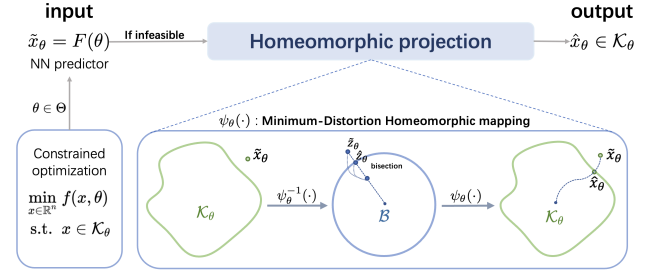

Figure 2: Overview of the HP framework.

4. Our Homeomorphic Projection Framework

We develop *homeomorphic projection* as a low-complexity approach to take an infeasible solution to the problem in (1) and generate a feasible solution with bounded optimality loss. As shown in Fig. 2, the idea is to (i) learn a minimum distortion homeomorphic (MDH) mapping between the constraint set $\mathcal{K}_\theta$ and a unit ball $\mathcal{B}$ as defined in Sec. 4.1, and then (ii) perform a simple bisection operation with respect to the unit ball so that the corresponding solution in the constraint set is feasible with minor optimality loss, as discussed in Sec. 4.2. We present a method to learn MDH mappings using INN in Sec. 5 and carry out performance analysis in Sec. 6.

4.1. Minimum distortion homeomorphic mapping

Definition 4.1 (Distortion). Let $\psi : \mathbb{R}^n \to \mathbb{R}^n$ be a homeomorphic mapping. Its distortion over a compact set $\mathcal{Z}$ is defined as $\mathrm{D}(\psi, \mathcal{Z}) = \kappa_2 / \kappa_1 \ge 1$, where

$$\kappa_1 = \inf_{z_1, z_2 \in \mathcal{Z}, \ z_1 \ne z_2} \{\|\psi(z_1) - \psi(z_2)\| / \|z_1 - z_2\|\}, \tag{2}$$

$$\kappa_2 = \sup_{z_1, z_2 \in \mathcal{Z}, \ z_1 \ne z_2} \{\|\psi(z_1) - \psi(z_2)\| / \|z_1 - z_2\|\}. \tag{3}$$

Distortion evaluates the variation of distance metrics in different spaces transformed by a mapping ψ. Small distortion, e.g., close to its minimum value 1, implies that geometrical operation in one space, e.g., projection onto a set, can be approximately done in its mapped space with respect to distance measure and vice versa. Mappings with

unit distortion are called isometric mappings. Mappings with bounded distortion are also referred as bi-Lipschitz mappings. The concept has been widely applied in the embedding studies (Xiao et al., 2018; Agrawal et al., 2021) and computational graphics (Schmidt et al., 2019; Liu et al., 2022).

The first step in our framework is to learn an MDH mapping ψ_θ between $\mathcal{K}_\theta$ and $\mathcal{B}$, critical for bounding optimality loss.

Definition 4.2 (MDH mapping). The MDH mapping is defined as the optimal solution for the following problem:

$$\min_{\psi_\theta \in \mathcal{H}^n} \ \log \mathrm{D}(\psi_\theta^{-1}, \mathcal{X}_\theta) \quad \text{s.t.} \ \mathcal{K}_\theta = \psi_\theta(\mathcal{B}), \quad (4)$$

where $\mathcal{H}^n$ is the set of all n-dim homeomorphic mappings, and we denote the set of homeomorphic mappings satisfying $\mathcal{K}_\theta = \psi_\theta(\mathcal{B})$ as $\mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$. The set $\mathcal{X}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\mathrm{pre}})$[3] contains all possible outputs of the NN predictor.

We **remark** that solving the problem in (4) gives (at least) one homeomorphic mapping with the minimum distortion among all such mappings between $\mathcal{K}_\theta$ and $\mathcal{B}$, as (i) it has feasible solution according to Assumption 1 and (ii) the distortion of a homeomorphic mapping over a compact set $\mathcal{X}_\theta$ is bounded (Behrmann et al., 2021). We then denote an optimal solution as ψ_θ^*.

In general, the problem in (4) is an infinite-dimension one and challenging to solve. We develop a method to learn an optimal solution approximately by INN in Sec. 5, denoted as Φ_θ. It is guaranteed to be a homeomorphic mapping, i.e., $\Phi_\theta \in \mathcal{H}^n$, with bounded distortion over $\mathcal{X}_\theta$. However, Φ_θ may not lie in $\mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$, i.e., $\Phi_\theta(\mathcal{B}) \neq \mathcal{K}_\theta$, due to INN approximation error, defined as $\mathrm{d}_{\mathrm{H}}(\Phi_\theta(\partial \mathcal{B}), \partial \mathcal{K}_\theta)$[4,5]. Nevertheless, if Φ_θ is also valid, the second step in the HP framework guarantees to "project" an infeasible NN predicted solution back to the feasible set $\mathcal{K}_\theta$.

Definition 4.3 (Valid mapping). The INN approximated mapping $\Phi_\theta \in \mathcal{H}^n$ is valid for $\mathcal{K}_\theta$ if $\Phi_\theta(0) \in \mathcal{K}_\theta$, i.e., it maps the origin in the unit ball to a feasible point in $\mathcal{K}_\theta$.
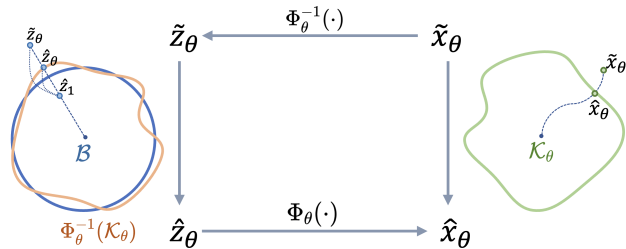

Figure 3: Illustration of homeomorphic bisection.

---

[3]Here + denotes the Minkowski sum between two sets, defined as $\mathcal{X} + \mathcal{Y} = \{x + y | x \in \mathcal{X}, \ y \in \mathcal{Y}\}$.

[4]$\mathrm{d}_{\mathrm{H}}(\cdot, \cdot)$ represents the Hausdorff distance between two sets, defined as $\mathrm{d}_{\mathrm{H}}(\mathcal{X}, \mathcal{Y}) = \max\{\mathrm{d}_{\mathrm{h}}(\mathcal{X}, \mathcal{Y}), \mathrm{d}_{\mathrm{h}}(\mathcal{Y}, \mathcal{X})\}$, where $\mathrm{d}_{\mathrm{h}}(\mathcal{X}, \mathcal{Y}) = \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} \{\|x - y\|\}$.

[5]Here ∂ indicates the boundary of a set.

4.2. Homeomorphic bisection

Suppose for an input θ, we are given an infeasible NN solution $\tilde{x}_\theta \notin \mathcal{K}_\theta$ and a valid INN mapping Φ_θ. Then we perform homeomorphic bisection to recover a feasible solution $\hat{x}_\theta$ as:

$$\hat{x}_\theta = \Phi_\theta(\alpha^* \cdot \tilde{z}_\theta), \quad (5)$$

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

As illustrated in Fig. 3, homeomorphic bisection consists of three steps: (i) map $\tilde{x}_\theta$ to the homeomorphic space as $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$ and set $\alpha = 1$, (ii) perform bisection search on α in $[0, 1]$ (trajectory shown as $\tilde{z}_\theta \to \hat{z}_1 \to \hat{z}_\theta$) to find the largest α, denoted as α^*, such that $\hat{x}_\theta = \Phi_\theta(\alpha^* \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta$, and (iii) return the feasible $\hat{x}_\theta$. The pseudo-code is in Alg. 1. Such a low-complexity operation, observed with respect to the constraint set $\mathcal{K}_\theta$, is to search along a curve connecting the infeasible $\tilde{x}_\theta$ and an internal point $\Phi_\theta(0)$ until reaching a boundary feasible point $\hat{x}_\theta$. As to be discussed in Sec. 6, such operation incurs a minor optimality loss as the homeomorphic mapping has a minimized distortion.

Algorithm 1 Homeomorphic bisection to recover feasibility.

Input: Infeasible solution $\tilde{x}_\theta \notin \mathcal{K}_\theta$ and valid mapping Φ_θ.
Output: Feasible solution $\hat{x}_\theta \in \mathcal{K}_\theta$.

1: initialize total iteration steps k
2: calculate $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$, set $n = 0$, $\alpha_l = 0$, $\alpha_u = 1$
3: **while** $n \leq k$ **do**
4:      bisection: $\alpha_n = (\alpha_l + \alpha_u)/2$
5:      candidate: $\hat{z}_{n+1} = \alpha_n \cdot \tilde{z}_\theta$
6:      **if** $\Phi_\theta(\hat{z}_{n+1}) \in \mathcal{K}_\theta$ **then**
7:          increase lower bound: $\alpha_l \leftarrow \alpha_n$
8:      **else**
9:          decrease upper bound: $\alpha_u \leftarrow \alpha_n$
10:      **end if**
11:      $n \leftarrow n + 1$
12: **end while**
13: find optimal $\alpha^* = \alpha_l$ and feasible point $\hat{z}_\theta = \alpha^* \cdot \tilde{z}_\theta$
14: **return** $\hat{x}_\theta = \Phi_\theta(\hat{z}_\theta)$

We make the following **remarks**. First, in the ideal case when $\Phi_\theta = \psi_\theta^*$ and $\mathcal{B} = \Phi_\theta^{-1}(\mathcal{K}_\theta)$, i.e., the INN learns the MDH mapping perfectly, the homeomorphic bisection in (5) is equivalent to projecting $\tilde{z}_\theta$ onto the unit ball boundary, with a closed-form expression $\hat{z}_\theta = \tilde{z}_\theta / \|\tilde{z}_\theta\|$. Such an operation incurs very low complexity, thanks to the unit ball's geometrical structure. Since $\Phi_\theta = \psi_\theta^*$ is an MDH mapping, the corresponding $\hat{x}_\theta = \Phi_\theta(\hat{z}_\theta)$ must be on the boundary of the feasible set $\mathcal{K}_\theta$, and the projection distance $\|\tilde{x}_\theta - \hat{x}_\theta\|$ is small because (i) Φ_θ has a minimum distortion and (ii) $\hat{z}_\theta$ is the closest point in the unit ball to $\tilde{z}_\theta$. These observations provide an intuitive justification for the complexity and performance of our framework.

Second, in practice, the INN may not learn ψ_θ^* exactly, i.e., $\mathcal{B} \neq \Phi_\theta^{-1}(\mathcal{K}_\theta)$ and the distortion $\mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$ is not the minimum. In fact, we may not know the exact shape of $\Phi_\theta^{-1}(\mathcal{K}_\theta)$ to project $\tilde{z}_\theta$ to its boundary directly. Instead, we perform the operation in (5) to mimic the operation in the ideal case described above, i.e., performing bisection between $\tilde{z}_\theta$ and the origin of the unit ball but evaluating the feasibility with respect to $\mathcal{K}_\theta$. As long as the INN mapping is valid and maps the origin of the unit ball to an internal point in $\mathcal{K}_\theta$, such operation is guaranteed to return a feasible point, with minor optimality loss, similar to the discussion for the ideal case. We formally prove the feasibility guarantee, bound the optimality loss, and discuss the complexity of imperfect INN learning in Sec. 6.

5. Learning θ-dependent MDH mappings

In this section, we propose an unsupervised learning method to train one conditional INN to learn the MDH mapping ψ_θ^* for every $\theta \in \Theta$. We first introduce INN for learning homeomorphic mappings. We then reformulate the MDH mapping problem in (4) and present the INN training procedure.

5.1. INN for homeomorphic mappings

INN is essentially an invertible NN owing to its design, such that it is differentiable in forward and inverse directions (Papamakarios et al., 2021). It is thus a homeomorphic mapping with tunable parameters; see Appendix C for more INN background. It is well known that feed-forward neural networks can approximate any continuous mapping arbitrarily well (Hornik et al., 1989). Similarly, INN can also learn any (piecewise) differentiable homeomorphic mapping arbitrarily well given a sufficient number of neurons (Teshima et al., 2020; Ishikawa et al., 2022). Therefore, we can utilize INN to represent the homeomorphic mapping and train it to approximate an optimal MDH mapping.

Further, we do not need to train separate INN Φ_θ for different input parameters θ. Instead, we can leverage the conditional INN (Winkler et al., 2019; Lyu et al., 2022), which also takes θ as input, to learn the augmented homeomorphism $\psi : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$ such that $\forall \theta \in \Theta$, $[\mathcal{K}_\theta, \theta] = \psi([\mathcal{B}, \theta])$. When given a new θ, we have a corresponding homeomorphism $\psi_\theta \in \mathcal{H}^n$ such that $\mathcal{K}_\theta = \psi_\theta(\mathcal{B})$.

For ease of discussion later, we denote an m-layer INN as $\Phi_\theta = \Phi_\theta^m \circ ... \circ \Phi_\theta^l \circ ... \circ \Phi_\theta^1$, where each layer is an affine coupling layer or an invertible linear layer (Papamakarios et al., 2021). Such a composition is also a universal differentiable homeomorphism approximator (Teshima et al., 2020). Further, for these layers, owing to their invertible design, the singular values for the Jacobian matrix of Φ_θ^l exist everywhere over its support set and have a closed-form expression by its tunable parameters; see Appendix C for detailed for-

mulations. As will become clear later, such closed-form singular values bring convenience for distortion approximation. We then denote the sorted singular values for the Jacobian matrix J of INN as $\sigma_1(\mathrm{J}) \geq ... \geq \sigma_n(\mathrm{J}) > 0$.

5.2. Reformulation of homeomorphism constraint

To facilitate INN learning, we first reformulate the MDH mapping problem in (4). We have the following understanding of its homeomorphism constraint $\psi_\theta \in \mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$ such that $\psi_\theta(\mathcal{B}) = \mathcal{K}_\theta$.

Proposition 5.1. *The feasible set $\mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$ is equivalent to the set of optimal solutions of the problem:*

$$\max_{\psi_\theta \in \mathcal{H}^n} \ \log \mathrm{V}(\psi_\theta(\mathcal{B})) \quad \text{s.t. } \psi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta. \qquad (6)$$

The complete proof is in Appendix E.

$\mathrm{V}(\psi_\theta(\mathcal{B}))$ evaluates the volume of set $\psi_\theta(\mathcal{B})$. The constraint means that the set $\psi_\theta(\mathcal{B})$ is a subset of $\mathcal{K}_\theta$. Intuitively, the proposition says that any feasible homeomorphic mapping must maximize the volume of mapped set $\psi_\theta(\mathcal{B})$ while keeping it within the constraint set $\mathcal{K}_\theta$, and vice versa.

Thus, the MDH mapping problem in (4) is equivalent to the following bi-level problem of minimizing the distortion among the optimal solutions of the problem in (6):

$$\min_{\psi_\theta \in \mathcal{H}^n} \ \log \mathrm{D}(\psi_\theta^{-1}, \mathcal{X}_\theta) \qquad (7)$$

$$\text{s.t. } \psi_\theta \in \arg\max\{\text{Problem in (6)}\}. \qquad (8)$$

As will become clear later, such a reformulation opens the door for unsupervised learning for INN training.

5.3. Unsupervised INN training

We employ the following loss function and maximize it to train an INN Φ_θ with m layers for learning an optimal solution to the problem in (7)-(8) in an unsupervised manner:

$$\mathcal{L}(\Phi_\theta) = \widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B})) - \lambda_1 \mathrm{P}(\Phi_\theta(\mathcal{B})) - \lambda_2 \widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta), \quad (9)$$

where λ_1 and λ_2 are positive coefficients to balance among the three terms. $\widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B}))$ is a computable approximation of the log-volume term $\log \mathrm{V}(\Phi_\theta(\mathcal{B}))$ in (6) as:

$$\widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B})) = \frac{1}{\mathrm{V}(\mathcal{B})} \int_{\mathcal{B}} \sum_{k=1}^{n} \sum_{l=1}^{m} \log \sigma_k(\mathrm{J}_{\Phi_\theta^l}(z^l)) \mathrm{d}z$$
$$+ \log \mathrm{V}(\mathcal{B}), \qquad (10)$$

where $z^l = \Phi_\theta^{l-1}(z^{l-1})$ for $l = 2, .., m$, and $z^1 = z \in \mathcal{B}$, $\mathrm{J}_{\Phi_\theta^l}(z^l)$ denotes the Jacobian matrix of $\Phi_\theta^l(\cdot)$ at z^l, and $\sigma_k(\cdot)$ returns the k-th largest singular value of a matrix, which has a closed-form expression as discussed earlier in Sec. 5.1.

$\mathrm{P}(\Phi_\theta(\mathcal{B}))$ is the penalty term for representing the constraint violation of $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$ in (6) as:

$$\mathrm{P}(\Phi_\theta(\mathcal{B})) = \int_\mathcal{B} \|\operatorname{ReLU}(\mathbf{g}(\Phi_\theta(z), \theta))\|_1 \mathrm{d}z, \qquad (11)$$

where $\mathbf{g}(\Phi_\theta(z), \theta)$ calculates the residual for each inequality constraint as $[g_1(\Phi_\theta(z), \theta), \dots, g_{n_{\text{ineq}}}(\Phi_\theta(z), \theta)]$. $\widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$ is a computable approximation of the log-distortion term $\log \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$ in (7) as:

$$\widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta) = \sup_{z \in \mathcal{Z}_\theta} \left\{ \sum_{l=1}^m \log \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l)) \right\}$$
$$- \inf_{z \in \mathcal{Z}_\theta} \left\{ \sum_{l=1}^m \log \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l)) \right\}, \qquad (12)$$

where $z^l = \Phi_\theta^{l-1}(z^{l-1})$ for $l = 2, .., m$, and $z^1 = z \in \mathcal{Z}_\theta = \Phi_\theta^{-1}(\mathcal{X}_\theta)$.

We have the following observations for the approximations.

Proposition 5.2. *The two approximation terms in (10) and (12) satisfy* $\log \mathrm{V}(\Phi_\theta(\mathcal{B}) \geq \widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B}))$ *and* $\log \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta) \leq \widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$. *The complete proof is in Appendix E.*

The above proposition implies that the loss function in (9) is actually a lower bound to the Lagrangian of the problem in (7)-(8). Therefore, we can maximize the loss function in (9) to approximate the MDH mapping under the equivalent reformulation in (7)-(8). Further, to train one conditional INN $\Phi \in \mathcal{H}^{n+d}$ to learn the θ-dependent MDH mappings for any $\theta \in \Theta$, we generalize the loss in (9) to

$$\mathcal{L}(\Phi) = \mathbb{E}_\theta[\mathcal{L}(\Phi_\theta)], \qquad (13)$$

where $\theta \in \Theta$ is uniformly sampled. For the INN training, we prepare quasi Monte Carlo (QMC)[6] samples $\{z_i\}_{i=1}^N \subset \mathcal{B}$ to approximate the integration in (10) and (11). When evaluating the distortion in (12), since we may not know $\mathcal{Z}_\theta$ in advance, we sample from $\mathcal{Z}_\theta = \Phi_\theta^{-1}(\mathcal{X}_\theta) \subset \mathcal{B}(0, R)$ over a larger ball as $\{R \cdot z_i\}_{i=1}^N$, where $R \geq 1$ is a hyperparameter as discussed in Appendix E.3. In each iteration, we sample a batch of collected data and employ the Adam optimizer to maximize the loss function $\mathcal{L}(\Phi)$, similar to training standard NNs (Kingma & Ba, 2014).

6. Performance analysis

In this section, we formally prove the feasibility guarantee and bound the optimality loss of homeomorphic projection. We also characterize its run-time complexity and a condition for the trained INN to be universally valid over the input-parameter set. Finally, we discuss its training complexity, scalability, and limitations.

---

[6]The integration error for the QMC approach is $\mathcal{O}\left((\log N)^{n-1}/N\right)$, which is faster in the rate of convergence than Monte Carlo using a pseudorandom sequence (Dick & Pillichshammer, 2010).

6.1. Feasibility, optimality, and run-time complexity

Theorem 1. *For compact constraint set $\mathcal{K}_\theta$ of diameter $\operatorname{diam}(\mathcal{K}_\theta)$, given an infeasible NN prediction $\tilde{x}_\theta = F(\theta) \notin \mathcal{K}_\theta$ with bounded prediction error $\epsilon_{\text{pre}} = \sup_{\theta \in \Theta} \|F(\theta) - x_\theta^*\|$, and a valid m-layer trained INN mapping Φ_θ with bounded approximation error $\epsilon_{\text{inn}} = \sup_{\theta \in \Theta} \{\mathrm{d}_\mathrm{H}(\Phi_\theta(\partial\mathcal{B}), \partial\mathcal{K}_\theta)\}$, the bisection procedure shown in Alg. 1 with maximum k steps will return a solution $\hat{x}_\theta^k$ such that:*

- *it is guaranteed to be feasible, i.e., $\hat{x}_\theta^k \in \mathcal{K}_\theta$;*

- *it has a bounded optimality loss as*

  $$\|\hat{x}_\theta^k - x_\theta^*\| \leq \epsilon_{\text{pre}} + \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pre}} + \epsilon_{\text{bis}}^k),$$

  where $\mathcal{Y}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pre}}, \epsilon_{\text{inn}}\})$ and $\epsilon_{\text{bis}}^k = 2^{-k}(\operatorname{diam}(\mathcal{K}_\theta) + \epsilon_{\text{pre}})$;

- *it has a run-time complexity as $\mathcal{O}(k(mn^2 + G))$, where G is the complexity for verifying the inequality constraints.*

The complete proof is in Appendix F.

First, given a valid INN, the bisection algorithm always returns a feasible solution because there exists at least one feasible solution along the curve connecting $\Phi_\theta(0)$ and $\tilde{x}_\theta$, as shown in Fig. 3. Second, the optimality gap depends on the prediction error ϵ_{pre}, the approximation error ϵ_{inn}, the distortion $\mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)$, and the k-step bisection-induced error ϵ_{bis}^k. The prediction error ϵ_{pre} is dominated by the given predictor. The approximation error ϵ_{inn} evaluates the quality of the trained INN in reaching the constraint $\mathcal{K}_\theta = \Phi_\theta(\mathcal{B})$. Therefore, our training scheme attempts to minimize it under the reformulation in Prop. 5.1. The distortion in our training scheme is evaluated and regularized over the set $\mathcal{B}(0, R)$, where $R \geq 1$ is a hyperparameter without knowing $\mathcal{Y}_\theta$ in advance. However, when the prediction and approximation errors are small such that the infeasible solution $\tilde{z}_\theta$ in the homeomorphic space is near $\mathcal{B}$, the bisection algorithm is robust under different selections of the hyperparameter R. Nevertheless, considering a poor-quality NN predictor, we can sample and regulate the distortion over a larger ball to reduce the optimality gap. The bisection error ϵ_{bis}^k can be exponentially reduced by increasing the maximum steps k, as shown in Alg. 1. Note that the results above are analyzed under the worst cases. In practice, the optimality loss of the homeomorphic projection can be much better.

Under an optimal MDH mapping ψ_θ^* in Problem (4), the approximation error $\epsilon_{\text{inn}} = 0$ and the distortion $\mathrm{D}(\psi_\theta^{-1,*}, \mathcal{Y}_\theta) = \mathrm{D}(\psi_\theta^{-1,*}, \mathcal{X}_\theta)$ is minimized, such that the upper bound of the optimality gap is also minimized. In summary, the distortion and the approximation error play a significant role in the optimality gap, which justifies our design in the MDH mapping problem in (4) and the proposed INN loss function in (9).

The overall run-time complexity, i.e., the number of arithmetic operations, when executing Alg. 1 is $\mathcal{O}(k(mn^2 + G))$, includes the INN forward calculation $\mathcal{O}(mn^2)$ and the constraint calculation $\mathcal{O}(G)$. If the inequality constraint $g_i(x, \theta)$ is a linear function, then $G = n \cdot n_{\text{ineq}}$.

6.2. Universal validity condition of INN

Theorem 2. *Let $\mathcal{D}_1 = \{\theta_i, i = 1, \ldots, M\} \subseteq \Theta$ be an r_c-covering training dataset, i.e., $\forall \theta \in \Theta, \exists \theta^0 \in \mathcal{D}_1$ such that $\|\theta - \theta^0\| \leq r_c$. Suppose the trained INN mapping Φ_θ is valid for the interior of constraint set on dataset $\mathcal{D}_1$, i.e., $\forall \theta^0 \in \mathcal{D}_1, \Phi_{\theta^0}(0) \in \mathcal{K}_{\theta^0}^\circ$. If $(C_0 + C_1)r_c \leq C_2$, then $\forall \theta \in \Theta, \Phi_\theta(0) \in \mathcal{K}_\theta$, i.e., Φ_θ will also be valid for any input parameter in Θ. Here $C_0 = \sup_{\theta_1, \theta_2 \in \Theta, \theta_1 \neq \theta_2} \{d_H(\partial\mathcal{K}_{\theta_1}, \partial\mathcal{K}_{\theta_2})/\|\theta_1 - \theta_2\|\}$, $C_1 = L(\Phi(0, \cdot), \Theta)$, $C_2 = \arg\sup_{r>0}\{\mathcal{B}(\Phi_{\theta^0}(0), r) \subseteq \mathcal{K}_{\theta^0}, \forall \theta^0 \in \mathcal{D}_1\}$.*

The complete proof is in Appendix G.

C_0 represents the "Lipschitz" of the constraint set over Θ, C_1 indicates the Lipschitz of the trained INN mapping over Θ, and C_2 denotes the radius of the largest inner approximation ball for the constraint set under dataset $\mathcal{D}_1$.

Theorem 2 provides a sufficient condition for the trained INN to be universally valid over the entire input-parameter set Θ, which is the premise for Theorem 1. First, we need to make the INN valid for finite samples, i.e., $\Phi_{\theta^0}(0) \in \mathcal{K}_{\theta^0}^\circ$, where $\theta^0 \in \mathcal{D}_1$ in the training dataset. In the empirical study in Sec. 7, we observe that this condition is easy to achieve by proper training. This observation is perhaps not surprising, as we penalize constraint violations $P(\Phi_{\theta^0}(\mathcal{B}))$ in (11) for all points in $\mathcal{B}$. Naturally, the origin $0 \in \mathcal{B}$ is likely mapped to an internal point $\Phi_{\theta^0}(0) \in \mathcal{K}_{\theta^0}^\circ$.

To generalize the valid condition to any input parameter $\theta \in \Theta$, a sufficient condition is $(C_0 + C_1)r_c \leq C_2$. These constants C_0, C_1, and C_2 depend on the geometric structure of the constraint set. For example, if the constraint set is very "thin", i.e., C_2 is small, or the constraint set varies dramatically according to the input parameters, i.e., C_0 is large, we need r_c to be small to satisfy the condition. Meanwhile, r_c is directly related to the data size and covering number, such that the number of collected input parameters for the r_c-covering set, i.e., M, has an order as $\mathcal{O}((\text{diam}(\Theta)/r_c)^d)$. Therefore, facing a highly irregular or input-sensitive constraint set, we may need to sample more input parameters θ to train an INN so that the trained INN will be universally valid over the entire input-parameter set Θ.

6.3. Training complexity, scalability, and limitation

First, we need QMC samples in a unit ball $\{z_i\}_{i=1}^N \subset \mathcal{B}$ to approximate the integration in (9) and uniform samples for the input parameters $\{\theta_j\}_{j=1}^M \subset \Theta$ to train the INN.

Both of these can be easily prepared using SciPy (Virtanen et al., 2020). Afterward, we sample a batch of z and θ separately at each iteration and train the INN using the Adam optimizer implemented in PyTorch (Kingma & Ba, 2014; Paszke et al., 2019). Due to the closed-form expressions of (13) by INN parameters, the training computation depends on the forward-backward propagation of the INN, which can be executed efficiently on a GPU.

The scalability of the HP framework depends on the INN structure, where both input and output have dimensions corresponding to the constraint set, and only invertible layers can be applied. Existing works have demonstrated its scalability, especially in generative models where both input and output are high-dimensional matrices (Kingma & Dhariwal, 2018; Papamakarios et al., 2021). Nevertheless, more flexible network structures, such as Neural ODEs (Chen et al., 2018), will be explored in future work. The limitation of the HP framework also lies in Assumption 1, which assumes the constraint set to be homeomorphic to a unit ball. To consider more general constraints, such as disconnected sets or manifold constraints, we discuss them in Appendix B.

7. Numerical Experiments

We carry out simulations to (i) evaluate whether our proposed INN unsupervised-learning scheme can approximate MDH mappings between constraint sets and a unit ball, and (ii) compare the performance of homeomorphic projection with existing methods in ensuring NN solution feasibility for constrained optimization problems, including non-convex AC-OPF problems in power grid operation. The detailed setting of hyper-parameters, INN implementation, and NN predictor implementation are in Appendix H.

7.1. Approximating MDH mappings

We first investigate the learning of MDH mapping Φ for a 2-dimension constraint approximation. The parametric constraint set is defined by a quadratic inequality as:

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

where $Q \in \mathbb{R}^{2\times2}, q \in \mathbb{R}^2, b \in \mathbb{R}$. By changing the input parameters θ such that assumption 1 is valid, the constraint set $\mathcal{K}_\theta$ can be convex or non-convex.

We train two MDH mappings from different norm ball $\mathcal{B}$ to the parametric constraint set $\mathcal{K}_\theta$ following the loss function in (13). The log-volume, constraint violation, and average log-distortion during iterations are shown in Figure 4(a) and 5(a). We find that the volume is maximized, and the constraint violation is converged to zero, which shows that the approximated homeomorphism between two sets is achieved. The log-distortion decreases to 0 over iterations, demonstrating the effectiveness of our training scheme.
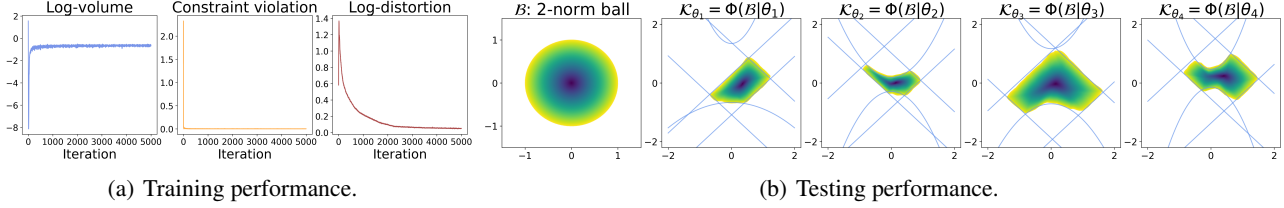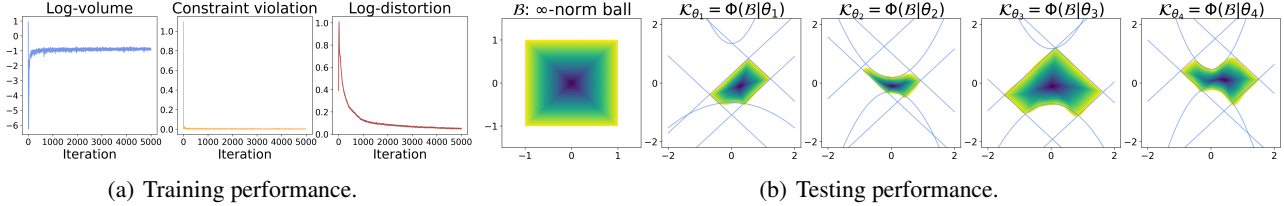
(a) Training performance.

(b) Testing performance.

Figure 4: Training the INN approximated MDH mapping from 2-norm ball to constraint set.

(a) Training performance.

(b) Testing performance.

Figure 5: Training the INN approximated MDH mapping from ∞-norm ball to constraint set.

After training, we visualize the constraint approximation $\Phi_\theta(\mathcal{B})$ for different input parameters θ in the test dataset, as shown in Figure 4(b) and 5(b). First, we find the mapped set well approximates the true constraint set (blue curve). Even the constraint $\mathcal{K}_\theta$ under different input parameters has widely varying geometric structure, the trained INN Φ still shows good generalization ability to fit the constraint set. Second, it does not show obvious differences between ∞-norm ball and $2-$norm ball to approximate the complex constraints. However, for the high-dimension case, we will select the ∞-norm ball (e.g., unit cube) since it is easy to conduct QMC sampling and the cube has an invariant volume as 1. Last, the mapped data point is equally distributed in the constraint set with low distortion, where the color represents the norm of data points in the norm ball $\mathcal{B}$.

In summary, the INN Approximated MDH mapping Φ_θ derives a well-fitted constraint set under different input parameters through the proposed training scheme.

7.2. Ensuring NN solution feasibility

We then cope with the constrained optimization problem with the HP framework. We select three numerical cases: QP, convex QCQP, and SDP, and one real-world case: AC-OPF, which is a non-convex problem. With different input parameters θ, the iterative solver solves the optimal solution x_θ^* as the training dataset. The NN predictor learns the mapping between the input parameters and optimal solutions. Detailed experimental settings of these problems are in Appendix H. The NN-based schemes use reconstruction techniques to ensure the feasibility of equality constraints (Pan et al., 2019; Donti et al., 2021), as discussed in Appendix A. To deal with the infeasible predictions under inequality constraints, we compare the proposed HP framework with the following approaches:

1) **Optimizer**: for convex optimization, we use CVXPY (Diamond & Boyd, 2016) to solve the optimal solution. For AC-OPF problems, we adopt PYPOWER as the specialized solver (Zimmerman et al., 1997; Brown et al., 2017); 2) **NN**: it directly maps input parameter to the solution using a fully connected NN; 3) **Proj**: the predicted solution by NN may be infeasible, so the projection is adopted for the post-processing. The projection problem is solved with the iterative Optimizer; 4) **WS**: The infeasible prediction of NN is regarded as the warm-start initialized solution for the iterative Optimizer; 5) **D-Proj**: this is proposed in DC3 (Donti et al., 2021), which applies gradient descent with implicit function theorem to conduct projection in a differentiable manner; 6) **H-Proj**: for the infeasible solution of NN, we apply bisection in Alg. 1 to recover the feasibility.

The criteria include 1) **Feasibility**: we calculate the percentage of feasible instances out of 1,024 testing instances and the average constraint violation; 2) **Optimality**: the solution error and objective error evaluate the optimality of predicted solutions, including all instances and infeasible instances; 3) **Speedup**: we record the total inference time (NN inference time + post-processing time) of testing instances and calculate the speedup compared with the iterative optimizer. The performances over four constraint optimization problems are shown in Table 2. We have the following observations.

First, existing NN predictors cannot guarantee the feasibility of the predicted solution due to prediction errors. Different post-processing approaches are used to recover feasibility. The proposed H-Proj achieves a **100%** feasibility rate for both convex and non-convex constraint sets, under testing input parameters. This demonstrates the effectiveness of the proposed training scheme and the bisection algorithm. In contrast, the gradient-based D-Proj method cannot guarantee feasibility and is highly sensitive to the stepsize.

Table 2: Performance in convex and non-convex constrained optimization

	Feasibility			Optimality				Speedup	
	feas. rate	ineq. vio.	eq. vio.	sol. err.	infeas. sol. err.	obj. err.	infeas. obj. err.	Total	Post.
	%	1-norm	1-norm	%	%	%	%	$\times$	$\times$
QP: $n = 200$, $d = 100$, $n_{eq} = 100$, $n_{ineq} = 100$									
NN	64.26	0.189	0	5.99	6.12	1.91	1.81	112269.3	–
NN+WS	100	0	0	3.8	0	1.26	0	2.9	1
NN+Proj	100	0	0	5.99	6.11	1.91	1.83	17.6	6.3
NN+D-Proj	67.09	0.037	0	5.99	6.12	1.9	1.8	1170	421.9
NN+H-Proj	100	0	0	6.22	6.75	2.10	2.36	700.8	250.1
Convex QCQP: $n = 200$, $d = 100$, $n_{eq} = 100$, $n_{ineq} = 100$									
NN	54.49	0.163	0	8.16	8.23	3.05	2.96	795657.1	–
NN+WS	100	0	0	4.41	0	1.7	0	2.1	1
NN+Proj	100	0	0	8.15	8.23	3.07	3	2.1	1
NN+D-Proj	56.54	0.023	0	8.15	8.21	3.06	2.98	10.8	4.9
NN+H-Proj	100	0	0	8.36	8.67	3.33	3.58	1618.5	738.8
SDP: $n = 15 \times 15$, $d = 100$, $n_{eq} = 100$, $n_{ineq} = 1$									
NN	74.02	11.43	0	6.77	6.99	4.08	3.7	21440.2	–
NN+WS	100	0	0	4.96	0	3.12	0	1.5	0.4
NN+Proj	100	0	0	6.60	6.31	4.43	5.06	1.5	0.4
NN+D-Proj	87.7	5.69	0	6.76	6.94	4.08	3.7	2.6	0.7
NN+H-Proj	100	0	0	7.49	9.76	4.94	7.03	87.6	22.8
118-node AC-OPF: $n = 344$, $d = 236$, $n_{eq} = 236$, $n_{ineq} = 452$									
NN	73.24	0.006	0	1.27	1.23	0.24	0.23	178.2	–
NN+WS	100	0	0	0.94	0	0.18	0	3.6	1
NN+Proj	100	0	0	1.55	2.31	0.24	0.23	3.8	1
NN+D-Proj	87.79	0.0001	0	1.26	1.23	0.24	0.23	4.9	1.4
NN+H-Proj	100	0	0	1.41	1.78	0.34	0.63	24.6	7.6

Second, H-Proj can recover feasibility with minor optimality loss even under a large prediction error. In other words, the main optimality gap is caused by the NN predictor, and our H-Proj introduces minor projection errors to find a feasible solution near the infeasible NN prediction.

Last, the speedup of the bisection operation is competitive compared to other post-processing approaches in all testing cases. Although WS and Proj methods can achieve feasibility and have a minor optimality gap, they have a poor speedup due to the reliance on iterative solvers. The D-Proj method also does not have the advantage of speeding up because of the expensive gradient computation for complex constraints such as quadratic constraints in convex QCQP.

Overall, the HP framework guarantees solution feasibility with respect to both convex and non-convex constraint sets, with considerable speedup and minor optimality loss.

8. Conclusions

We propose *homeomorphic projection* as the first scheme in the literature that (i) guarantees NN solution feasibility for optimization over a general set homeomorphic to a unit ball, covering all compact convex sets and certain classes of nonconvex sets, (ii) incurs low run-time complexity, and (iii) attains bounded optimality loss. Our design leverages the universal approximation capability of INN to learn a minimum distortion homeomorphic mapping between the constraint set and a unit ball. We then perform a bisection operation concerning the unit ball so that the INN-mapped final solution is feasible with respect to the constraint set. We prove the feasibility guarantee and bound the optimality loss. Simulation results corroborate our analysis and show that homeomorphic projection outperforms existing methods. Future directions include generalizing the approach to optimization over multiple disjoint constraint sets.

Acknowledgements

References

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, J. Z. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

Agrawal, A., Ali, A., Boyd, S., et al. Minimum-distortion embedding. *Foundations and Trends® in Machine Learning*, 14(3):211–378, 2021.

Amos, B. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022.

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

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582. PMLR, 2019.

Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R., and Jacobsen, J.-H. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 1792–1800. PMLR, 2021.

Brown, T., Hörsch, J., and Schlachtberger, D. Pypsa: Python for power system analysis. *arXiv preprint arXiv:1707.09913*, 2017.

Chen, B., Donti, P. L., Baker, K., Kolter, J. Z., and Bergés, M. Enforcing policy feasibility constraints through differentiable projection for energy optimization. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pp. 199–210, 2021a.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Chen, T., Chen, X., Chen, W., Heaton, H., Liu, J., Wang, Z., and Yin, W. Learning to optimize: A primer and a benchmark. *arXiv preprint arXiv:2103.12828*, 2021b.

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

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

Dick, J. and Pillichshammer, F. *Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration*. Cambridge University Press, 2010.

Donti, P. L., Rolnick, D., and Kolter, J. Z. Dc3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*, 2021.

Fioretto, F., Mak, T. W., and Van Hentenryck, P. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 630–637, 2020.

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

Geschke, S. Convex open subsets of rn are homeomorphic to n-dimensional open balls. *Preprint, http://relaunch. hcm. uni-bonn. de/fileadmin/geschke/papers/ConvexOpen. pdf*, 2012.

Guha, N., Wang, Z., Wytock, M., and Majumdar, A. Machine learning for ac optimal power flow. *arXiv preprint arXiv:1910.08842*, 2019.

Heaton, H., Wu Fung, S., Gibali, A., and Yin, W. Feasibility-based fixed point networks. *Fixed Point Theory and Algorithms for Sciences and Engineering*, 2021(1):1–19, 2021.

Heaton, H., Fung, S. W., Lin, A. T., Osher, S., and Yin, W. Wasserstein-based projections with applications to inverse problems. *SIAM Journal on Mathematics of Data Science*, 4(2):581–603, 2022.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2078–2087. PMLR, 2018.

Ishikawa, I., Teshima, T., Tojo, K., Oono, K., Ikeda, M., and Sugiyama, M. Universal approximation property of invertible neural networks. *arXiv preprint arXiv:2204.07415*, 2022.

Khalil, E., Le Bodic, P., Song, L., Nemhauser, G., and Dilkina, B. Learning to branch in mixed integer programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

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

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Kotary, J., Fioretto, F., Van Hentenryck, P., and Wilder, B. End-to-end constrained optimization learning: A survey. *arXiv preprint arXiv:2103.16378*, 2021.

Kratsios, A., Zamanlooy, B., Liu, T., and Dokmanić, I. Universal approximation under constraints is possible with transformers. *arXiv preprint arXiv:2110.03303*, 2021.

Lee, J. M. Smooth manifolds. In *Introduction to smooth manifolds*, pp. 1–31. Springer, 2013.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Liu, H.-T. D., Williams, F., Jacobson, A., Fidler, S., and Litany, O. Learning smooth neural functions via lipschitz regularization. *arXiv preprint arXiv:2202.08345*, 2022.

Lyu, J., Chen, Z., Feng, C., Cun, W., Zhu, S., Geng, Y., Xu, Z., and Chen, Y. Universality of parametric coupling flows over parametric diffeomorphisms. *arXiv preprint arXiv:2202.02906*, 2022.

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

Nellikkath, R. and Chatzivasileiadis, S. Physics-informed neural networks for ac optimal power flow. *arXiv preprint arXiv:2110.02672*, 2021b.

Pan, X., Zhao, T., and Chen, M. Deepopf: A deep neural network approach for security-constrained dc optimal power flow. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019.

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

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga,

L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Sapkota, S. and Bhattarai, B. Input invex neural network. *arXiv preprint arXiv:2106.08748*, 2021.

Schmidt, P., Born, J., Campen, M., and Kobbelt, L. Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019.

Tabas, D. and Zhang, B. Computationally efficient safe reinforcement learning for power systems. *arXiv preprint arXiv:2110.10333*, 2021.

Tabas, D. and Zhang, B. Safe and efficient model predictive control using neural networks: An interior point approach. *arXiv preprint arXiv:2203.12196*, 2022.

Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., and Sugiyama, M. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.

Virmaux, A. and Scaman, K. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Winkler, C., Worrall, D., Hoogeboom, E., and Welling, M. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

Xia, Y. and Wang, J. A recurrent neural network for solving linear projection equations. *Neural Networks*, 13(3):337–350, 2000.

Xiao, C., Zhong, P., and Zheng, C. Bourgan: Generative networks with metric embeddings. *Advances in Neural Information Processing Systems*, 31, 2018.

Zamzam, A. S. and Baker, K. Learning optimal solutions for extremely fast ac optimal power flow. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6. IEEE, 2020.

Zhang, H., Gao, X., Unterman, J., and Arodz, T. Approximation capabilities of neural odes and invertible residual networks. In *International Conference on Machine Learning*, pp. 11086–11095. PMLR, 2020.

Zhang, L., Chen, Y., and Zhang, B. A convex neural network solver for dcopf with generalization guarantees. *IEEE Transactions on Control of Network Systems*, 2021.

Zhao, T., Pan, X., Chen, M., and Low, S. H. 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.

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

Zimmerman, R. D., Murillo-Sánchez, C. E., and Gan, D. Matpower. *PSERC.[Online]. Software Available at: http://www. pserc. cornell. edu/matpower*, 1997.
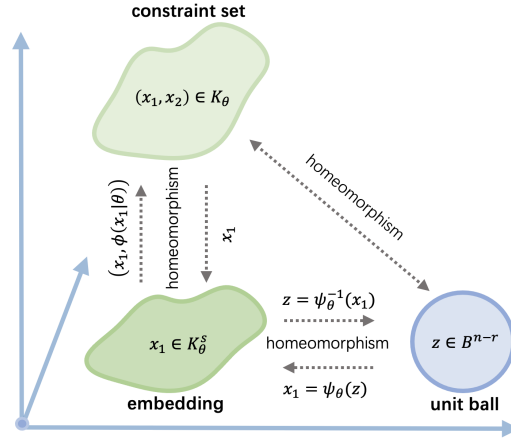
A. Handling equality constraints

Figure 6: Constraint homeomorphism with equality constraints.

Consider the following constraint set $\mathcal{K}_\theta$ defined by inequality and equality constraints:

$$\mathcal{K}_\theta = \{x \in \mathbb{R}^n | \mathbf{h}(x, \theta) = 0, g_1(x, \theta) \leq 0, ..., g_{n_{\text{ineq}}}(x, \theta) \leq 0\}, \tag{14}$$

where $\mathbf{h}(x, \theta) : \mathbb{R}^n \to \mathbb{R}^r$ is continuous with respect to x and θ. Given input θ, we use $\mathbf{h}_\theta(x)$ for convenience.

By assuming the constant rank of equality constraint as:

$$\text{rank}(\mathrm{J}_{\mathbf{h}_\theta}(x)) = r, \quad \forall \theta \in \Theta \text{ and } \forall x \in \mathcal{K}_\theta. \tag{15}$$

The equality constraint with constant rank implies $\mathcal{K}_\theta$ is Euclidean of dimension[7] of $n - r$ by the *Constant-Rank Level Set Theorem* (Lee, 2013). In other words, we can use a subset of decision variables $x_1 \in \mathbb{R}^{n-r}$ and reconstruct full decision variable $[x_1, x_2] \in \mathbb{R}^n$ via the equality constraint as shown in Figure 6, where $x_2 = \phi_\theta(x_1)$ and $\mathbf{h}_\theta([x_1, \phi_\theta(x_1)]) = 0$. Such a reconstruction process ensuring the feasibility of equality constraint is widely used in the literature (Pan et al., 2019; Zamzam & Baker, 2020; Donti et al., 2021).

We then denote the equivalent constraint set with equality reconstruction as $\mathcal{K}_\theta^s = \{x \in \mathbb{R}^{n-r} | g_1([x_1, \phi_\theta(x_1)], \theta) \leq 0, ..., g_{n_{\text{ineq}}}([x_1, \phi_\theta(x_1)], \theta) \leq 0\}$. Under assumption 1, it is homeomorphic to the original constraint set $\mathcal{K}_\theta \cong \mathcal{K}_\theta^s$ (Lee, 2013). The forward and inverse mappings of homeomorphism are as follows:

$$[x_1, x_2] \in \mathcal{K}_\theta \to x_1 \in \mathcal{K}_\theta^s, \tag{16}$$

$$x_1 \in \mathcal{K}_\theta^s \to [x_1, \phi_\theta(x_1)] \in \mathcal{K}_\theta. \tag{17}$$

We give two examples to illustrate such an equality completion/reconstruction process:

Case 1: Linear equality

Let us consider the following equality constraint as $\{x \in \mathbb{R}^n | Ax = \theta, A \in \mathbb{R}^{r \times n}, \theta \in \mathbb{R}^r\}$, where x is the decision variable and θ is the input parameter. Without loss of generality, we assume the rank of matrix A as $\text{rank}(A) = r$. We then partition the decision variable into two groups as $x_1 \in \mathbb{R}^{n-r}$ and $x_2 \in \mathbb{R}^r$. Matrix A is correspondingly partitioned as $A = [A_1, A_2]$, where $A_1 \in \mathbb{R}^{r \times (n-r)}$ and $A_2 \in \mathbb{R}^{r \times r}$. The equality constraint is also partitioned as $A_1 x_1 + A_2 x_2 = \theta$.

The reconstruction is to say, we only need a subset of variable x_1 to ensure the equality constraints as:

$$x_2 = \phi_\theta(x_1) = A_2^{-1}(\theta - A_1 x_1). \tag{18}$$

There exist multiple partitions of x_1 and x_2 leading A_2 to have the full rank of r. Empirically, we select the partition leading A_2 with a large determinant. After taking the inverse of A_2, x_2 is insensitive to x_1, which is beneficial to the prediction and

---

[7]If an open set $\mathcal{K}_\theta$ is Euclidean of dimension, then every point $x \in \mathcal{K}_\theta$ has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^n$ (Lee, 2013).

error propagation when training the NN predictor and MDH mapping. The Jacobian matrix used for back-propagation is as follows:

$$\mathrm{J}_{\phi_\theta}(x_1) = -A_2^{-1} A_1. \tag{19}$$

Case 2: Non-linear equality

If the equality constraint is defined by the following non-linear equality $\{x \in \mathbb{R}^n | \mathbf{h}(x, \theta) = 0, \theta \in \mathbb{R}^d, \mathbf{h} : \mathbb{R}^n \to \mathbb{R}^r\}$. Similarly, we partition the decision variable as x_1 and x_2, the completion function ϕ_θ is implicitly decided by:

$$\mathbf{h}([x_1, \phi_\theta(x_1)], \theta) = 0. \tag{20}$$

Applying Newton's methods, such a non-linear equation can be solved iteratively. The Jacobian matrix used for back-propagation is as follows:

$$\mathrm{J}_{\phi_\theta}(x_1) = - \mathrm{J}_{\mathbf{h}_\theta}^{-1}(x_2) \, \mathrm{J}_{\mathbf{h}_\theta}(x_1). \tag{21}$$

In summary, we can leverage reconstruction techniques to exploit the equality constraints and reduce the predicted decision variables. So we only need to model the constraint set with inequality constraints. Further, such a reconstruction is differentiable and can be integrated into the training process (Donti et al., 2021).

B. Discussion on disconnected constraint and manifold constraint.

Assumption 1 indicates that the constraint set $\mathcal{K}_\theta$ is topologically equivalent to a unit ball. Under Assumption 1, the constraint set contains (i) all compact convex sets and (ii) certain classes of compact and simply-connected non-convex sets. Thus, we leverage the homeomorphism between them to learn the mapping and conduct projection with low run-time complexity.

However, a general disconnected constraint set $\mathcal{K}_\theta$ may contain multiple ball-homeomorphic subsets as $\mathcal{K}_\theta = \bigcup_{c=1}^N \mathcal{K}_\theta^c$, where $\mathcal{K}_\theta^c \cong \mathcal{B}$. We can still apply the HP framework to learn multiple MDH mappings such that $\mathcal{K}_\theta = \bigcup_{c=1}^N \Phi(\mathcal{B}|\theta, c)$. However, it may be non-trivial to determine the number of disconnected parts in advance.

On the other hand, a general manifold constraint can be formulated as the level set $\mathcal{K}_\theta = \{x \in \mathbb{R}^n | f(x) = 0, f : \mathbb{R}^n \to \mathbb{R}^m\}$ (Lee, 2013). It may not be homomorphic to a unit ball or have a non-constant rank. However, a similar idea can be applied again to learn multiple MDH mappings to cover the original constraint. For example, the sphere constraint $\mathcal{K}_\theta = \{x \in \mathbb{R}^n | \|x\|_2 = 1\}$ has a constant rank of 1, but it is not homeomorphic to a unit ball. After we apply the reconstruction techniques in Appendix A, we have two separated constraint sets as follows: $\mathcal{K}_\theta^1 = \{[x_s \in \mathbb{R}^{n-1}, \sqrt{1 - \|x_s\|_2}] \mid \|x_s\|_2 \leq 1\}$ and $\mathcal{K}_\theta^2 = \{[x_s \in \mathbb{R}^{n-1}, -\sqrt{1 - \|x_s\|_2}] \mid \|x_s\|_2 \leq 1\}$, where both sets are homeomorphic to the $n - 1$ dimensional unit ball.

Therefore, more general constraint sets will be investigated in future work.

C. INN introduction

The INN $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ is a class of neural networks that is a continuous bijection with continuous inverse (homeomorphic mapping). It is a finite composition of invertible layers, where each layer is also a homeomorphic mapping. Let us denote an m-layer INN as: $\Phi = \Phi^m \circ ... \circ \Phi^l \circ ... \circ \Phi^1$.

First, the Jacobian of such a composited mapping can be expressed as

$$\mathrm{J}_\Phi(x) = \prod_{l=1}^m \mathrm{J}_{\Phi^l}(x^l), \tag{22}$$

where $x^l = \Phi^{l-1}(x^{l-1})$ for $l = 2, .., m$ and $x^1 = x$.

Then the Jacobian determinant $\det \mathrm{J}(\cdot)$ of INN can be expressed as:

$$|\det \mathrm{J}_\Phi(x)| = \prod_{l=1}^m |\det \mathrm{J}_{\Phi^l}(x^l)|. \tag{23}$$

For each layer, the Jacobian determinant can be expressed as the multiplication of singular values:

$$|\det \mathrm{J}_{\Phi^l}(x^l)| = \prod_{k=1}^{n} \sigma_k(\mathrm{J}_{\Phi^l}(x^l)), \tag{24}$$

where $\sigma_1(\cdot) \geq ... \geq \sigma_n(\cdot) > 0$ are the sorted singular values of the Jacobian matrix of mapping $\Phi^l(\cdot)$ at x.

Owing to the invertibility of INN, the singular value is non-zero and exists everywhere over the support set of INN (Papamakarios et al., 2021). By the design of each layer, such an invertible transformation has a closed-form expression of singular values, and we give several examples of them in the following sections. For a more comprehensive overview of INN, we refer readers to (Papamakarios et al., 2021)

C.1. Some examples

- **Coupling layer**

  The coupling layer first randomly splits the input into two parts as $x = [x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}]$, and then the invertible transformation is defined as:

$$\text{Forward:} \quad [y_1, y_2] = [x_1, \mathrm{w}(x_1) \cdot x_2 + \mathrm{b}(x_1)], \tag{25}$$
$$\text{Inverse:} \quad [x_1, x_2] = [y_1, (y_2 - \mathrm{b}(y_1))/\mathrm{w}(y_1)], \tag{26}$$

  where $\mathrm{w}(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ and $\mathrm{b}(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ are two normal NNs with learnable parameters, which takes x_1 as input and predicts the weight and bias for the element-wise affine transformation of x_2. Since the transformation is element-wise, the Jacobian matrix is diagonal, and the singular values are 1 and $|\mathrm{w}(x_1)|$. For the conditional layer, we augment the input parameters θ as $w(x_1, \theta), b(x_1, \theta)$.

  One particular case of the coupling layer is the auto-regressive flow (Huang et al., 2018), where $y_i = w(x_{1:i-1})x_i + b(x_{1:i-1})$ and it can be efficiently implemented by masking weight matrix. Therefore, the run-time complexity for the coupling layer or the auto-regressive layer is $\mathcal{O}(n^2)$

- **Invertible 1×1 convolution**

  The invertible convolution layer is an invertible linear layer as:

$$\text{Forward:} \quad y = Wx, \tag{27}$$
$$\text{Inverse:} \quad x = W^{-1}y, \tag{28}$$

  where $W \in \mathbb{R}^{n \times n}$ is an invertible and learnable matrix. Further, by the LU decomposition, the invertible matrix is designed as $W = PL(U + \mathrm{diag}(s))$, where P is a fixed permutation matrix, L is a lower triangular matrix, U is an upper triangular matrix, and $s \in \mathbb{R}^n$ is the diagonal elements. The singular values are then as $|s|$. The run-time complexity of this invertible linear layer is $\mathcal{O}(n^2)$.

- **Actnorm**

  The Actnorm layer can be regarded as a normalization layer, i.e., the input is scaled and translated as:

$$\text{Forward:} \quad y = w \cdot x + b, \tag{29}$$
$$\text{Inverse:} \quad x = (y - b)/w, \tag{30}$$

  where $w, b \in \mathbb{R}^n$ are trainable parameters. The singular value of the Actnorm layer is $|w|$. The run-time complexity of this Actnorm layer is $\mathcal{O}(n)$

C.2. Approximation ability

A neural network (NN) is known as the universal approximator for continuous functions (Hornik et al., 1989; Leshno et al., 1993; Kratsios et al., 2021), i.e., it can approximate any continuous function with arbitrarily small error given a sufficient number of neurons. However, for an INN, to the best of our knowledge, existing works only prove that it is the universal approximator of (piecewise) diffeomorphisms $\mathcal{D}^n$ (differentiable bijections with differentiable inverses) (Zhang et al., 2020; Teshima et al., 2020; Lyu et al., 2022; Ishikawa et al., 2022), which is a subset of homeomorphisms $\mathcal{D}^n \subset \mathcal{H}^n$. Nevertheless, in our empirical study, INNs still show good performance when approximating the homeomorphism between sets.

D. Technical Lemmas

Lemma 1 (Property of distortion). *The distortion for a homeomorphic mapping $\psi \in \mathcal{H}^n$ over the compact set $\mathcal{Z}$ is denoted as $D(\psi, \mathcal{Z})$. Then the following properties hold:*

(1) $D(\psi, \mathcal{Z}) = L(\psi, \mathcal{Z}) L(\psi^{-1}, \mathcal{X})$;

(2) $D(\psi, \mathcal{Z}) = D(\psi^{-1}, \mathcal{X}) = \frac{\sup_{z \in \mathcal{Z}} \{\sigma_1(J_\psi(z))\}}{\inf_{z \in \mathcal{Z}} \{\sigma_n(J_\psi(z))\}} = \frac{\sup_{x \in \mathcal{X}} \{\sigma_1(J_{\psi^{-1}}(x))\}}{\inf_{x \in \mathcal{X}} \{\sigma_n(J_{\psi^{-1}}(x))\}}$;

(3) if $\mathcal{Z}_1 \subseteq \mathcal{Z}_2$, then $D(\psi, \mathcal{Z}_1) \leq D(\psi, \mathcal{Z}_2)$;

where $\mathcal{X} = \psi(\mathcal{Z})$, $L(\cdot, \cdot)$ indicates the Lipschitz constant of a mapping over a set, $\sigma_1(\cdot) \geq ... \geq \sigma_n(\cdot)$ denotes the sorted singular values of the Jacobian matrix J for a mapping at a certain point.

Proof:

(1) According to the definition of distortion in 4.1, The distortion of mapping ψ over set $\mathcal{Z}$ is defined as $D(\psi, \mathcal{Z}) = \kappa_2/\kappa_1$ such that:

$$\kappa_2 = \sup_{z_1, z_2 \in \mathcal{Z}, z_1 \neq z_2} \{\|\psi(z_1) - \psi(z_2)\|/\|z_1 - z_2\|\} = L(\psi, \mathcal{Z}), \tag{31}$$

$$1/\kappa_1 = 1/ \inf_{z_1, z_2 \in \mathcal{Z}, z_1 \neq z_2} \{\|\psi(z_1) - \psi(z_2)\|/\|z_1 - z_2\|\} \tag{32}$$

$$= \sup_{z_1, z_2 \in \mathcal{Z}, z_1 \neq z_2} \{\|z_1 - z_2\|/\|\psi(z_1) - \psi(z_2)\|\} \tag{33}$$

$$= \sup_{x_1, x_2 \in \mathcal{X}, x_1 \neq x_2} \{\|\psi^{-1}(x_1) - \psi^{-1}(x_2)\|/\|x_1 - x_2\|\} = L(\psi^{-1}, \mathcal{X}). \tag{34}$$

Therefore, we have $D(\psi, \mathcal{Z}) = L(\psi, \mathcal{Z}) L(\psi^{-1}, \mathcal{X})$.

(2) Further, we adopt the spectral norm, i.e., the largest singular value, to represent the Lipschitz constant as

$$L(\psi, \mathcal{Z}) = \sup_{z \in \mathcal{Z}} \{\sigma_1(J_\psi(z))\}, \tag{35}$$

$$L(\psi^{-1}, \mathcal{X}) = \sup_{x \in \mathcal{X}} \{\sigma_1(J_{\psi^{-1}}(x))\}. \tag{36}$$

Because of the invertibility of homeomorphic mapping such that $J_\psi^{-1}(z) = J_{\psi^{-1}}(x)$ for $x = \psi(z)$, we have $\sigma_1(J_\psi(z)) = 1/\sigma_n(J_{\psi^{-1}}(x))$ for $x = \psi(z)$. The distortion can be represented as:

$$D(\psi, \mathcal{Z}) = \frac{\sup_{z \in \mathcal{Z}} \{\sigma_1(J_\psi(z))\}}{\inf_{z \in \mathcal{Z}} \{\sigma_n(J_\psi(z))\}} = \frac{\sup_{x \in \mathcal{X}} \{\sigma_1(J_{\psi^{-1}}(x))\}}{\inf_{x \in \mathcal{X}} \{\sigma_n(J_{\psi^{-1}}(x))\}} = D(\psi^{-1}, \mathcal{X}). \tag{37}$$

As a result, we can evaluate the distortion of a homeomorphic mapping ψ over $\mathcal{X}$ or $\mathcal{Z}$ equivalently as $D(\psi, \mathcal{Z}) = D(\psi^{-1}, \mathcal{X})$.

(3) According to Eq. (37), if $\mathcal{Z}_1 \subseteq \mathcal{Z}_2$, then:

$$D(\psi, \mathcal{Z}_1) = \frac{\sup_{z \in \mathcal{Z}_1} \{\sigma_1(J_\psi(z))\}}{\inf_{z \in \mathcal{Z}_1} \{\sigma_n(J_\psi(z))\}} \leq \frac{\sup_{z \in \mathcal{Z}_2} \{\sigma_1(J_\psi(z))\}}{\inf_{z \in \mathcal{Z}_2} \{\sigma_n(J_\psi(z))\}} = D(\psi, \mathcal{Z}_2). \tag{38}$$

Lemma 2 (Property of Hausdroff distance). *The Hausdroff distance between sets is denoted as $d_H(\mathcal{X}, \mathcal{Y}) = \max\{d_h(\mathcal{X}, \mathcal{Y}), d_h(\mathcal{Y}, \mathcal{X})\}$, where the one-side Hausdroff distance is defined as: $d_h(\mathcal{X}, \mathcal{Y}) = \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} \{\|x - y\|\}$. Then the following properties hold:*

(1) $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0, r) \Leftrightarrow d_h(\mathcal{Y}, \mathcal{X}) \leq r$, further, $\mathcal{Y} = \mathcal{X} + \mathcal{B}(0, r) \Rightarrow d_H(\mathcal{Y}, \mathcal{X}) \leq r$;

(2) For a homeomorphic mapping $\psi \in \mathcal{H}^n$, we have $\mathrm{d_h}(\psi(\mathcal{X}), \psi(\mathcal{Y})) \leq \mathrm{L}(\psi, \mathcal{X} \cup \mathcal{Y}) \, \mathrm{d_h}(\mathcal{X}, \mathcal{Y}) \leq \mathrm{L}(\psi, \mathcal{X} \cup \mathcal{Y}) \, \mathrm{d_H}(\mathcal{X}, \mathcal{Y})$;

(3) Let $\mathcal{Y} = \{x \in \mathbb{R}^n \,|\, \|x\|_2 \leq r\}$, then $\mathrm{d_h}(\partial \mathcal{X}, \partial \mathcal{Y}) \geq \sup_{x \in \partial \mathcal{X}} \{|\|x\| - r|\}$.

Proof:

(1) First, to prove $\mathrm{d_h}(\mathcal{Y}, \mathcal{X}) \leq r \;\Rightarrow\; \mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0, r)$, we have:

$$\inf_{x \in \mathcal{X}} \{\|x - y\|\} \leq \sup_{y \in \mathcal{Y}} \inf_{x \in \mathcal{X}} \{\|x - y\|\} = \mathrm{d_h}(\mathcal{Y}, \mathcal{X}) \leq r, \tag{39}$$

such that $\forall y \in \mathcal{Y}$, the distance $\inf_{x \in \mathcal{X}} \{\|x - y\|\} \leq r$. In other words, $\forall y \in \mathcal{Y}$, there exist a $x_0 \in \mathcal{X}$ such that $y \in \mathcal{B}(x_0, r)$, which indicates $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0, r)$.

Secondly, to prove $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0, r) \;\Rightarrow\; \mathrm{d_h}(\mathcal{Y}, \mathcal{X}) \leq r$, we have the Minkowski sum for the set $\mathcal{Y} \subseteq \mathcal{Y}' = \mathcal{X} + \mathcal{B}(0, r)$, where $\mathcal{Y}' = \{y' = x + \vec{r} \mid x \in \mathcal{X}, \|\vec{r}\| = r\}$. Then we have

$$\mathrm{d_h}(\mathcal{Y}, \mathcal{X}) = \sup_{y \in \mathcal{Y}} \inf_{x \in \mathcal{X}} \{\|x - y\|\} \leq \sup_{y \in \mathcal{Y}'} \inf_{x \in \mathcal{X}} \{\|x - y\|\} \tag{40}$$

$$= \sup_{x' \in \mathcal{X}, \vec{r}} \inf_{x \in \mathcal{X}} \{\|x - (x' + \vec{r})\|\} \leq \sup_{x' \in \mathcal{X}, \vec{r}} \{\|x' - (x' + \vec{r})\|\} = r. \tag{41}$$

Therefore, we conclude $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0, r) \;\Leftrightarrow\; \mathrm{d_h}(\mathcal{Y}, \mathcal{X}) \leq r$

Further, if $\mathcal{Y} = \mathcal{X} + \mathcal{B}(0, r)$, where $\mathcal{Y} = \{y = x + \vec{r} \mid x \in \mathcal{X}, \|\vec{r}\| = r\}$, we have:

$$\mathrm{d_h}(\mathcal{X}, \mathcal{Y}) = \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} \{\|x - y\|\} = \sup_{x \in \mathcal{X}} \inf_{x' \in \mathcal{X}, \vec{r}} \{\|x - (x' + \vec{r})\|\} \leq \sup_{x \in \mathcal{X}} \{\|x - (x + \vec{r})\|\} = r. \tag{42}$$

Therefore, we have $\mathrm{d_h}(\mathcal{X}, \mathcal{Y}) \leq r$ and $\mathrm{d_h}(\mathcal{Y}, \mathcal{X}) \leq r$, leading to $\mathrm{d_H}(\mathcal{Y}, \mathcal{X}) \leq r$.

(2) According to the definition of one-side Hausdroff distance, we have:

$$\mathrm{d_h}(\psi(\mathcal{X}), \psi(\mathcal{Y})) = \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} \{\|\psi(x) - \psi(y)\|\} \tag{43}$$

$$\leq \mathrm{L}(\psi, \mathcal{X} \cup \mathcal{Y}) \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} \{\|x - y\|\} \tag{44}$$

$$= \mathrm{L}(\psi, \mathcal{X} \cup \mathcal{Y}) \, \mathrm{d_h}(\mathcal{X}, \mathcal{Y}) \leq \mathrm{L}(\psi, \mathcal{X} \cup \mathcal{Y}) \, \mathrm{d_H}(\mathcal{X}, \mathcal{Y}), \tag{45}$$

where $\mathrm{L}(\cdot, \cdot)$ indicates the Lipschitz constant of a mapping over a set.

(3) According to the definition of one-side Hausdroff distance, we have:

$$\mathrm{d_h}(\partial \mathcal{X}, \partial \mathcal{Y}) = \sup_{x \in \partial \mathcal{X}} \inf_{y \in \partial \mathcal{Y}} \{\|x - y\|\} = \sup_{x \in \partial \mathcal{X}} \{\|x - \mathrm{Proj}_{\partial \mathcal{Y}}(x)\|\} \tag{46}$$

$$\geq \sup_{x \in \partial \mathcal{X}} \{|\|x\| - \|\mathrm{Proj}_{\partial \mathcal{Y}}(x)\||\} \tag{47}$$

$$= \sup_{x \in \partial \mathcal{X}} \{|\|x\| - r|\}. \tag{48}$$

Lemma 3. *For $a, b \in \mathbb{R}^n$ and $\|a\|_2 = \|b\|_2$, if $\lambda \geq 0$, then $\|\lambda a - a\|_2 \leq \|\lambda a - b\|_2$*

Proof:

Let the angle between a and b as β, where $0 \leq \beta \leq \pi$, then we have:

$$\|\lambda a - b\|_2^2 - \|\lambda a - a\|_2^2 = \lambda^2 \|a\|_2^2 + \|b\|_2^2 - 2\lambda \|a\|_2 \|b\|_2 \cos(\beta) - \lambda^2 \|a\|_2^2 - \|a\|_2^2 + 2\lambda \|a\|_2^2 \tag{49}$$

$$= 2\lambda \|a\|_2^2 - 2\lambda \|a\|_2 \|b\|_2 \cos(\beta) \tag{50}$$

$$= 2\lambda \|a\|_2^2 (1 - \cos(\beta)) \geq 0. \tag{51}$$

E. Proof for Propositions

E.1. Reformulation in Proposition 5.1

Proof:

We consider the following reformulation of the set equivalent constraints $\psi_\theta \in \mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$ such that $\mathcal{K}_\theta = \psi_\theta(\mathcal{B})$.

The feasible set $\mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$ of problem in (4) is equivalent to the set of optimal solutions of the problem:

$$\max_{\psi_\theta \in \mathcal{H}^n} \; \log \mathrm{V}(\psi_\theta(\mathcal{B})) \quad \text{s.t.} \; \psi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta. \tag{52}$$

Under Assumption 1, those sets $\mathcal{K}_\theta \cong \psi_\theta(\mathcal{B}) \cong \mathcal{B}$ are topologically equivalent and have non-zero volume in $\mathbb{R}^n$. Therefore, we can define the volume of those sets such as $\mathrm{V}(\mathcal{B}) = \int_\mathcal{B} dz$. Because of the containment constraint $\psi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$, the volume satisfies $\mathrm{V}(\psi_\theta(\mathcal{B})) \leq \mathrm{V}(\mathcal{K}_\theta)$. Therefore, the maximum of $\mathrm{V}(\psi_\theta(\mathcal{B}))$ under constraint $\psi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$ is $\mathrm{V}(\mathcal{K}_\theta)$. The maximum is reached only when $\psi_\theta(\mathcal{B}) = \mathcal{K}_\theta$, which is the homeomorphism constraint.

E.2. Volume estimation in Proposition 5.2

Proof:

We consider a uniform probability distribution over $\mathcal{B}$ as $z \sim \mathrm{Unif}(\mathcal{B})$, where the probability density function is as $p(z) = \frac{1}{\mathrm{V}(\mathcal{B})}$. To approximate the volume $\log \mathrm{V}(\Phi_\theta(\mathcal{B}))$ under an m-layer INN Φ_θ, we apply the Jensen inequality for $\log \frac{\mathrm{V}(\Phi_\theta(\mathcal{B}))}{\mathrm{V}(\mathcal{B})}$:

$$\log \frac{\mathrm{V}(\Phi_\theta(\mathcal{B}))}{\mathrm{V}(\mathcal{B})} = \log \frac{\int_\mathcal{B} |\det \mathrm{J}_{\Phi_\theta}(z)| dz}{\mathrm{V}(\mathcal{B})} \tag{53}$$

$$= \log \mathbb{E}_{z \sim p(z)}[|\det \mathrm{J}_{\Phi_\theta}(z)|] \tag{54}$$

$$\geq \mathbb{E}_{z \sim p(z)}[\log |\det \mathrm{J}_{\Phi_\theta}(z)|] \tag{55}$$

$$= \mathbb{E}_{z \sim p(z)}[\sum_{k=1}^{n} \sum_{l=1}^{m} \log \sigma_k(\mathrm{J}_{\Phi_\theta^l}(z^l))], \tag{56}$$

where $z^l = \Phi_\theta^{l-1}(z^{l-1})$ for $l = 2, .., m$ and the initial value $z^1 = z \sim p(z)$.

We then have a lower bound of the log-volume for the maximization problem as

$$\log \mathrm{V}(\Phi_\theta(\mathcal{B})) \geq \widehat{\mathrm{V}}(\Phi_\theta(B)) = \frac{1}{\mathrm{V}(\mathcal{B})} \int_\mathcal{B} \sum_{k=1}^{n} \sum_{l=1}^{m} \log \sigma_k(\mathrm{J}_{\Phi_\theta^l}(z^l)) dz + \log \mathrm{V}(\mathcal{B}). \tag{57}$$

We use this bound because of numerical stability and the closed-form expression of singular values for INN, as shown in Appendix C.

E.3. Distortion estimation in Proposition 5.2

Proof:

According to Lemma 1, the distortion of m-layer INN Φ_θ can be expressed as the multiplication of the Lipschitz constant as:

$$\log \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta) = \log \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta) + \log \mathrm{L}(\Phi_\theta^{-1}, \mathcal{X}_\theta) \tag{58}$$

$$= \sup_{z \in \mathcal{Z}_\theta} \{\log \sigma_1(\mathrm{J}_{\Phi_\theta}(z))\} + \sup_{x \in \mathcal{X}_\theta} \{\log \sigma_1(\mathrm{J}_{\Phi_\theta^{-1}}(x))\} \tag{59}$$

$$\leq \sup_{z \in \mathcal{Z}_\theta} \{\log \prod_{l=1}^{m} \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} + \sup_{x \in \mathcal{X}_\theta} \{\log \prod_{l=1}^{m} \sigma_1(\mathrm{J}_{\Phi_\theta^{-1,l}}(x^l))\} \tag{60}$$

$$= \sup_{z \in \mathcal{Z}_\theta} \{\log \prod_{l=1}^{m} \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} - \inf_{z \in \mathcal{Z}_\theta} \{\log \prod_{l=1}^{m} \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))\} \tag{61}$$

$$= \sup_{z \in \mathcal{Z}_\theta} \{\sum_{l=1}^{m} \log \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} - \inf_{z \in \mathcal{Z}_\theta} \{\sum_{l=1}^{m} \log \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))\} \tag{62}$$

$$= \widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta), \tag{63}$$

where $z^l = \Phi_\theta^{l-1}(z^{l-1})$ for $l = 2, .., m$ and the initial value is $z^1 = z \in \mathcal{Z}_\theta$, while $x^l = z^{m-l+1}$ and $x^1 = x \in \mathcal{X}_\theta$. The set is defined as $\mathcal{Z}_\theta = \Phi_\theta^{-1}(\mathcal{X}_\theta) = \Phi_\theta^{-1}(\mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\mathrm{pre}}))$.

Inequality in (60) follows the property of spectral norm as $\|AB\| \leq \|A\|\|B\|$, where the singular values for the Jacobian matrix of a composited function are limited by the multiplication of singular values for each layer (Virmaux & Scaman, 2018; Behrmann et al., 2019; 2021). It can be viewed as we try to limit the distortion of each invertible layer such that the distortion of the composition of them is also bounded.

Further, when evaluating distortion over set $\mathcal{Z}_\theta$ in the homeomorphic space, we can bound the set as:

$$\mathrm{d}_{\mathrm{h}}(\partial \mathcal{Z}_\theta, \partial \mathcal{B}) = \mathrm{d}_{\mathrm{h}}(\Phi_\theta^{-1}(\partial \mathcal{X}_\theta), \partial \mathcal{B}) \tag{64}$$

$$\leq \mathrm{L}(\Phi_\theta^{-1}, \mathcal{X}_\theta \cup \Phi_\theta(\mathcal{B}))\, \mathrm{d}_{\mathrm{H}}(\partial \mathcal{X}_\theta, \Phi_\theta(\partial \mathcal{B})) \quad \text{(by Lemma 2)} \tag{65}$$

$$\leq \mathrm{L}(\Phi_\theta^{-1}, \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\mathrm{pre}}, \epsilon_{\mathrm{inn}}\}))(\mathrm{d}_{\mathrm{H}}(\partial \mathcal{X}_\theta, \partial \mathcal{K}_\theta) + \mathrm{d}_{\mathrm{H}}(\partial \mathcal{K}_\theta, \Phi_\theta(\partial \mathcal{B}))) \tag{66}$$

$$\leq \mathrm{L}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\mathrm{pre}} + \epsilon_{\mathrm{inn}}), \tag{67}$$

where $\mathcal{Y}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\mathrm{pre}}, \epsilon_{\mathrm{inn}}\})$.

Under the prediction error ϵ_{pre} and approximation error ϵ_{inn}, the sampling region $\mathcal{Z}_\theta = \Phi^{-1}(\mathcal{X}_\theta)$ for distortion estimation is as:

$$\mathcal{Z}_\theta \subseteq \mathcal{B} + \mathcal{B}(0, \mathrm{d}_{\mathrm{h}}(\partial \mathcal{Z}_\theta, \partial \mathcal{B})) \quad \text{(by Lemma 2)} \tag{68}$$

$$\subseteq \mathcal{B}(0, 1 + \mathrm{L}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\mathrm{pre}} + \epsilon_{\mathrm{inn}})). \tag{69}$$

Therefore, theoretically, we can sample and evaluate the distortion over such a ball instead of the original irregular regions $\mathcal{Z}_\theta$. However, we do not know the prediction error nor the approximation error in advance to set such a radius of the ball as $R = 1 + \mathrm{L}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\mathrm{pre}} + \epsilon_{\mathrm{inn}})$. In practice, we set $R \geq 1$ as a hyperparameter.

On the other hand, for the implementation of distortion estimation, we use the expectation instead of the $\sup$ and $\inf$ operators as $\widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta) \approx \mathbb{E}_{z \in \mathcal{Z}_\theta}[\sum_{l=1}^{m} \log \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l)) - \log \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))]$. In other words, the overall approximation estimates the average local distortion at each point over each layer. Such an average-case distortion simplifies the original worst-case formulation and stabilizes the INN training. Such an average approximation has also been demonstrated efficient in the related Lipschitz regularization schemes (Virmaux & Scaman, 2018; Behrmann et al., 2019; 2021).

We select the singular values to represent distortion because of the closed-form expression under some INN structures. When using a more general INN structure (e.g., i-ResFlow (Behrmann et al., 2019)), we may leverage the commonly used finite difference estimation to evaluate and limit the distortion (Schmidt et al., 2019; Behrmann et al., 2021; Liu et al., 2022).

F. Proof for Theorem 1

F.1. Feasibility

Since Φ_θ is a valid mapping, we have $\Phi_\theta(0) \in \mathcal{K}_\theta \leftrightarrow 0 \in \Phi_\theta^{-1}(\mathcal{K}_\theta)$. Therefore, from 0 to 1, at least one feasible point exists (e.g., 0). The bisection process is to find the nearest feasible point to the infeasible prediction $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta) \notin \Phi_\theta^{-1}(\mathcal{K}_\theta)$ in the homeomorphic space by solving:

$$\alpha^* = \sup_{\alpha \in [0,1]} \{\Phi_\theta(\alpha \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta\}. \tag{70}$$

The process can be view as "projecting" the infeasible point $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$ to the mapped constraint boundary $\Phi_\theta^{-1}(\partial\mathcal{K}_\theta)$ as $\hat{z}_\theta = \alpha^* \cdot \tilde{z}_\theta \in \Phi_\theta^{-1}(\partial\mathcal{K}_\theta)$.

(a) The illustration of approximation error in homeomorphic space. (b) The illustration of the lower bound such that $\alpha_l^* \mathcal{B} \subseteq \Phi_\theta^{-1}(\mathcal{K}_\theta)$.

Consider point over constraint boundary $z \in \Phi_\theta^{-1}(\partial\mathcal{K}_\theta)$, we have:

$$0 \leq r + \inf_{x \in \partial\mathcal{K}_\theta}\{\|\Phi_\theta^{-1}(x)\| - r\} \leq \|z\| \leq r + \sup_{x \in \partial\mathcal{K}_\theta}\{\|\Phi_\theta^{-1}(x)\| - r\}, \quad \forall z \in \Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \tag{71}$$

where r is the radius for the ball (we use $r = 1$ for implementation). For simplicity, we denote $\delta_{\text{inn}}^- = \inf_{x \in \partial\mathcal{K}_\theta}\{\|\Phi_\theta^{-1}(x)\| - r\}$, $\delta_{\text{inn}}^+ = \sup_{x \in \partial\mathcal{K}_\theta}\{\|\Phi_\theta^{-1}(x)\| - r\}$ and $\delta_{\text{inn}} = \text{d}_\text{h}(\Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \partial\mathcal{B})$. It can be viewed as evaluating the approximation error of INN between $\partial\mathcal{B}$ and $\Phi_\theta^{-1}(\partial\mathcal{K}_\theta)$.

According to Lemma 2, we have $\text{d}_\text{h}(\Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \partial\mathcal{B}) \geq \sup_{x \in \partial\mathcal{K}_\theta}\{|\|\Phi_\theta^{-1}(x)\|_2 - r|\}$, which leads to $\delta_{\text{inn}} = \text{d}_\text{h}(\Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \partial\mathcal{B}) \geq \max\{|\delta_{\text{inn}}^+|, |\delta_{\text{inn}}^-|\}$.

Therefore, we have the bound for the optimal α^*:

$$0 \leq r + \delta_{\text{inn}}^- \leq \|\alpha^*\tilde{z}_\theta\| \leq r + \delta_{\text{inn}}^+ \quad \rightarrow \quad 0 \leq \frac{r + \delta_{\text{inn}}^-}{\|\tilde{z}_\theta\|} \leq \alpha^* \leq \frac{r + \delta_{\text{inn}}^+}{\|\tilde{z}_\theta\|}. \tag{72}$$

By taking the lower bound of $\alpha_l^* = \frac{r + \delta_{\text{inn}}^-}{\|\tilde{z}_\theta\|}$, we can next analyze the worst-case optimality loss after ensuring the feasibility given infeasible predictions.

F.2. Optimality

To analyze the optimality loss of returned solution $\hat{x}_\theta^k$ by Algorithm 1, we decompose the loss by following terms:

$$\|\hat{x}_\theta^k - x_\theta^*\| \leq \underbrace{\|\hat{x}_\theta^k - \hat{x}_\theta\|}_{\text{bisection error}} + \underbrace{\|\hat{x}_\theta - \tilde{x}_\theta\|}_{\text{projection error}} + \underbrace{\|\tilde{x}_\theta - x_\theta^*\|}_{\text{prediction error}}, \tag{73}$$

where $\hat{x}_\theta$ is the feasible solution as $\hat{x}_\theta = \Phi_\theta(\hat{z}_\theta) = \Phi_\theta(\alpha^* \cdot \tilde{z}_\theta) = \Phi_\theta(\alpha^* \cdot \Phi_\theta^{-1}(\tilde{x}_\theta))$, $\tilde{x}_\theta \in \mathcal{X}_\theta$ is the predicted infeasible solution, and x_θ^* is the optimal solution for problem in (1).

- **Prediction error**

  By definition of $\epsilon_{\text{pre}} = \sup_{\theta \in \Theta}\{\|F(\theta) - x_\theta^*\|\}$, where $\tilde{x}_\theta = F(\theta)$, the prediction error is then bounded as:

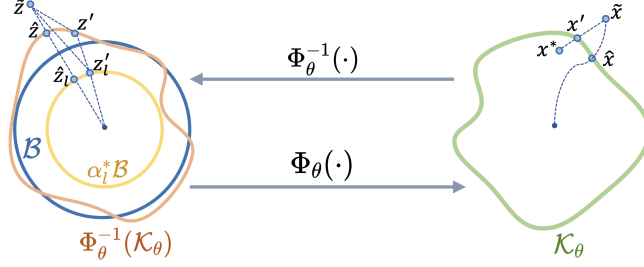$$\|\tilde{x}_\theta - x_\theta^*\| \leq \epsilon_{\text{pre}}. \tag{74}$$

Figure 7: Illustration of Proof for projection error.

- **Projection error**

  The second part is the error by "projecting" the infeasible solution back to the constraint set after perfecting solving $\alpha^* = \sup_{\alpha \in [0,1]} \{ \Phi_\theta(\alpha^* \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta \}$. For predicted infeasible solution $\tilde{x}_\theta \in \mathcal{X}_\theta$ and projected feasible solution $\hat{x}_\theta \in \mathcal{K}_\theta \subseteq \mathcal{X}_\theta$, we have the projection distance as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| = \|\Phi_\theta(\hat{z}_\theta) - \Phi_\theta(\tilde{z}_\theta)\| \tag{75}$$

$$\leq \underbrace{L(\Phi_\theta, \mathcal{Z}_\theta)\|\hat{z}_\theta - \tilde{z}_\theta\| = L(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha^* \tilde{z}_\theta - \tilde{z}_\theta\|}_{\text{where } \hat{x}_\theta, \tilde{x}_\theta \in \mathcal{X}_\theta \text{ and } \hat{z}_\theta, \tilde{z}_\theta \in \mathcal{Z}_\theta} \tag{76}$$

$$\leq \underbrace{L(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^* \tilde{z}_\theta - \tilde{z}_\theta\|}_{\text{Lower bound: } \alpha_l^* = (r + \delta_{\text{inn}}^-)/\|\tilde{z}_\theta\|} \tag{77}$$

$$\leq \underbrace{L(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z_\theta'\|} z_\theta' - \tilde{z}_\theta\|}_{\text{Lemma 3: } \|\hat{z}_l - \tilde{z}\| \leq \|z_l' - \tilde{z}\|} \quad (\text{Let } z_\theta' = \Phi_\theta^{-1}(x_\theta'), \text{ where } x_\theta' = \lambda x_\theta^* + (1 - \lambda)\tilde{x}_\theta \in \partial\mathcal{K}_\theta.) \tag{78}$$

$$\leq L(\Phi_\theta, \mathcal{Z}_\theta)(\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z_\theta'\|} z_\theta' - z_\theta'\| + \|z_\theta' - \tilde{z}_\theta\|) \tag{79}$$

$$\leq L(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z_\theta'\|} z_\theta' - z_\theta'\| + \underbrace{L(\Phi_\theta, \mathcal{Z}_\theta) L(\Phi_\theta^{-1}, \mathcal{X}_\theta)(\|x_\theta' - \tilde{x}_\theta\|)}_{\text{Lemma 1: } D(\Phi_\theta, \mathcal{Z}_\theta) = L(\Phi_\theta, \mathcal{Z}_\theta) L(\Phi_\theta^{-1}, \mathcal{X}_\theta)} \tag{80}$$

$$\leq L(\Phi_\theta, \mathcal{Z}_\theta)|\alpha_l^*\|\tilde{z}_\theta\| - \|z_\theta'\|| \cdot \|\frac{z_\theta'}{\|z_\theta'\|}\| + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{81}$$

$$\leq L(\Phi_\theta, \mathcal{Z}_\theta)|r + \delta_{\text{inn}}^- - \|z_\theta'\|| + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{82}$$

$$\leq \underbrace{L(\Phi_\theta, \mathcal{Z}_\theta)(|r - \|z_\theta'\|| + |\delta_{\text{inn}}^-|)}_{|r - \|z_\theta'\|| \leq \delta_{\text{inn}}, \ |\delta_{\text{inn}}^-| \leq \delta_{\text{inn}}} + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{83}$$

$$\leq 2 L(\Phi_\theta, \mathcal{Z}_\theta)\delta_{\text{inn}} + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{84}$$

$$\leq \underbrace{2 L(\Phi_\theta, \mathcal{Z}_\theta) L(\Phi_\theta^{-1}, \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\text{inn}}}_{\text{Lemma 2: } \delta_{\text{inn}} \leq L(\Phi_\theta^{-1}, \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\text{inn}}} + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{85}$$

$$\leq \underbrace{2 L(\Phi_\theta, \mathcal{Z}_\theta) L(\Phi_\theta^{-1}, \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\text{inn}}}_{\text{Lemma 2: } \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pre}}, \epsilon_{\text{inn}}\})} + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{86}$$

$$\leq \underbrace{2 L(\Phi_\theta, \mathcal{Z}_\theta) L(\Phi_\theta^{-1}, \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pre}}, \epsilon_{\text{inn}}\}))\epsilon_{\text{inn}}}_{\mathcal{X}_\theta = \Phi_\theta(\mathcal{Z}_\theta) = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\text{pre}}) \subseteq \mathcal{Y}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pre}}, \epsilon_{\text{inn}}\})} + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{87}$$

$$\leq \underbrace{D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pre}})}_{\text{Lemma 1: } D(\Phi_\theta^{-1}, \mathcal{X}_\theta) \leq D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)} . \tag{88}$$

Further, if we select the x_θ' as the point by standard projection, i.e., $x_\theta' = \hat{x}_\theta^* \in \text{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \arg\min_{y \in \mathcal{K}_\theta}\{\|\tilde{x}_\theta - y\|\}$ and denoted the projection distance as $\epsilon_{\text{pro}} = \|\hat{x}_\theta^* - \tilde{x}_\theta\| \leq \epsilon_{\text{pre}}$, then we have a bound for $\|\hat{x}_\theta - \tilde{x}_\theta\|$ related to the standard projection distance: $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pro}})$.

- **Bisection error**

The last part is the error coming from solving $\alpha^* = \sup_{\alpha \in [0,1]} \{\Phi_\theta(\alpha^* \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta\}$ using bisection in Algo. 1. Since we set the maximum bisection steps as k, the error is bounded as:

$$\|\hat{x}_\theta^k - \hat{x}_\theta\| \leq L(\Phi_\theta, \mathcal{Z}_\theta)\|\hat{z}_\theta^k - \hat{z}_\theta\| \tag{89}$$

$$= L(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha^k \tilde{z}_\theta - \alpha^* \tilde{z}_\theta\| \tag{90}$$

$$= L(\Phi_\theta, \mathcal{Z}_\theta)|\alpha^k - \alpha^*|\|\tilde{z}_\theta - 0\| \tag{91}$$

$$\leq L(\Phi_\theta, \mathcal{Z}_\theta)\frac{1}{2^k}\|\tilde{z}_\theta - 0\| \tag{92}$$

$$\leq L(\Phi_\theta, \mathcal{Z}_\theta) L(\Phi_\theta^{-1}, \mathcal{X}_\theta)\frac{1}{2^k}\|\tilde{x}_\theta - \Phi_\theta(0)\| \tag{93}$$

$$\leq D(\Phi_\theta^{-1}, \mathcal{X}_\theta)\frac{1}{2^k}(\|\tilde{x}_\theta - x_\theta^*\| + \|x_\theta^* - \Phi_\theta(0)\|) \tag{94}$$

$$\leq D(\Phi_\theta^{-1}, \mathcal{X}_\theta)\frac{1}{2^k}(\epsilon_{\text{pre}} + \text{diam}(\mathcal{K}_\theta)) \tag{95}$$

$$\leq D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)\frac{1}{2^k}(\epsilon_{\text{pre}} + \text{diam}(\mathcal{K}_\theta)). \tag{96}$$

Combing the prediction error, projection error, and bisection error, we have the results:

$$\|\hat{x}_\theta^k - x_\theta^*\| \leq \epsilon_{\text{pre}} + D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pre}} + \epsilon_{\text{bis}}^k), \tag{97}$$

where $\mathcal{Y}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pre}}, \epsilon_{\text{inn}}\})$ and $\epsilon_{\text{bis}}^k = 2^{-k}(\text{diam}(\mathcal{K}_\theta) + \epsilon_{\text{pre}})$.

F.2.1. OPTIMALITY UNDER SPECIAL CASES

From the analysis of optimality loss, we find the prediction error is dominated by the given predictor, the bisection error converges to zero with increasing bisection steps, and the main gap comes from the projection error. To minimize the projection error, we formulate the MDH mapping problem to minimize the distortion and the approximation error. To better understand the projection error, we discuss its upper bound under several special cases, as shown in Table 3.

Table 3: Projection error under different cases.

Setting	Projection error: $\|\hat{x}_\theta - \tilde{x}_\theta\|$
Valid INN mapping $\Phi_\theta(0) \in \mathcal{K}_\theta$	$D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pre}})$
Valid outer approximation $\Phi_\theta(0) \in \mathcal{K}_\theta$ and $\Phi_\theta(\mathcal{B}) \supseteq \mathcal{K}_\theta$	$D(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\text{inn}} + \epsilon_{\text{pre}})$
Inner approximation $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$	$D(\Phi_\theta^{-1}, \mathcal{X}_\theta)(\epsilon_{\text{inn}} + \epsilon_{\text{pre}})$
Lower bound by standard projection $\hat{x}_\theta^* \in \text{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta)$	$\epsilon_{\text{pro}} = \|\hat{x}_\theta^* - \tilde{x}_\theta\| \leq \epsilon_{\text{pre}}$
Feasible homeomorphic mapping $\Phi_\theta(\mathcal{B}) = \mathcal{K}_\theta$	$D(\Phi_\theta^{-1}, \mathcal{X}_\theta)\epsilon_{\text{pro}}$
Feasible isometric homeomorphic mapping $\Phi_\theta(\mathcal{B}) = \mathcal{K}_\theta$ and $D(\Phi_\theta^{-1}, \mathcal{X}_\theta) = 1$	ϵ_{pro}

- **Case 1: inner approximation**

If we have an approximated MDH such that $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$, which constructs an inner approximation of the constraint set and it is also valid $\Phi_\theta(0) \in \mathcal{K}_\theta$.

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq D(\Phi_\theta^{-1}, \mathcal{X}_\theta)(\epsilon_{\text{inn}} + \epsilon_{\text{pre}}). \tag{98}$$

Proof: Since $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$, after homeomorphic mapping $\mathcal{B} \subseteq \Phi_\theta^{-1}(\mathcal{K}_\theta)$, we have $0 \leq \delta_{\text{inn}}^- \leq \delta_{\text{inn}}^+ = \delta_{\text{inn}}$. Following (82), we have:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \underbrace{L(\Phi_\theta, \mathcal{Z}_\theta)|r + \delta_{\text{inn}}^- - \|z_\theta'\||}_{\delta_{\text{inn}}^+ \geq \delta_{\text{inn}}^- \geq 0} + D(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{99}$$

22

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)(|\delta_{\mathrm{inn}}^+ - \delta_{\mathrm{inn}}^-|) + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{100}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\delta_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{101}$$

$$\leq \underbrace{\mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\,\mathrm{L}(\Phi_\theta^{-1}, \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\mathrm{inn}}}_{\text{Lemma 2: } \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B})) = \mathcal{X}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\mathrm{pre}})} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{102}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\,\mathrm{L}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\epsilon_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{103}$$

$$= \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pre}}). \tag{104}$$

Similarly, if we select the x'_θ as the point by standard projection, i.e., $x'_\theta = \hat{x}_\theta^* \in \mathrm{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \arg\min_{y \in \mathcal{K}_\theta}\{\|\tilde{x}_\theta - y\|\}$, then we have a bound related to the standard projection distance: $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pro}})$.

- **Case 2: valid outer approximation**

Similarly, if we have an approximated MDH such that $\Phi_\theta(0) \in \mathcal{K}_\theta$ and $\Phi_\theta(\mathcal{B}) \supseteq \mathcal{K}_\theta$, which constructs an otter approximation of the constraint set.

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pre}}). \tag{105}$$

Proof: Since $\Phi_\theta(\mathcal{B}) \supseteq \mathcal{K}_\theta$, after homeomorphic mapping $\mathcal{B} \supseteq \Phi_\theta^{-1}(\mathcal{K}_\theta)$, we have $\delta_{\mathrm{inn}}^- \leq \delta_{\mathrm{inn}}^+ \leq 0$ and $\delta_{\mathrm{inn}} = |\delta_{\mathrm{inn}}^-|$ Following (82), we have:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \underbrace{\mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)(|r + \delta_{\mathrm{inn}}^- - \|z_\theta^b\||)}_{\delta_{\mathrm{inn}}^- \leq \delta_{\mathrm{inn}}^+ \leq 0} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{106}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)(|\delta_{\mathrm{inn}}^+ - \delta_{\mathrm{inn}}^-|) + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{107}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\delta_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{108}$$

$$\leq \underbrace{\mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\,\mathrm{L}(\Phi_\theta^{-1}, \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\mathrm{inn}}}_{\text{Lemma 2: } \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B})) \subset \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\mathrm{pre}}, \epsilon_{\mathrm{inn}}\})} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{109}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\,\mathrm{L}(\Phi_\theta^{-1}, \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\mathrm{pre}}, \epsilon_{\mathrm{inn}}\}))\epsilon_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{110}$$

$$\leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pre}}). \tag{111}$$

Similarly, if we select the x'_θ as the point by standard projection, i.e., $x'_\theta = \hat{x}_\theta^* \in \mathrm{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \arg\min_{y \in \mathcal{K}_\theta}\{\|\tilde{x}_\theta - y\|\}$, then we have a bound related to the standard projection distance: $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pro}})$.

- **Case 3: feasible homeomorphic mapping**

Under the feasible mapping $\Phi_\theta \in \mathcal{H}^n(\mathcal{K}_\theta, \mathcal{B})$, given an infeasible prediction $\tilde{x}_\theta$, the homeomorphic bisection algorithm is equivalent with following projection operator: $\hat{x}_\theta = \mathrm{HP}_{\mathcal{K}_\theta}^{\Phi_\theta}(\tilde{x}_\theta) = \Phi_\theta(\mathrm{Proj}_{\mathcal{B}}(\Phi_\theta^{-1}(\tilde{x}_\theta)))$. Let the optimal projection solution as $\hat{x}_\theta^* \in \mathrm{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \arg\min_{y \in \mathcal{K}_\theta}\{\|\tilde{x}_\theta - y\|\} \subset \partial\mathcal{K}_\theta$ and the projection distance as $\epsilon_{\mathrm{pro}} = \|\hat{x}_\theta^* - \tilde{x}_\theta\|$.

The homeomorphic projection error is bounded as:

$$\epsilon_{\mathrm{pro}} \leq \|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\epsilon_{\mathrm{pro}} \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\epsilon_{\mathrm{pre}}. \tag{112}$$

Proof: By definition, the homeomorphic projection is bounded as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| = \|\Phi_\theta(\mathrm{Proj}_{\mathcal{B}}(\Phi_\theta^{-1}(\tilde{x}_\theta))) - \Phi_\theta(\psi_\theta^{-1}(\tilde{x}_\theta))\| \tag{113}$$

$$\leq \underbrace{\mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\|\mathrm{Proj}_{\mathcal{B}}(\Phi_\theta^{-1}(\tilde{x}_\theta)) - \psi_\theta^{-1}(\tilde{x}_\theta)\|}_{\mathcal{Z}_\theta = \psi_\theta^{-1}(\mathcal{X}_\theta)} \tag{114}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\underbrace{\|\Phi_\theta^{-1}(\hat{x}_\theta^*) - \psi_\theta^{-1}(\tilde{x}_\theta)\|}_{\text{property of projection}} \tag{115}$$

$$\leq \mathrm{L}(\Phi_\theta, \mathcal{Z}_\theta)\,\mathrm{L}(\psi_\theta^{-1}, \mathcal{X}_\theta)\|\hat{x}_\theta^* - \tilde{x}_\theta\| = \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\epsilon_{\mathrm{pro}} \tag{116}$$

$$\leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\|x_\theta^* - \tilde{x}_\theta\| = \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\epsilon_{\mathrm{pre}}. \tag{117}$$

Therefore, the homeomorphic projection error has a lower bounded as the standard projection error and an upper bounded by the standard projection error multiplied by the distortion of homeomorphic mapping. Further, if the feasible mapping is isometric, the homeomorphic projection error reaches the lower bound.

Definition F.1. A mapping $\psi : \mathbb{R}^n \to \mathbb{R}^n$ is isometric if it preserves the distance as: $\|\psi(x) - \psi(y)\| = \kappa\|x - y\|, \forall x, y \in \mathcal{Z}$, where $\kappa > 0$.

In other words, if the homeomorphic mapping Φ_θ is isometric over $\mathcal{Z}_\theta$, the distortion $\mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta) = 1$. The homeomorphic projection distance is exactly the same as the standard projection distance.

F.3. Run-time complexity

The run-time complexity involves the m-layer INN forward calculation and the inequality constraint calculation.

- INN forward calculation: We adopt three kinds of INN layers in our design: the auto-regressive layer, the actnorm layer, and the invertible linear layer. As mentioned in Appendix C. Those layers have the run-time complexity as $\mathcal{O}(n^2)$. The inverse of those layers has exactly the same run-time complexity.

- Constraint calculation: After the INN forward calculation, we have a candidate solution $\hat{x}$ and calculate $\mathbf{g}(\hat{x}, \theta) = [g_i(\hat{x}, \theta), ..., g_{n_{\text{ineq}}}(\hat{x}, \theta)]$ to find if it is a feasible solution. We denote the complexity for calculating $\mathbf{g}(\hat{x}, \theta)$ as G.

Therefore, to execute k-step binary search, we have the total run-time complexity as $\mathcal{O}(k(mn^2 + G))$

G. Proof for Theorem 2

Since $\mathcal{D}_1$ is r_c-covering dataset, i.e., for any $\theta \in \Theta$, there exist $\theta_0 \in \mathcal{D}_1$ such that $\|\theta - \theta^0\| \leq r_c$. For such a pair of (θ, θ^0), first, we bound the variation of the constraint boundary:

$$\mathrm{d_H}(\partial\mathcal{K}_\theta, \partial\mathcal{K}_{\theta^0}) \leq \|\theta - \theta^0\|C_0 \leq r_c C_0, \tag{118}$$

where $C_0 = \sup_{\theta_1, \theta_2 \in \Theta, \theta_1 \neq \theta_2}\{\mathrm{d_H}(\partial\mathcal{K}_{\theta_1}, \partial\mathcal{K}_{\theta_2})/\|\theta_1 - \theta_2\|\}$ indicates the "Lipschitz" of the constraint set over input parameters. Then $r_c C_0$ provides an upper bound for the constraint boundary variation according to the input parameter variation under r_c- covering dataset $\mathcal{D}_1$. Further, by Lemma 2, we have $\partial\mathcal{K}_\theta \subseteq \partial\mathcal{K}_{\theta^0} + \mathcal{B}(0, r_c C_0)$.
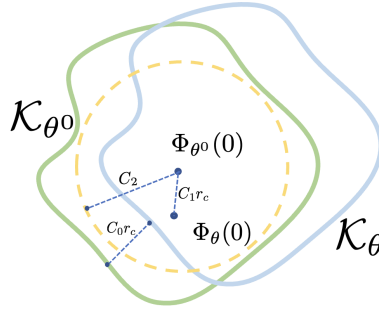

Figure 8: Illustration of $C_0 r_c$, $C_1 r_c$, and C_2.

Since the INN is valid for the interior of the constraint set under the training dataset, i.e., for any $\theta^0 \in \mathcal{D}_1$, $\Phi_{\theta^0}(0) \in \mathcal{K}_{\theta^0}^\circ$, there exists $r > 0$ such that $\mathcal{B}(\Phi_{\theta^0}(0), r) \subseteq \mathcal{K}_{\theta^0}$, i.e., the inner approximation ball for constraint set centered in $\Phi_{\theta^0}(0)$, we then denote the largest radius of the inner ball as $C_2 = \arg\sup_{r>0}\{\mathcal{B}(\Phi_{\theta^0}(0), r) \subseteq \mathcal{K}_{\theta^0}, \theta^0 \in \mathcal{D}_1\}$. In other words, we have to move the point $\Phi_{\theta^0}(0)$ or the boundary $\partial\mathcal{K}_{\theta^0}$ with a distance of at least C_2 such that the point would be outside the boundary.

If the condition $C_1 r_c + C_0 r_c \leq C_2$ holds, the distance between boundaries $\mathrm{d_H}(\partial\mathcal{K}_\theta, \partial\mathcal{K}_{\theta^0}) \leq C_0 r_c \leq C_2$ holds, then the point is inside the boundary as $\Phi_{\theta^0}(0) \in \mathcal{K}_\theta$.

Given $y \in \partial\mathcal{K}_\theta$, let $x_y = \arg\inf_{x \in \partial\mathcal{K}_{\theta^0}}\{\|x - y\|\}$ and $z_{x_y} = \lambda\Phi_{\theta^0}(0) + (1 - \lambda)x_y \in \partial\mathcal{B}(\Phi_{\theta^0}(0), C_2)$, then

$$\|\Phi_{\theta^0}(0) - y\| \geq \|\Phi_{\theta^0}(0) - x_y\| - \|x_y - y\| \tag{119}$$

$$\geq \|\Phi_{\theta^0}(0) - z_{x_y}\| - \inf_{x \in \partial\mathcal{K}_{\theta^0}}\{\|x - y\|\} \tag{120}$$

$$\geq C_2 - \sup_{y \in \partial \mathcal{K}_\theta} \inf_{x \in \partial \mathcal{K}_{\theta^0}} \{\|x - y\|\} \tag{121}$$

$$\geq C_2 - r_c C_0. \tag{122}$$

Therefore $\inf_{y \in \partial \mathcal{K}_\theta} \{\|\Phi_{\theta^0}(0) - y\|\} \geq C_2 - r_c C_0$, i.e., $\mathcal{B}(\Phi_{\theta^0}(0), C_2 - r_c C_0) \subseteq \mathcal{K}_\theta$.

Second, we bound the variation of the mapped center for a pair of (θ, θ^0) as:

$$\|\Phi(0, \theta) - \Phi(0, \theta^0)\| \leq \|\theta - \theta^0\| \, \mathrm{L}(\Phi(0, \cdot), \Theta) \leq r_c C_1, \tag{123}$$

where $C_1 = \mathrm{L}(\Phi(0, \cdot), \Theta)$ indicates the Lipschitz of the trained INN over in the input parameters. $r_c C_1$ provides an upper bound for the variation of the mapped center according to the input parameter variation under r_c- covering dataset $\mathcal{D}_1$ such that $\Phi_\theta(0) \in \mathcal{B}(\Phi_{\theta^0}(0), r_c C_1)$.

If the condition $C_1 r_c + C_0 r_c \leq C_2$ holds, we have:

$$\Phi_\theta(0) \in \mathcal{B}(\Phi_{\theta^0}(0), r_c C_1) \subseteq \mathcal{B}(\Phi_{\theta^0}(0), C_2 - r_c C_0) \subseteq \mathcal{K}_\theta. \tag{124}$$

Therefore, if $C_1 r_c + C_0 r_c \leq C_2$ holds, $\forall \theta \in \Theta$, we have $\Phi_\theta(0) \in \mathcal{K}_\theta$.

H. Experiment settings

In this appendix, we will specify the formulation of constrained optimization problems in Appendix H.1, the detailed structure of NN predictors and training parameters in Appendix H.2, and the structure of INN mapping and the training parameters of INN approximated MDH mapping are in Appendix H.3.

H.1. Formulation of constrained optimization problem

We test the HP framework for three constraint optimization problems: QP, convex QCQP, SDP, and AC-OPF. The mathematical formulation is presented below:

$$\textbf{QP}: \min_{x \in \mathbb{R}^n} \frac{1}{2} x^\mathsf{T} Q x + p^\mathsf{T} x \tag{125}$$
$$\text{s.t. } Gx \leq h, \ Ax = \theta,$$

where $Q \in \mathbb{S}_{++}^n, p \in \mathbb{R}^n, A \in \mathbb{R}^{n_{\mathrm{eq}} \times n}, G \in \mathbb{R}^{n_{\mathrm{ineq}} \times n}, h \in \mathbb{R}^{n_{\mathrm{ineq}}}$, and $\theta \in \mathbb{R}^{n_{\mathrm{eq}}}$.

$$\textbf{Convex QCQP}: \min_{x \in \mathbb{R}^n} \frac{1}{2} x^\mathsf{T} Q x + p^\mathsf{T} x \tag{126}$$
$$\text{s.t. } x^\mathsf{T} H_i x + g_i^\mathsf{T} x \leq h_i, \ Ax = \theta,$$

where $H_i \in \mathbb{S}_{++}^n, g_i \in \mathbb{R}^n$, and $h_i \in \mathbb{R}$ for $i = 1, ..., n_{\mathrm{ineq}}$.

$$\textbf{SDP}: \min_{X \in \mathbb{R}^{n \times n}} \mathrm{tr}(CX) \tag{127}$$
$$\text{s.t. } X \succeq 0, \ \mathrm{tr}(A_i X) = \theta_i,$$

where $C \in \mathbb{R}^{n \times n}, A_i \in \mathbb{R}^{n \times n}$, and $\theta_i \in \mathbb{R}$ for $i = 1, ..., n_{\mathrm{eq}}$.

A specific introduction of AC-OPF (non-convex QCQP) is as follows.

$$\textbf{AC-OPF}: \min_{p_g \in \mathbb{R}^n, q_g \in \mathbb{R}^n, v \in \mathbb{C}^n} p_g^\mathsf{T} Q p_g + b^\mathsf{T} p_g \tag{128}$$
$$\text{s.t. } p_g^{\min} \leq p_g \leq p_g^{\max}, \tag{129}$$
$$q_g^{\min} \leq q_g \leq q_g^{\max}, \tag{130}$$
$$v^{\min} \leq |v| \leq v^{\max}, \tag{131}$$
$$(p_g - p_d) + (q_g - q_d)\, i = \mathrm{diag}(v) \bar{W} \bar{v}. \tag{132}$$

For a power network with n nodes, $p_d \in \mathbb{R}^n, q_d \in \mathbb{R}^n$ represent the real power and reactive power demand. $p_g \in \mathbb{R}^n, q_g \in \mathbb{R}^n$ represent the real power and reactive power generation. $v \in \mathbb{C}^n$ denotes the voltage (both real and imaginary parts). $W \in \mathbb{C}^{n \times n}$ is the nodal admittance matrix of the power network, which represents its topology. Given different power demands, we solve the optimal power generation satisfying the flow-balance constraints and minimizes the generation cost.

H.2. Structure and training for NN predictor

We adopt the fully connected NN to predict the optimal solution for constrained optimization problems given the input parameters. The parameters are in Table 4.

H.3. Structure and training for INN mapping

Motivated by the GLOW (Kingma & Dhariwal, 2018), we build INN containing three basic layers as shown in Table 5. To train the MDH mapping for different examples, we select the parameters shown in Table 5.

Table 4: Structure of NN predictor in experiments

Parameter	Value
NN structure	
dimension of input layer	d
dimension of output layer	n
dimension of hidden layer	$\lfloor (d+n)/2 \rfloor$
activation function	$\mathrm{ReLU}(\cdot)$
number of layer	3
last layer	$\mathrm{Sigmoid}(\cdot)$
NN training parameters	
number of training samples (θ, x_θ^*)	20,000
number of testing samples	1,024
number of iteration	5,000
optimizer	Adam
learning rate	0.001
batch size	512
the coefficient for objective function	0.1
the coefficient for inequality penalty	10
the coefficient for equality penalty	10

Table 5: Structure of INN mapping in experiments

Parameter	Value
INN structure	
dimension of input layer	$n+d$
dimension of output layer	n
basic INN block	$\left\{\begin{array}{l}\text{Actnorm, Inv. Conv.}\\ \text{Actnorm, Autoregressive}\end{array}\right\}$
number of INN block	3
the last layer	$\mathrm{Sigmoid}(\cdot)$
INN training parameters	
$\mathcal{B}$	$2/\infty$-norm
scale ratio R	1
number of QMC samples $z \in \mathcal{B}$	1,024
number of Uniform samples $\theta \in \Theta$	10,000
number of iteration	10,000
optimizer	Adam
learning rate	0.0001
batch size	512
the coefficient for penalty	50
the coefficient for distortion	5

H.4. More examples for MDH mapping approximated constraint set

We train two MDH mappings to approximate the constraint set $\mathcal{K}_\theta$ from a 2-norm ball and a ∞-norm ball, respectively. After that, we visualize the approximated constraint set by plotting $\Phi(\mathcal{B}|\theta_{\text{test}})$ under new input parameters. The visualization is shown in Fig. 9 and 10, respectively.
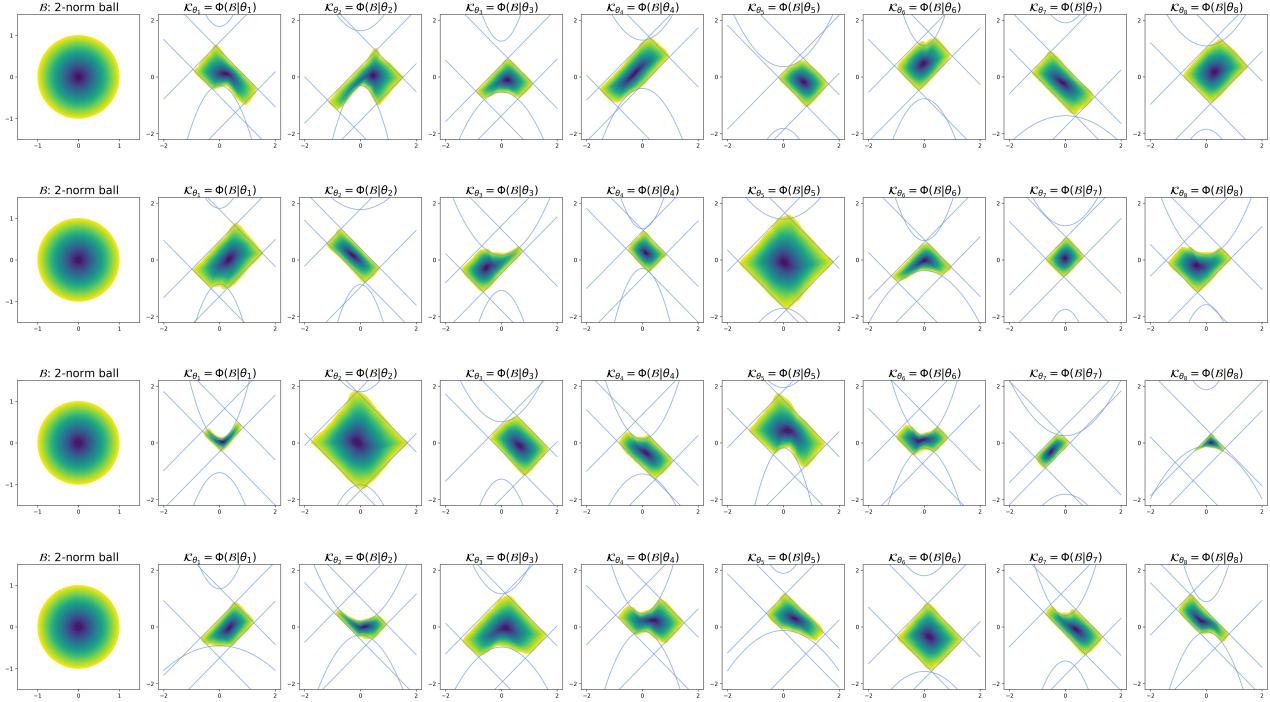
Figure 9: Testing performance for 2-norm ball to constraint sets under different testing input parameters.
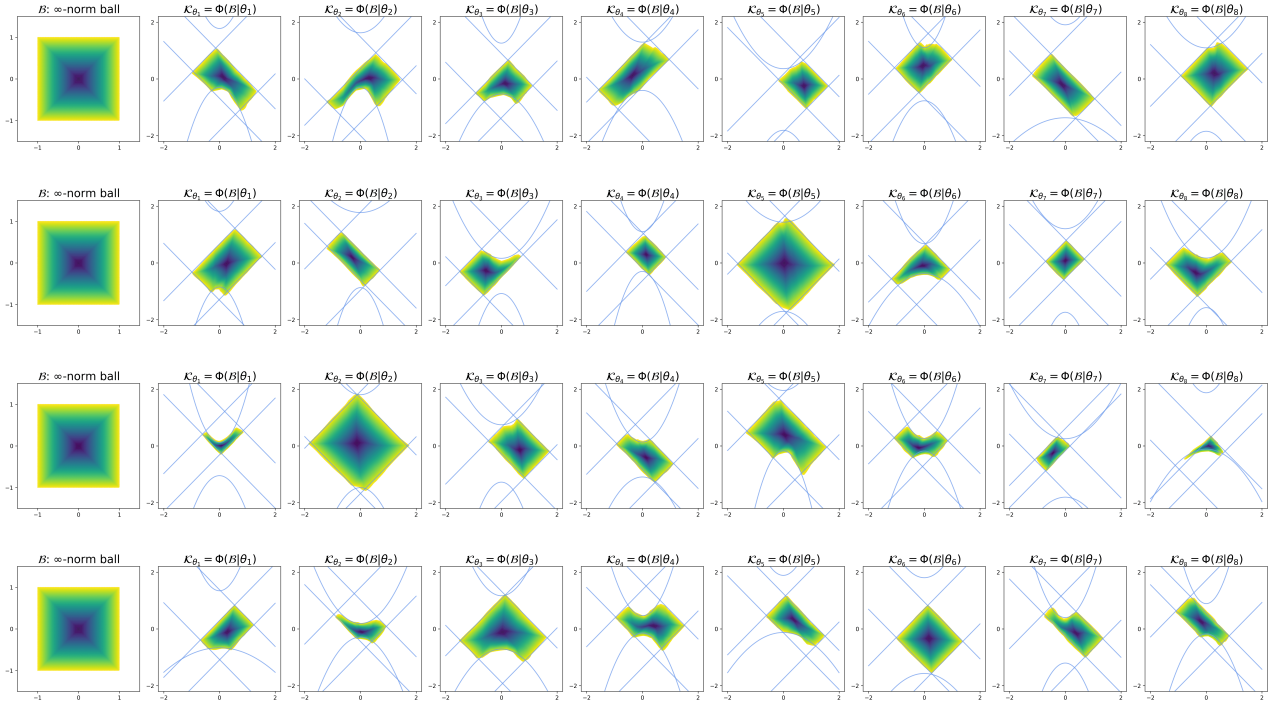

Figure 10: Testing performance for ∞-norm ball to constraint sets under different testing input parameters.