Journal of Machine Learning Research 25 (2024) 1-55 Submitted 11/23; Revised 9/24; Published 9/24

Homeomorphic Projection to Ensure Neural-Network Solution Feasibility for Constrained Optimization

Enming Liang ELIANG4-C@MY.CITYU.EDU.HK
Department of Data Science
City University of Hong Kong
Hong Kong, China

Minghua Chen* MINGHUA.CHEN@CITYU.EDU.HK
Department of Data Science
City University of Hong Kong
Hong Kong, China

Steven H. Low SLOW@CALTECH.EDU
Departments of EE and CMS
California Institute of Technology
California, USA

Editor: Animashree Anandkumar

Abstract

There has been growing interest in employing neural networks (NNs) to directly solve constrained optimization problems with low run-time complexity. However, it is non-trivial to ensure NN solutions strictly satisfy problem constraints due to inherent NN prediction errors. Existing feasibility-ensuring methods are either 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 a bi-Lipschitz invertible NN (INN), and then (ii) perform a simple bisection operation concerning the unit ball such 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 bounded 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.

Keywords: constrained optimization, feasibility, homeomorphism, distortion, projection

1. Introduction

Constrained Optimization (CO) plays a pivotal role across a spectrum of engineering fields, including supply chain, transportation, power systems, and resource allocation. To solve diverse CO problems, various iterative algorithms have been developed and incorporated into commercial solvers such as Gurobi and MOSEK. Despite their success, iterative algorithms often fail to solve challenging CO problems in real-time, limiting their usefulness in time-

*. Corresponding author.

sensitive applications. For instance, solving AC optimal power flow (AC-OPF) problems in real-time is crucial for efficient power grid operations. In these contexts, the inability of iterative algorithms to deliver timely solutions can significantly impair operational efficiency and effectiveness.

Recently, machine learning (ML) schemes have been developed for solving CO in real-time, including the end-to-end (E2E) mapping scheme (Pan et al., 2019; Kotary et al., 2021), the learning-to-optimize (L2O) iterative scheme (Khalil et al., 2016; Chen et al., 2021b), and hybrid approaches (Diehl, 2019; Baker, 2019). For instance, boosted by the universal approximation capacity of neural networks (NNs) (Hornik et al., 1989; Leshno et al., 1993), the E2E 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, achieving a speedup of several orders of magnitude compared to traditional iterative solvers (Pan et al., 2019). This efficiency is particularly evident in complex applications such as optimal power flow (OPF) problems (Guha et al., 2019; Pan et al., 2020; Fioretto et al., 2020; Zamzam and Baker, 2020; Donti et al., 2020; Park et al., 2023; Huang et al., 2024).

In addition to the stringent real-time requirements, ensuring the safety of predicted solutions is significant, meaning that these solutions must consistently satisfy the problem constraints. However, guaranteeing the feasibility of NN solutions poses a non-trivial challenge due to inherent NN prediction errors. Existing methods that aim to ensure solution feasibility are either burdened by high computational costs or lack performance guarantees over general constraints. For detailed discussions on these methods, refer to Sec. 2.

In this paper, we develop *Homeomorphic Projection* (HP) as a novel low-complexity approach to "project" infeasible NN solutions on the constraint set with bounded optimality loss. We make the following contributions:

▷ We consider the optimization problem over a ball-homeomorphic set in Sec. 3, which encompasses all compact convex sets and certain non-convex sets, which is significantly broader than previous works that focused primarily on linear or convex sets. We also provide sufficient conditions to characterize ball-homeomorphic sets.

▷ We propose an HP framework for ensuring NN solution feasibility in Secs. 4 and 5. It includes (i) training a bi-Lipschitz invertible neural network (INN) to approximate a minimum distortion homeomorphic (MDH) mapping between the constraint set and a unit ball, and (ii) performing a simple bisection operation in the unit ball so that the INN-mapped final solution is feasible with minimal distortion-induced optimality loss.

▷ In Secs. 5 and 6, we establish theoretical foundations for the HP framework, including (i) universal approximation capability and bounded distortion of bi-Lipschitz INN for MDH mappings, and (ii) feasibility guarantee, bounded optimality loss, and runtime complexity of the bisection operation for recovering infeasible NN predictions to feasible solutions.

▷ In Sec. 7, we carry out simulations, including four benchmark convex optimization problems and real-world non-convex AC-OPF problems in power grid operations. The results show that homeomorphic projection outperforms existing methods in feasibility and run-time complexity while achieving similar optimality loss.

Table 1: Existing work for ensuring NN solution feasibility.

Existing Work (see Sec. 2 for references)	**Constraint Set**		**Performance Guarantee**		
	Input-adaptive	Non-convex	Feasibility	Optimality bound	Low run-time
Penalty method	✓	✓	✗	✗	✓
Warm-start method	✓	✓	✓	✓	✗
Orthogonal Projection	✓	✓	✓	✓	✗
Sampling approach	✗	✓	✓	✓	✗
DC3	✓	✓	✗	✗	✓
RAYEN	✗	✗ (convex)	✓	✓	✓
Preventive learning	✓	✗ (linear)	✓	✗	✓
Gauge mapping	✓	✗ (linear)	✓	✗	✓
Homeomorphic Projection	✓	✓ (BH)	✓	✓	✓

[1] BH indicates ball-homeomorphic sets, including all compact convex sets and certain non-convex sets.

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[1].

2. Related Work

ML schemes have been developed for solving CO in real-time, including the end-to-end (E2E) mapping scheme (Amos, 2022), the learning-to-optimize (L2O) iterative scheme (Chen et al., 2021b), and hybrid approaches (Kotary et al., 2021). However, guaranteeing the feasibility of NN solutions with respect to the input-dependent constraint is non-trivial. Researchers have developed different approaches to improve the feasibility, and a summary is in Table 1.

Activation layers. Simple constraints can often be directly enforced through specific NN activation layers. For instance, box constraints can be managed using Sigmoid or Tanh activation layers with an additional scaling operation to fit the constraint range. Probabilistic simplex constraints, which require outputs to sum to one and be non-negative, are typically handled with Softmax layers.

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 and Baker, 2020; Fioretto et al., 2020). Considering the optimality condition of CO, Karush–Kuhn–Tucker (KKT) conditions (a set of equations) are treated as equality constraints to improve the performance of NN (Nellikkath and Chatzivasileiadis, 2021a,b; Zhang et al., 2021). However, those approaches do not guarantee feasibility over constraints set due to the prediction error of NN.

Warm-start approach. The NN predictions can serve as effective warm-start points for iterative solvers, potentially reducing the number of iterations required to reach the optimal solution (Diehl, 2019; Baker, 2019; Sambharya et al., 2023, 2024). However, despite this reduction in iterations, the process of solving a warm-start problem may still entail significant computational demands, posing challenges for real-time operational applications.

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

1. This work is a substantial extension of its conference version (Liang et al., 2023).

vertexes and rays is adopted to ensure feasibility under linear constraints (Frerix et al., 2020; Zheng et al., 2021). For general but input-invariant constraint sets, sampling-based methods are theoretically studied in (Kratsios et al., 2021). However, the number of required feasible samples grows exponentially with the dimension of the decision variable, which limits their potential for complex CO problems.

Projection approach. Orthogonal projection can be applied to ensure the feasibility of NN predictions. However, either solving projection problem by optimization solver (Diamond and Boyd, 2016) or applying equivalent projection layers (Amos and 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 in DC3 (Donti et al., 2020). L2O-based methods are also proposed to learn the iterative process of projection by different types of NN (Heaton et al., 2021, 2022). However, those projection-analogous approaches do not guarantee feasibility for general input-adaptive constraints.

Preventive learning. A preventive learning framework is proposed for ensuring linear constraint feasibility in (Zhao et al., 2020, 2023). It initially adjusts inequality constraints to account for NN prediction errors. Subsequently, it trains the NN using mixed-integer programming techniques to limit the worst-case prediction error. Additionally, the worst-case calculation is also related to the NN verification techniques (Fazlyab et al., 2020; Liu et al., 2021; Fazlyab et al., 2021) to offer a form of performance guarantee (Venzke et al., 2020; ul Abdeen et al., 2022). However, preventive learning only works for linear sets, lacks an optimality guarantee, and has high training complexity.

Gauge functions. This line of research utilizes gauge functions (Blanchini and Miani, 2008), a generalization of norms, to constrain NN output. A closed-form bijection between a hypercube and a polytope, termed Gauge Mapping, is proposed to keep the NN output within the polytope (Tabas and Zhang, 2022a,b; Li et al., 2023). For convex constraints, analytic expressions for the gauge function can also be applied to scale infeasible predictions down to the constraint boundary, termed RAYEN (Tordesillas et al., 2023). However, these approaches only work for specific convex sets.

In summary, existing schemes to ensure NN solution feasibility either incur high runtime complexity or lack feasibility and optimality guarantee over general sets. In this paper, we propose *Homeomorphic Projection* as a low complexity scheme to guarantee NN solution feasibility with bounded optimality loss. Our scheme is conceptually related to the projection approach and gauge mapping. However, it is uniquely different in its design, applicability to non-convex constraints, and performance guarantee.

3. Settings and Open Issue to Address

We consider a parametric 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) is denoted as $x^*_\theta\in\arg\min_{x\in\mathcal{K}_\theta}\{f(x,\theta)\}$. The constraint set $\mathcal{K}_\theta$ is compact and specified

by n_{ineq} inequalities: $\mathcal{K}_\theta = \{x \mid g_i(x,\theta) \leq 0, i = 1,\ldots,n_{\text{ineq}}\}$, where $g_i(x,\theta)$ is a continuous function. While our primary formulation in (1) does not explicitly incorporate equality constraints, we remark that equality constraints of constant rank (e.g., linear equations) can be exploited and removed without losing optimality as presented in Appendix A. We thus focus on formulations with only inequality constraints. Nevertheless, we do conduct simulations for problems with linear/nonlinear equality constraints in Sec. 7

We further specify the constraint set as follows:

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

Definition 2 (Homeomorphism) *A mapping $\psi : \mathcal{X} \to \mathcal{Y}$ between two topological spaces is a homeomorphic mapping, i.e., homeomorphism, if it is a continuous bijection with a continuous inverse.*

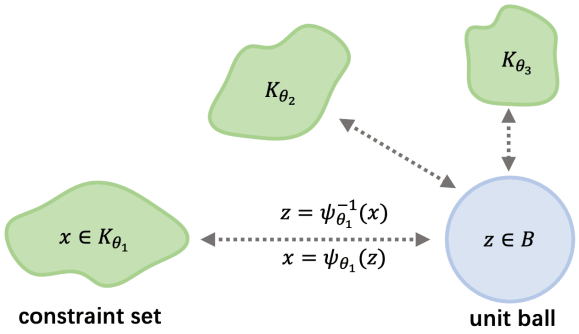

Figure 1: Homeomorphism between constraint sets and a unit ball.

Homeomorphism represents a class of mapping, continuous bijection with a continuous inverse, which preserves topological properties like compactness and connectedness. For example, any invertible linear transformation can be a homeomorphic mapping. While the homeomorphism between two sets (e.g., $\mathcal{K}_\theta$ and $\mathcal{B}$) is to say there exists at least one homeomorphic mapping between them (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 proceed to outline several sufficient conditions under which a set, potentially non-convex, is homeomorphic to a ball:

Proposition 3 *An n-dimensional compact set $\mathcal{K}$ in Euclidean space is homeomorphic to a ball, denoted as $\mathcal{K} \cong \mathcal{B}$, if it satisfies any of the following conditions:*

- *$\mathcal{K}$ is convex or star-convex (Geschke, 2012),*
- *$n = 2$ and $\mathcal{K}$ is simply connected (Ahlfors and Bers, 1960),*
- *$n = 5$ and $\partial\mathcal{K}$ is diffeomorphic to $\mathcal{S}^{n-1}$ (Smale, 1962),*
- *$n \geq 6$ and $\partial\mathcal{K}$ is simply connected (Smale, 1962),*

where $\partial\mathcal{K}$ denotes the boundary of $\mathcal{K}$ and $\mathcal{S}^{n-1}$ represents an $n-1$ dimensional sphere. A set $\mathcal{X}$ is star-convex if there exists a point $x \in \mathcal{X}$ such that the line segment from x to any point in $\mathcal{X}$ is contained in $\mathcal{X}$. A set $\mathcal{X}$ is simply connected if it is path-connected, and any loop within $\mathcal{X}$ can be contracted to a point without leaving $\mathcal{X}$. The term "diffeomorphic" is used when differentiable homeomorphic mapping exists between one set and another.

Based on the characterization of the ball-homeomorphic set in Proposition 3, Assumption 1 is easy to satisfy, e.g., by *any compact convex set*, and *a class of compact and simply-connected non-convex set*. 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 Sec. 1, there have been NN schemes that learn the input-solution mapping $F : \mathbb{R}^d \to \mathbb{R}^n$ for a constrained optimization problem and pass inputs through the NN to obtain high-quality solutions instantly. However, it is non-trivial to ensure NN solution feasibility with respect to the constraints due to inherent NN prediction error. As discussed in Sec. 2, existing feasibility-ensuring methods are either computationally expensive or lack 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.

4. Our Homeomorphic Projection Framework

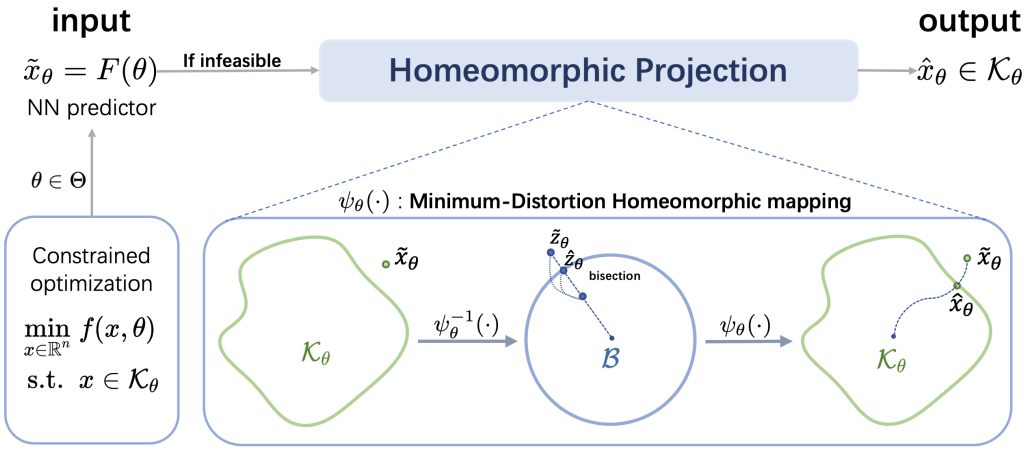

Figure 2: Overview of the HP framework.

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 in 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 (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 $\mathrm{D}(\psi, \mathcal{Z}) = \kappa_2/\kappa_1 \geq 1$, where*

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

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

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 to 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 5 (MDH mapping) *The MDH mapping is defined as the optimal solution for the following problem:*

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

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 distortion of homeomorphic mapping is evaluated over the region, denoted as $\mathcal{X}_\theta$, which contains all possible infeasible predictions with bounded prediction error ϵ_{pre}, for example, $\mathcal{X}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\text{pre}})$[2,3]*.*

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 optimization and challenging to solve. We develop a method to learn an optimal solution approximately by bi-Lipschitz 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 $\mathbb{R}^n$. 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. Nevertheless, if Φ_θ is also **valid** as Definition 6, the second step in the HP framework guarantees to "project" an infeasible NN predicted solution back to the feasible set $\mathcal{K}_\theta$.

Definition 6 (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 center of the unit ball to a feasible point in $\mathcal{K}_\theta$.*

2. $+$ between sets represents the Minkowski addition, defined as $\mathcal{X} + \mathcal{Y} = \{x + y \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$.
3. $\mathcal{B}(x, r)$ represents a ball with a radius of r and is centered in x.

4.2 Homeomorphic Bisection

Given an infeasible NN prediction $\tilde{x}_\theta \notin \mathcal{K}_\theta$ and a valid INN mapping Φ_θ, we perform homeomorphic bisection to recover a feasible solution $\hat{x}_\theta$ as:

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

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 interior 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.

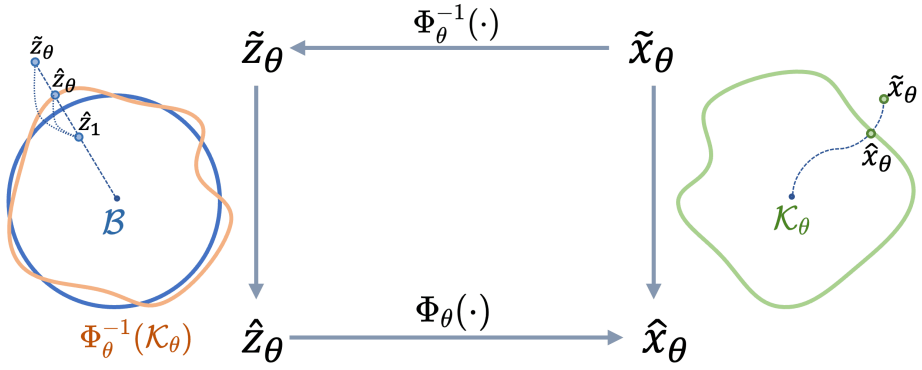

Figure 3: Illustration of homeomorphic bisection.

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 boundary of unit ball, 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 INN training in Sec. 6.

Algorithm 1 Homeomorphic bisection to recover solution feasibility.

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

initialize total iteration steps k
calculate $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$, set $n = 0,\ \alpha_l = 0,\ \alpha_u = 1$
while $n \leq k$ **do**
  bisection: $\alpha_n = (\alpha_l + \alpha_u)/2$
  candidate: $\hat{z}_{n+1} = \alpha_n \cdot \tilde{z}_\theta$
  if $\Phi_\theta(\hat{z}_{n+1}) \in \mathcal{K}_\theta$ **then**
    increase lower bound: $\alpha_l \leftarrow \alpha_n$
  else
    decrease upper bound: $\alpha_u \leftarrow \alpha_n$
  end if
  $n \leftarrow n + 1$
end while
set $\alpha^* = \alpha_l$

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

5. Learning Input-dependent MDH Mapping

In this section, we propose an efficient 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 with bounded distortion. We then reformulate the MDH mapping problem in (4) and present the INN training procedure.

5.1 Invertible Neural Network (INN)

INN is essentially an invertible NN that is differentiable in forward and inverse directions; thus, it can be employed to learn a homeomorphic mapping. This is owing to its design on each invertible layer (Papamakarios et al., 2021). Particular to our study, we build our INN by stacking up multiple layers, each from the following two adapted from those in (Dinh et al., 2014; Kingma and Dhariwal, 2018):

- **Bi-Lip affine coupling layer**

$$\text{Forward:} \quad [y_1, y_2] = [x_1, \mathrm{S}(\mathbf{w}(x_1), L) \cdot x_2 + \mathbf{b}(x_1)], \tag{6}$$

$$\text{Inverse:} \quad [x_1, x_2] = [y_1, (y_2 - \mathbf{b}(y_1))/\mathrm{S}(\mathbf{w}(y_1), L)], \tag{7}$$

  where $x = [x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}]$, $\mathbf{w}(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ and $\mathbf{b}(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ are two regular NNs with learnable parameters, which take x_1 as input and predict the weight and bias respectively for the element-wise affine transformation of x_2. The function $S(x, L) := \exp(\log(L) \cdot \tanh(x))$ limits the upper and lower bound of the output weights, such that $S(x, L) \in (1/L, L)$, where $L > 1$ is a predefined parameter. Such an output constraint of weight ensures the bi-Lipschitz property of this layer, which is crucial to bound the distortion.

- **Invertible linear layer**:

$$\text{Forward:} \quad y = Wx + b, \qquad \text{Inverse:} \quad x = W^{-1}(y - b), \tag{8}$$

  where $W \in \mathbb{R}^{n\times n}$ is an invertible and learnable matrix, and $b \in \mathbb{R}^n$ is the learnable bias. The invertibility of W is ensured by construction such as $W = W_P W_L(W_U + \text{diag(s)})$, where W_P is a permutation matrix, W_L is a lower triangular matrix, W_U is an upper triangular matrix, and $s \in \mathbb{R}^n$ is the non-zero diagonal elements.

The bi-Lip affine coupling layer first partitions the input x into two disjoint subsets denoted as $[x_1, x_2]$. It performs an identical mapping for x_1 and applies an element-wise affine mapping for x_2. The invertible linear layer directly employs linear transformation for the input. As a result, both these layers exhibit properties of bijections with continuous inverses, making them homeomorphic mappings.

Existing work has shown that the INN composed of affine coupling[4] and invertible linear layers can approximate any (differentiable) homeomorphic mapping (Teshima et al., 2020; Koehler et al., 2021; Ishikawa et al., 2022), i.e., it can transform a unit ball to any constraint set under Assumption 1 and vice versa through closed-form INN forward/inverse calculation.

In our study, we design INN with unique bi-Lip affine coupling and invertible linear layers so that the resulting INN not only has the universal approximation capability, but also has bounded distortion over $\mathbb{R}^n$, i.e., for any infeasible solution in $\mathbb{R}^n$, the "projected" feasible solution by homeomorphic bisection in (5) has a bounded distortion-induced optimality loss.

The following proposition outlines the universal approximation capability and the upper bound of distortion of the proposed INN:

Proposition 7 *Given an m-layer INN, denoted as $\Phi = \Phi^m \circ \cdots \circ \Phi^l \circ \cdots \circ \Phi^1$, where Φ^l an invertible linear layer for $l = 1, 3, \cdots, m-1$ and a bi-Lip affine coupling layer with a hyper-parameter $L > 1$ for $l = 2, 4, \cdots, m$. The following properties hold:*

- *Φ is capable of representing any n-dimensional differentiable homeomorphic mapping over a compact set, requiring $m = \mathcal{O}(n \cdot \lceil c_1/L \rceil)$ layers, where c_1 is a constant associated with the target mapping.*
- *Φ assures a bounded distortion over $\mathbb{R}^n$, expressed as $\mathrm{D}(\Phi, \mathbb{R}^n) \le c_2 L^m$, where c_2 is a constant related to those invertible linear layers of the INN.*

The complete proof is in Appendix D.1. We make the following **remarks**. First, to exactly represent a differentiable homeomorphic mapping over a compact set, i.e., transform a unit ball to the desired constraint set, the number of layers is influenced by the dimension of the mapping and the hyperparameter L. On the other hand, L contributes to bounding the distortion over $\mathbb{R}^n$, thereby reducing the optimality loss of the "projected" infeasible solution. Consequently, we can leverage the INN with bi-Lip affine coupling and invertible linear layers to approximate homeomorphic mappings with a bounded distortion in our HP framework.

4. The affine coupling layer (Dinh et al., 2014; Kingma and Dhariwal, 2018) has no upper bound constraints for the weights of the input-dependent affine transformation in (6) and (7).

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

In the following sections, we will introduce an unsupervised training approach to approximate the homeomorphism and further minimize the distortion over the region of interest. The quantitative analysis detailing the optimality loss under INN approximation error and distortion will be elaborated in Sec. 6

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

$$\max_{\psi_\theta \in \mathcal{H}^n} \log \mathrm{V}(\psi_\theta(\mathcal{B})) \quad \text{s.t. } \psi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta, \tag{9}$$

where $\mathrm{V}(\psi_\theta(\mathcal{B}))$ computes the volume of set $\psi_\theta(\mathcal{B})$ and the constraint means that the set $\psi_\theta(\mathcal{B})$ is a subset of $\mathcal{K}_\theta$.

The complete proof is in Appendix D.2. Intuitively, the proposition says that any feasible homeomorphic mapping resulting in $\psi_\theta(\mathcal{B}) = \mathcal{K}_\theta$ must maximize the volume of the 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 (9):

$$\min_{\psi_\theta \in \mathcal{H}^n} \log \mathrm{D}(\psi_\theta^{-1}, \mathcal{X}_\theta) \quad \text{s.t. } \psi_\theta \in \arg\max\{\text{Problem in (9)}\}. \tag{10}$$

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

5.3 Unsupervised INN Training

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 either a bi-Lip affine coupling layer or an invertible linear layer. Importantly, due to the invertible design of these layers, the singular values for the Jacobian matrix of Φ_θ^l exist and have a closed-form expression. For example, the singular values for the affine coupling layer are either 1 or the weights of the affine transformation. See Appendix C for detailed formulations. As will become clear later, these closed-form singular values greatly simplify the process of homeomorphism and distortion approximation. We then denote the sorted singular values for the Jacobian matrix J of INN as $\sigma_1(\mathrm{J}) \geq ... \geq \sigma_n(\mathrm{J})$.

We employ the following loss function and maximize it to train an INN Φ_θ with m layers for learning the MDH mapping under the reformulated problem in (10) in an unsupervised

manner:

$$\mathcal{L}(\Phi_\theta) = \widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B})) - \lambda_1 \mathrm{P}(\Phi_\theta(\mathcal{B})) - \lambda_2 \widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta), \tag{11}$$

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 (9) as:

$$\widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B})) = \frac{1}{\mathrm{V}(\mathcal{B})} \int_{\mathcal{B}} \sum_{k=1}^{n} \sum_{l=1}^{m} \log \sigma_k(\mathrm{J}_{\Phi_\theta^l}(z^l)) \mathrm{d}z + \log \mathrm{V}(\mathcal{B}), \tag{12}$$

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

$\mathrm{P}(\Phi_\theta(\mathcal{B}))$ is the penalty term for the constraint violation of $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$ in (9) as:

$$\mathrm{P}(\Phi_\theta(\mathcal{B})) = \int_{\mathcal{B}} \|\mathrm{ReLU}(\mathbf{g}(\Phi_\theta(z), \theta))\|_1 \mathrm{d}z, \tag{13}$$

where $\mathrm{ReLU}(\cdot) = \max\{0, \cdot\}$ and $\mathbf{g}(\Phi_\theta(z), \theta)$ calculates the residual for each inequality constraint as $[g_1(\Phi_\theta(z), \theta), \ldots, g_{n_{\text{ineq}}}(\Phi_\theta(z), \theta)]$.

$\widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$ is a computable approximation of the log-distortion term $\log \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$ in (10) as:

$$\widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta) = \sup_{z^1 \in \mathcal{Z}_\theta} \{\sum_{l=1}^{m} \log \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} - \inf_{z^1 \in \mathcal{Z}_\theta} \{\sum_{l=1}^{m} \log \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))\}, \tag{14}$$

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

We have the following observations for the approximations.

Proposition 9 *The two approximation terms in* (12) *and* (14) *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 D.3 and D.4. We make the following **remarks**. First, the above proposition implies that the loss function defined in (11) actually provides a surrogate for the problem presented in (10). Specifically, maximizing the lower bound of the volume term and minimizing the penalty term in (11) indeed approximate the feasible homeomorphic mapping under reformulation in (8). Meanwhile, we also try to minimize the upper bound of the distortion term in (11), which in turn guides the INN to find an approximated homeomorphic mapping with less distortion. Consequently, we aim to maximize the loss function in (11) as a means to approximate the MDH mapping under the equivalent reformation outlined in (10). Additionally, as demonstrated in the proof in Appendix D.4, minimizing the distortion also contributes to reducing the gap between the volume estimator $\widehat{\mathrm{V}}(\Phi_\theta(\mathcal{B}))$ and the exact log-volume term $\log \mathrm{V}(\Phi_\theta(\mathcal{B})$. Thus, the loss function defined in (11) serves as an efficient surrogate loss for the original MDH mapping problem.

Algorithm 2 Unsupervised training of INN to learn MDH mapping.

Input: m-layer conditional INN Φ, dataset $\mathcal{D}_1 = \{\theta_i\}_{i=1}^M \subseteq \Theta$, $\mathcal{D}_2 = \{z_j\}_{j=1}^N \subseteq \mathcal{B}$.
Output: trained INN.

initialize total epoch E, batch size B, and coefficients λ_1 and λ_2 for loss function.
while $e \leq E$ **do**
  batch sampling: $\{\theta_i\}_{i=1}^B \subset \mathcal{D}_1, \{z_i\}_{i=1}^B \subset \mathcal{D}_2$
  compute volume term: $\widehat{\mathrm{V}}(\Phi) = \frac{1}{Bn}\sum_{i=1}^B \sum_{k=1}^n \sum_{l=1}^m \log \sigma_k(\mathrm{J}_{\Phi_{\theta_i}^l}(z_i^l))$
  compute penalty term: $\mathrm{P}(\Phi) = \frac{1}{B}\sum_{i=1}^B \|\mathrm{ReLU}(\mathbf{g}(\Phi_{\theta_i}(z_i), \theta_i))\|_1$
  compute distortion term: $\widehat{\mathrm{D}}(\Phi) = \frac{1}{Bm}\sum_{i=1}^B [\sum_{l=1}^m \log \sigma_1(\mathrm{J}_{\Phi_{\theta_i}^l}(z_i^l)) - \log \sigma_n(\mathrm{J}_{\Phi_{\theta_i}^l}(z_i^l))]$
  compute loss function $\mathcal{L}(\Phi) = \widehat{\mathrm{V}}(\Phi) - \lambda_1 \mathrm{P}(\Phi) - \lambda_2 \widehat{\mathrm{D}}(\Phi)$
  update INN: $\Phi \leftarrow \mathrm{Adam}(\mathcal{L}(\Phi))$
  $e \leftarrow e + 1$
end while
Return trained INN Φ

To maximize the loss function in (11) for INN training, we generate quasi Monte Carlo (QMC)[5] samples $\{z_i\}_{i=1}^N \subset \mathcal{B}$ to approximate the integration terms in (12) and (13). For practical implementation of distortion regularization in (14), we first substitute both the sup and inf operators with the expectation operator. It results in an "average" distortion computed as $\mathbb{E}_{z^1 \in \mathcal{Z}_\theta}[\sum_{l=1}^m \log \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l)) - \log \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))]$. This substitution has been demonstrated to be efficient in existing Lipschitz regularization schemes (Virmaux and Scaman, 2018; Behrmann et al., 2021), and it also contributes to the stabilization of INN training.

Next, we sample $z^1 \in \mathcal{B}(0, R)$, where $R \geq 1$ is a hyper-parameter that defines the region such that $\mathcal{Z}_\theta$ is a subset of $\mathcal{B}(0, R)$. The distortion can then be expressed as $\mathrm{D}(\Phi_\theta, \mathcal{B}(0, R))$, or equivalently as $\mathrm{D}(\Phi_\theta^{-1}, \Phi_\theta(\mathcal{B}(0, R)))$. Nevertheless, we typically set $R = 1$ in practice because of (i) training efficiency, where we only need to sample over the unit ball $\mathcal{B}$ to simultaneously compute the three terms in (12), (13), and (14) via INN forward propagation; and (ii) the distortion remains bounded even outside the evaluated region due to the proposed bi-Lipschitz INN, as demonstrated in Proposition 7. Consequently, the efficient implementation of maximizing the surrogate loss in (11) further demonstrates the efficiency of the unsupervised training scheme for the MDH mapping problem.

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 extend the loss in (11) to

$$\mathcal{L}(\Phi) = \mathbb{E}_\theta[\mathcal{L}(\Phi_\theta)], \tag{15}$$

where $\mathcal{L}(\Phi_\theta)$ is defined in (11) and $\theta \in \Theta$ is uniformly sampled.

In summary, to train the input-dependent MDH mappings, we utilize the conditional INN and train it based on the loss function in (15). As shown in Alg. 2, in each iteration, we sample a batch of $\theta \in \Theta$ and $z \in \mathcal{B}$, approximate the three terms in (11), and use the

5. 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 and Pillichshammer, 2010).

Adam optimizer to maximize the loss function, similar to the process of training regular NNs (Kingma and 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.

6.1 Feasibility, Optimality, and Run-time Complexity

Theorem 10 *For constraint set $\mathcal{K}_\theta$ under Assumption 1, given an infeasible NN prediction $\tilde{x}_\theta = F(\theta) \notin \mathcal{K}_\theta$ with bounded prediction error to the optimal solution, defined as $\epsilon_{\text{pre}} = \sup_{\theta\in\Theta}\{\|F(\theta) - x^*_\theta\|\}$, and a valid m-layer INN Φ_θ with bounded approximation error $\epsilon_{\text{inn}} = \sup_{\theta\in\Theta}\{\mathrm{d_H}(\Phi_\theta(\partial\mathcal{B}), \partial\mathcal{K}_\theta)\}$*[6]*, running the bisection procedure in Alg. 1 for k steps will return a solution $\hat{x}^k_\theta$ such that:*

- *it is guaranteed to be feasible, i.e., $\hat{x}^k_\theta \in \mathcal{K}_\theta$;*
- *it has a bounded optimality loss as*

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

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

The corresponding run-time complexity is $\mathcal{O}(k(mn^2 + G))$, where G is the complexity for verifying the inequality constraints.

The complete proof is in Appendix E. We make the following **remarks**. First, given a valid INN, the bisection algorithm is guaranteed to return a feasible solution due to the initial feasible point, i.e., $\Phi_\theta(0) \in \mathcal{K}_\theta$. It then searches for better feasible points, moving from the interior to the constraint boundary, as illustrated in Fig. 3. When $k \to \infty$, the returned feasible solution will be located on the constraint boundary. 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^k_{bis}.

The prediction error ϵ_{pre}, which measures the gap between the predicted and optimal solutions, is determined by the quality of the provided predictor. The approximation error ϵ_{inn} quantifies the distance between the INN-approximated constraint set $\Phi_\theta(\mathcal{B})$ and the exact constraint set $\mathcal{K}_\theta$. Thus, our training scheme aims to minimize this approximation error based on the reformulation of the homomorphism constraint, as detailed in Proposition 8.

The distortion of Φ_θ^{-1} shown in our loss function (11) is designed to be minimized over set $\mathcal{X}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\text{pre}})$. This ensures that, when the approximation error is minimized, the distortion over the region $\mathcal{Y}_\theta$ will be close to the one over the region $\mathcal{X}_\theta$. In practice, we sample over $\mathcal{B}$ to approximate the distortion of Φ_θ^{-1} over the region $\Phi_\theta(\mathcal{B})$. This approach

6. $\mathrm{d_H}(\cdot,\cdot)$ represents the Hausdorff distance between two sets, defined as $\mathrm{d_H}(\mathcal{X},\mathcal{Y}) = \max\{\mathrm{d_h}(\mathcal{X},\mathcal{Y}), \mathrm{d_h}(\mathcal{Y},\mathcal{X})\}$, where $\mathrm{d_h}(\mathcal{X},\mathcal{Y}) = \sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}}\{\|x-y\|\}$.

is chosen for its training efficiency, as discussed after Proposition 9. Despite this simplification, it still performs robustly due to the bi-Lipschitz INN design, which ensures that the distortion remains bounded even outside the evaluated region, as shown in Proposition 7. Nevertheless, in the case of a low-quality NN predictor with a large ϵ_{pre}, we can sample from $\mathcal{B}(0, R)$ with radius $R > 1$, to regulate the distortion over a larger region during training, and further reduce the distortion-induced optimality gap.

The bisection error ϵ_{bis}^k can be exponentially reduced by increasing the maximum steps k, as detailed in Alg. 1. This property allows for precise control over the error tolerance, offering a flexible trade-off between computational effort and solution accuracy.

The overall run-time complexity, i.e., the number of arithmetic operations, when executing k-step bisection is $\mathcal{O}(k(mn^2 + G))$. It includes the INN inverse/forward calculation complexity $\mathcal{O}(mn^2)$ and the constraint verification complexity $\mathcal{O}(G)$ at each bisection. If the inequality constraint $g_i(x, \theta)$ is a linear function for all $i = 1, \cdots, n_{\text{ineq}}$, then $G = n \cdot n_{\text{ineq}}$. In contrast, iterative algorithms such as interior point methods have a complexity of $\mathcal{O}((n+n_{\text{ineq}})^3)$ at each iteration due to the matrix inversion operation. Therefore, our homeomorphic projection exhibits lower run-time complexity when recovering feasible solutions for ball-homeomorphic constraint sets.

From the analysis above, the distortion and the approximation error are significant contributors to the optimality gap. This validates our design of the MDH mapping problem as defined in (4), as well as our proposed INN loss function given in (11). By carefully managing these elements, we can effectively reduce the optimality gap, leading to more accurate and reliable solutions. It's also important to note that the above results are analyzed under worst-case scenarios. In practical applications, the optimality loss of the homeomorphic projection is often much smaller, resulting in better empirical performance.

The optimality analysis provided in Theorem 10 applies under a general but valid INN. In the following corollary, we present the optimality analysis, especially for the projection distance, under a valid INN in several special cases. These scenarios yield a smaller optimality gap.

Corollary 11 *Given an infeasible solution $\tilde{x}_\theta \notin \mathcal{K}_\theta$, let $\hat{x}_\theta^*$ be the feasible solution retuned by standard projection operation as $\hat{x}_\theta^* \in \text{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \arg\min_{y \in \mathcal{K}_\theta}\{\|\tilde{x}_\theta - y\|\}$. The minimum projection distance is then denoted as $\epsilon_{\text{pro}} = \|\hat{x}_\theta^* - \tilde{x}_\theta\|$. Let $\hat{x}_\theta$ be the solution returned by homeomorphic bisection with a valid INN as in Eq. (5). The projection distance of $\hat{x}_\theta$ is bounded as:*

$$\epsilon_{\text{pro}} \leq \|\hat{x}_\theta - \tilde{x}_\theta\| \leq \text{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta')(2\epsilon_{\text{inn}} + \epsilon_{\text{pro}}). \tag{16}$$

Further, the upper bound can be tightened under the following special cases:

- *if the valid INN is an outer approximation as $\Phi_\theta(\mathcal{B}) \supseteq \mathcal{K}_\theta$, then the homeomorphic projection distance is bounded as $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \text{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta')(\epsilon_{\text{inn}} + \epsilon_{\text{pro}})$.*
- *if the valid INN is an inner approximation as $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$, then the homeomorphic projection distance is bounded as $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \text{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta')(\epsilon_{\text{inn}} + \epsilon_{\text{pro}})$.*
- *if the INN is feasible as $\Phi_\theta(\mathcal{B}) = \mathcal{K}_\theta$, then the homeomorphic projection distance is bounded as $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \text{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta')\epsilon_{\text{pro}}$.*

- *if the INN is feasible and also isometric as* $\mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}'_\theta) = 1$*, then the homeomorphic projection distance reaches the lower bound as* $\|\hat{x}_\theta - \tilde{x}_\theta\| = \epsilon_{\mathrm{pro}}$.

Here $\mathcal{Y}'_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\mathrm{pro}}, \epsilon_{\mathrm{inn}}\})$ *and* $\mathcal{X}'_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\mathrm{pro}})$.

The complete proof is in Appendix F. This corollary reveals the gap between homeomorphic projection and standard orthogonal projection, which is influenced by the distortion and approximation error of the INN. To achieve the minimum projection distance, we need to find an exact feasible homeomorphic mapping with a unit distortion. However, for general constraint sets, the distortion of feasible homeomorphic mapping cannot be 1 by Mazur–Ulam theorem (Väisälä, 2003). Thus, in practice, we aim to regulate the distortion and minimize the approximation error of INN by tuning the coefficients in loss function (11).

6.2 Universal Validity Condition of INN

Theorem 12 *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.,* $\forall \theta^0 \in \mathcal{D}_1$, $\Phi_{\theta^0}(0) \in \mathcal{K}^\circ_{\theta^0}$*. 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* $C_0 = \sup_{\theta_1, \theta_2 \in \Theta, \theta_1 \neq \theta_2} \{\mathrm{d_H}(\partial\mathcal{K}_{\theta_1}, \partial\mathcal{K}_{\theta_2})/\|\theta_1 - \theta_2\|\}$, $C_1 = \mathrm{Lip}(\Phi(0, \cdot), \Theta)$, $C_2 = \arg\sup_{r>0}\{\mathcal{B}(\Phi_{\theta^0}(0), r) \subseteq \mathcal{K}_{\theta^0},\ \forall \theta^0 \in \mathcal{D}_1\}$.

The complete proof is in Appendix G. Here 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 $\mathcal{D}_1$.

We make the following **remarks**. Theorem 12 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 10. First, we need to make the INN valid under finite samples, i.e., $\Phi_{\theta^0}(0) \in \mathcal{K}^\circ_{\theta^0}$, 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 with proper training. This observation is perhaps not surprising, as we penalize constraint violations $\mathrm{P}(\Phi_{\theta^0}(\mathcal{B}))$ in (13) for all points in $\mathcal{B}$. Naturally, the center $0 \in \mathcal{B}$ is likely mapped to an internal point $\Phi_{\theta^0}(0) \in \mathcal{K}^\circ_{\theta^0}$.

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 $\mathcal{O}((\mathrm{diam}(\Theta)/r_c)^d)$. 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 Limitations

Training complexity. First, we need QMC samples in a unit ball $\{z_i\}_{i=1}^N \subset \mathcal{B}$ to approximate three terms in (11) 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 and Ba, 2014; Paszke et al., 2019). Due to the closed-form expressions of (15) by INN parameters, the training computation depends on the forward-backward propagation of the INN, which can be executed efficiently on a GPU.

Scalability. 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 and Dhariwal, 2018; Papamakarios et al., 2021). On the other hand, efficient INN training needs a closed-form singular value representation, limiting the choices of invertible layers. We may also leverage the numerical approximation of the loss function under general invertible layers, as discussed in Appendix C.

Limitations. The limitation of the HP framework also lies in Assumption 1, which assumes the constraint set to be homeomorphic to a unit ball. If we apply the HP framework to a non-ball-homeomorphic set, e.g., a donut, two scenarios may occur: (i) the INN approximation error is small, but the distortion becomes infinitely large as it attempts to stretch a ball into a donut shape, and (ii) the distortion is low, but the approximation error becomes large, meaning we may only capture a subset of the constraint set. Although feasibility can still be guaranteed under a valid INN, the projection error could be substantial due to either significant distortion or a large approximation error of INN. To address more general constraints, such as disconnected sets or manifold constraints, we discuss potential solutions in Appendix B.

7. Numerical Experiments

We carry out simulations to (i) evaluate the proposed INN unsupervised-learning scheme to approximate MDH mappings between complex 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 standard convex optimization problems and non-convex AC-OPF problems in power grid operation. The detailed setting of hyper-parameters, INN implementation, and NN predictor implementation are in Appendix I. The code to reproduce our results is available at

https://github.com/emliang/Homeomorphic-Projection.

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 quadratic inequalities as:

$$\mathcal{K}_\theta = \{x \in \mathbb{R}^2 \mid x^\mathsf{T} Q_i x + q_i^\mathsf{T} x + b_i \leq 0,\ i = 1, \cdots, 6\}, \tag{17}$$

where $Q_i \in \mathbb{R}^{2\times 2}, q_i \in \mathbb{R}^2, b_i \in \mathbb{R}$. The input parameters for $\mathcal{K}_\theta$ is the denoted as $\theta = \{Q_i, q_i, b_i\}_{i=1}^6$. By changing the input parameters θ such that assumption 1 is valid, the constraint set $\mathcal{K}_\theta$ can be convex or non-convex.

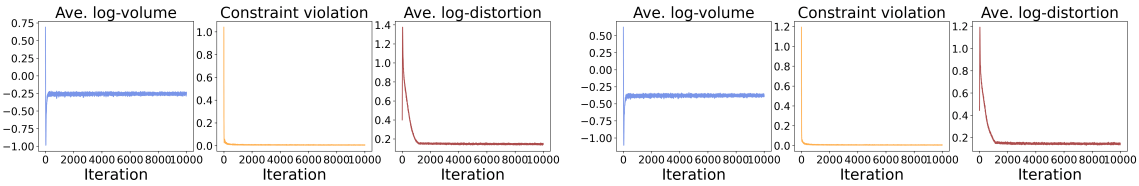

Figure 4: Unsupervised training for INN approximated MDH mappings from 2-norm ball (left) and ∞-norm ball (right) to the constraint set $\mathcal{K}_\theta$ under different input parameters. The log-volume term, constrain violation, and log-distortion term, are computed based on Alg. 2.

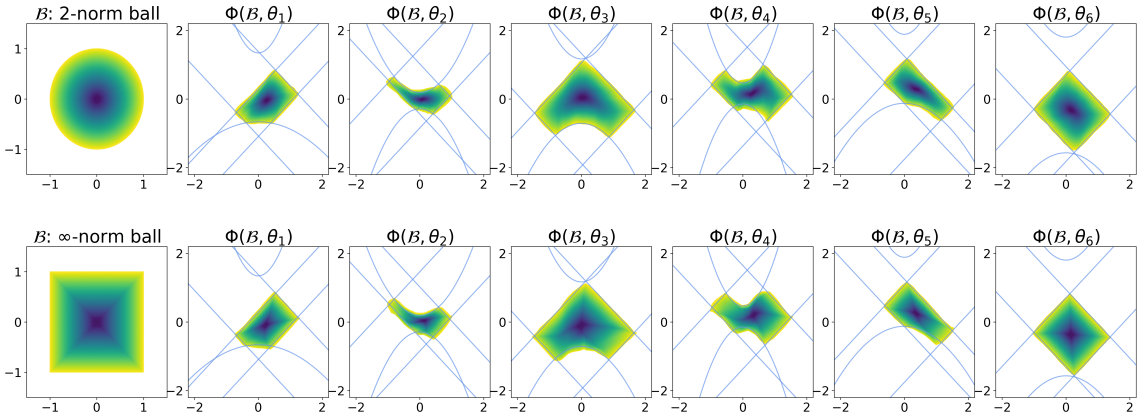

Figure 5: Approximated constraint set $\Phi(\mathcal{B},\theta)$ by trained MDH mappings from 2-norm ball (top) and ∞-norm ball (bottom) under new input parameters. The blue curves are the inequalities defining the constraint set $\mathcal{K}_\theta$ in (17). The color of points in the approximated constraint sets represents the norm of their one-to-one points in the initial ball.

Following the loss function in (15) and the unsupervised training algorithm in Alg. 2, we train two MDH mappings from a 2-norm ball and a ∞-norm ball to the parametric constraint set $\mathcal{K}_\theta$, respectively. The average log-volume term, constraint violation, and average log-distortion term during training iterations are depicted in Fig. 4. First, we observe that the volume is maximized, and the constraint violation converges to zero. This indicates that the approximated homeomorphism under the reformulation in (9) between two sets is achieved at the early stage of the training process. After that, the average log-distortion term decreases over iterations, showcasing the effectiveness of our unsupervised training scheme.

After completing the training phase, we visualize the constraint approximation $\Phi_\theta(\mathcal{B}) = \Phi(\mathcal{B},\theta)$ under varying input parameters θ within the test dataset, as depicted in Fig. 5. Our observation is that the mapped set closely approximates the true constraint set, reaching the constraint boundary, which is indicated by the blue curves. Despite the varying geometric structures exhibited by the constraint $\mathcal{K}_\theta$ under different input parameters, the trained INN

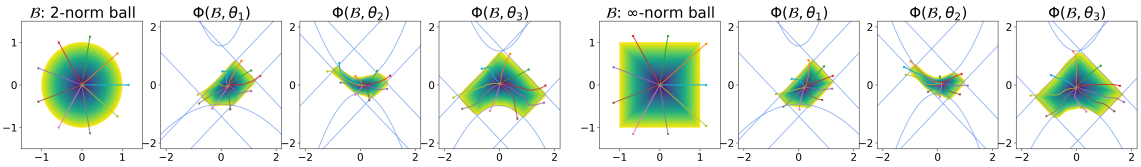

Figure 6: The visualization of several bisection trajectories in the 2-norm ball (left) and ∞-norm ball (right), and their corresponding ones in the constraint set after INN approximated MDH mapping. In the unit ball, we connect the infeasible points to the center and conduct bisection to find the feasible point with respect to the constraint set.

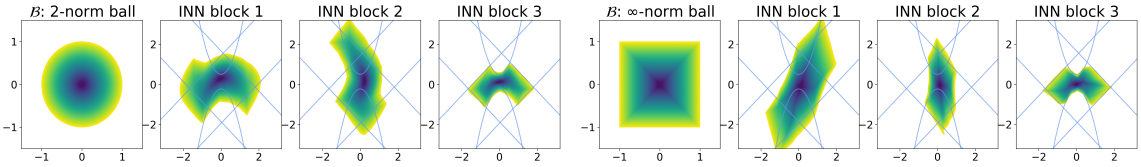

Figure 7: The visualization of the 3-block INN approximated MDH mapping from 2-norm ball (left) and ∞-norm ball (right) to the constraint set $\mathcal{K}_\theta$. Each INN block consists of one linear layer and one coupling layer. Given the sampled points from the initial ball as the input of INN, the output by each block of INN is visualized.

Φ consistently showcases good generalization capabilities, effectively fitting the constraint set under new input parameters.

Subsequently, we execute the homeomorphic bisection shown in Alg. 1 to project the infeasible points back to the constraint under varying inputs, as depicted in Fig. 6. Similar to the illustration of the bisection process in Fig. 3, the bisection conducted along the straight line in the unit ball is equivalent to the bisection carried out along the corresponding curve in the constraint set. We observe that the feasible point over the constraint boundary along the curve is close to the original infeasible points, which underscores the effectiveness of the MDH mapping design and the bisection algorithm.

Further, we visualize the process of the INN for transforming the unit ball to the complex constraint set in Fig. 7. Owing to the universal approximation capabilities demonstrated in Proposition 7, the 3-block INN already displays a potent ability for approximating complex constraint learning. Therefore, in the following experiments, we fix the number of INN blocks to 3 to balance the network complexity and approximation capabilities.

Additionally, we observe that there are no noticeable differences between the ∞-norm ball and the 2-norm ball when used as initial sets for approximating complex constraints. However, in high-dimensional scenarios, the ∞-norm ball (or unit cube $[0, 1]^n$) is our preferred choice, as it facilitates QMC sampling and offers numerical stability as maintaining an invariant volume of 1, while the volume of 2-norm ball converges to zero in high dimension.

In summary, the INN approximated MDH mapping Φ_θ derives a well-fitted constraint set under different input parameters through the proposed unsupervised training scheme. In the next section, we will test the homeomorphic bisection with trained INN in more challenging optimization problems to show its ability to achieve feasibility with minor optimality loss.

7.2 Ensuring NN Solution Feasibility

We then tackle the constrained optimization problem using the HP framework. We select four convex problems with different constraint sets: convex quadratic programming (**QP**), second-order cone programming (**SOCP**), convex quadratically constrained quadratic programming (**QCQP**), and semidefinite programming (**SDP**); and one real-world case: alternating current optimal power flow **AC-OPF**, which is a non-convex QCQP. Detailed experimental settings for these problems are provided in Appendix I.

We evaluate our method based on the following criteria: (i) **Feasibility**: we calculate the percentage of feasible instances out of testing instances and the average constraint violation; (ii) **Optimality**: we compute the mean absolute percentage error for the solution and objective value to assess optimality, including all predictions and infeasible ones only; (iii) **Speedup**: we record the average inference time (NN inference time + post-processing time) of testing instances and calculate the speedup compared to the iterative optimizer, where all NN-based approaches are executed in batch on the NVIDIA A800 GPU, and the solver-based approaches are executed in parallel on the AMD EPYC 7763 CPU.

We compare the following approaches to solve constrained optimization:

- **Optimizer**: for convex problems, we use CVXPY (Diamond and Boyd, 2016) with MOSEK, a general interior-point solver, to obtain the optimal solution. For AC-OPF problems, we adopt PYPOWER, a specialized interior-point solver (Zimmerman et al., 1997; Brown et al., 2017), to solve it efficiently. Optimizer-based solutions are designated as the ground truth compared with the following NN-based methods.

- **NN predictor**: NN maps the input parameter to the solution using a fully connected NN with ReLU activation and residual connections. We select two NN predictors for constrained optimization problems.

    1. **Eq-NN-S**: it first maps the input to a subset of decision variables and reconstructs the others by solving the equality constraints (Pan et al., 2019; Zamzam and Baker, 2020; Pan et al., 2022). The training is performed by supervised learning on optimal input-solution pairs with MSE and regularized by the penalty for constraint violation. The training dataset is prepared by the iterative solver with sampled input parameters.
    2. **Eq-NN-U**: it first maps the input to a subset of decision variables and reconstructs the others by solving the equality constraints (Donti et al., 2020; Huang et al., 2021, 2024). The training is performed by unsupervised learning, minimizing the objective function and regularized by the penalty for constraint violation.

- **Post-processing module**: the NN-predicted solution may be infeasible, so we compare different post-processing techniques for the infeasible predictions.

1. **Proj**: standard orthogonal projection is adopted for post-processing, which is solved by the iterative Optimizer (Chen et al., 2021a).
2. **WS**: the infeasible predicted solution is used as the warm-start initialization point for the iterative Optimizer (Diehl, 2019; Baker, 2019).
3. **D-Proj**: this is proposed in DC3 (Donti et al., 2020), which applies gradient descent in a differentiable manner to minimize constraint violation for infeasible predictions. We tune the hyper-parameters for gradient descent, such as stepsize and momentum coefficient for each problem for convergence (Donti et al., 2020).
4. **H-Proj**: for the infeasible solution of NN, we apply bisection in Alg. 1 with the INN-approximated MDH mapping to recover feasibility.

We **remark** that homeomorphic projection is a general projection module that can be integrated into existing NN predictors. Therefore, we adopt two initial NN predictors (Eq-NN-S and Eq-NN-U) with different levels of feasibility and optimality. The same infeasible solutions by the NN predictor are then processed by the different post-processing approaches mentioned above. This comparison ensures the fairness of different methods and also tests the robustness of HP under different NN predictors.

7.2.1 Convex Constrained Optimization

We first evaluate homeomorphic projection on four benchmark convex problems, including QP, QCQP, SOCP, and SDP, adhering to the hierarchy of convex programs outlined in (Boyd et al., 2004). Detailed formulations of these problems are provided in Appendix I.1. We generate training and testing cases using randomly sampled parameters based on the public code (Donti et al., 2020) and the basic examples in the CVXPY documents. The performance of different methods for those convex problems is presented in Tables 2 and 3. We have the following observations.

Existing NN-based predictors, under various training schemes, fail to guarantee the feasibility of the predicted solutions due to inherent prediction errors. Since the optimal solution of a constrained optimization problem is typically located at the boundary with active constraints, any non-zero approximation error of the NN predictor may shift the prediction outside of the boundary, resulting in infeasibility. While equality constraints can be managed through schemes such as the equality completion method (Pan et al., 2020; Donti et al., 2020), inequality constraints remain a challenge and prevent a full feasibility guarantee. Integrating penalty functions into the NN loss function has been shown to mitigate constraint violations during training. However, this approach cannot assure the feasibility of predictions under new samples due to NN generalization errors, thereby necessitating the use of various post-processing techniques to restore feasibility. We then evaluate four post-processing approaches to address infeasible predictions from three NN predictors with different levels of feasibility and optimality.

First, the homeomorphic projection consistently achieves a **100%** feasibility rate across all tested convex constraint sets. This underscores the effectiveness of our unsupervised training scheme for MDH mapping, complemented by the bisection algorithm's role in restoring feasibility. The gradient-based D-Proj method decreases constraint violation and improves the feasibility rate by minimizing the penalty function. However, it fails to guaran-

Table 2: Performance comparison in convex constrained optimization: QP and QCQP.

Method		**Feasibility**			**Optimality**				**Speedup**	
NN Predictor	Post Processing	feas. rate % (↑)	ineq. vio. 1-norm (↓)	eq. vio. 1-norm (↓)	solution error ave. % (↓)	solution error cor. % (↓)	objective error ave. % (↓)	objective error cor. % (↓)	ave. × (↑)	cor. × (↑)
		QP: $n = 200,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
Eq-NN-S	–	90.72	0.023	0	1.62	1.67	0.45	0.45	10^5	–
Eq-NN-S	WS	100	0	0	1.46	0	0.41	0	21.1	2
Eq-NN-S	Proj	100	0	0	1.62	1.67	0.45	0.45	49.5	4.6
Eq-NN-S	D-Proj	90.82	0.01	0	1.62	1.66	0.45	0.44	28.9	2.7
Eq-NN-S	H-Proj	100	0	0	1.63	1.84	0.47	0.59	184	17.2
		QP: $n = 200,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
Eq-NN-U	–	49.32	0.016	0	3.55	3.55	0.21	0.21	10^5	–
Eq-NN-U	WS	100	0	0	1.75	0	0.11	0	6.3	3.2
Eq-NN-U	Proj	100	0	0	3.55	3.55	0.22	0.21	20.5	10.4
Eq-NN-U	D-Proj	50.39	0.006	0	3.55	3.54	0.21	0.21	5.1	2.6
Eq-NN-U	H-Proj	100	0	0	3.56	3.57	0.26	0.3	173	19.0
		Convex QCQP: $n = 200,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
Eq-NN-S	–	93.95	0.047	0	4.16	4.36	1.45	1.42	10^6	–
Eq-NN-S	WS	100	0	0	3.9	0	1.37	0	12.8	0.8
Eq-NN-S	Proj	100	0	0	4.16	4.36	1.45	1.43	21.3	1.3
Eq-NN-S	D-Proj	94.14	0.015	0	4.16	4.36	1.45	1.42	805	48.8
Eq-NN-S	H-Proj	100	0	0	4.17	4.52	1.47	1.69	8353	511
		Convex QCQP: $n = 200,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
Eq-NN-U	–	63.96	0.034	0	6.47	6.68	0.54	0.54	10^6	–
Eq-NN-U	WS	100	0	0	4.06	0	0.35	0	3.5	1.3
Eq-NN-U	Proj	100	0	0	6.47	6.68	0.55	0.55	4.2	1.5
Eq-NN-U	D-Proj	91.6	0.004	0	6.47	6.68	0.55	0.54	120	43.2
Eq-NN-U	H-Proj	100	0	0	6.48	6.72	0.63	0.78	7729	2819

[1] n is the dimension of the decision variable, d is the dimension of input parameters, n_{eq} and n_{ineq} are the numbers of equality and inequality constraints, respectively.

[2] All NN-based schemes use reconstruction techniques to ensure the feasibility of equality constraints (Pan et al., 2020; Donti et al., 2020), as discussed in Appendix A.

[3] The ave. metric is calculated based on the average of all predicted solutions, while the cor. metric is evaluated based only on the corrected infeasible solutions.

tee feasibility in all cases, and it exhibits significant sensitivity to step size during parameter tuning in experiments.

Moreover, the homeomorphic projection restores feasibility with minor optimality loss, even in the presence of relatively large prediction errors. As shown in the columns of optimality in the Tables, the primary source of the optimality gap is the initial NN prediction, while our method adds only minor projection errors to locate a feasible solution near the infeasible prediction. This performance is also supported by Theorem 10 and Corollary 11.

Furthermore, the efficient bisection operation within our framework offers a competitive speedup in all tested scenarios. Solver-based methods like WS and Proj, although capable of achieving feasibility with minimal optimality loss, did not provide significant speed improvements due to their reliance on iterative solvers. The D-Proj method, particularly when handling complex constraints such as quadratic constraints in convex QCQP or the positive definite matrix constraints in SDP, suffers from computationally intensive gradient calculations during the gradient descent process.

Table 3: Performance comparison in convex constrained optimization: SOCP and SDP.

Method		Feasibility			Optimality				Speedup	
NN Predictor	Post Processing	feas. rate % (↑)	ineq. vio. 1-norm (↓)	eq. vio. 1-norm (↓)	solution error ave. % (↓)	solution error cor. % (↓)	objective error ave. % (↓)	objective error cor. % (↓)	ave. × (↑)	cor. × (↑)
		SOCP: $n = 200,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
Eq-NN-S	–	88.96	0.192	0	4.8	5.27	1.35	0.99	10^6	–
Eq-NN-S	WS	100	0	0	4.22	0	1.24	0	12.9	1.4
Eq-NN-S	Proj	100	0	0	4.8	5.26	1.37	1.14	13.6	1.5
Eq-NN-S	D-Proj	93.85	0.007	0	4.84	5.56	1.38	1.22	308	34
Eq-NN-S	H-Proj	100	0	0	4.83	5.47	1.41	1.56	6724	749
		SOCP: $n = 200,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
Eq-NN-U	–	75.88	0.268	0	19.56	20.96	3.53	3.36	10^6	–
Eq-NN-U	WS	100	0	0	14.5	0	2.72	0	6.3	1.5
Eq-NN-U	Proj	100	0	0	19.55	20.94	3.58	3.55	5.9	1.4
Eq-NN-U	D-Proj	82.13	0.012	0	19.58	21.07	3.61	3.69	94	22.7
Eq-NN-U	H-Proj	100	0	0	19.56	20.99	3.69	4	6257	827
		SDP: $n = 20 \times 20,\ d = 20,\ n_{\text{eq}} = 20,\ n_{\text{ineq}} = 1$								
Eq-NN-S	–	91.21	0.05	0	12.88	13	12.39	12.09	10^5	–
Eq-NN-S	WS	100	0	0	11.73	0	11.33	0	39.2	3.4
Eq-NN-S	Proj	100	0	0	12.88	12.99	12.39	12.12	27.4	2.4
Eq-NN-S	D-Proj	91.21	0.05	0	12.88	12.99	12.39	12.09	4.6	0.4
Eq-NN-S	H-Proj	100	0	0	12.9	13.28	12.45	12.71	369	32.7
		SDP: $n = 20 \times 20,\ d = 20,\ n_{\text{eq}} = 20,\ n_{\text{ineq}} = 1$								
Eq-NN-U	–	90.92	0.024	0	29.03	28.9	11.99	11.86	10^5	–
Eq-NN-U	WS	100	0	0	26.4	0	10.92	0	35.1	3.2
Eq-NN-U	Proj	100	0	0	29.03	28.89	12	11.88	28.3	2.6
Eq-NN-U	D-Proj	90.92	0.01	0	29.03	28.9	11.99	11.87	4.2	0.4
Eq-NN-U	H-Proj	100	0	0	29.05	29.18	12.06	12.64	464	42.4

[1] The inequality constraint violation for SDP is evaluated by the minimum eigenvalue. In the training stage, we adopt the penalty function in (Needell et al., 2022) for positive-definite matrix constraint.

7.2.2 Non-convex Constrained Optimization

Subsequently, we assess homeomorphic projection on more challenging non-convex NP-hard AC-OPF problems in real-world power grids (Bienstock and Verma, 2019; Babaeinejadsarookolaee et al., 2019). The detailed formulations are presented in Appendix I.1. AC-OPF problems include non-convex quadratic equality constraints, commonly referred to as power flow equations, along with a set of inequality constraints that represent the physical limits of the system. System operators must determine optimal power generation and dispatch strategies to balance real-time demand with minimum costs, where the feasibility of generation schemes is crucial for power grid safety. The performance of different methods for AC-OPF is presented in Table 4. We have the following observations.

First, the homeomorphic projection still achieves **100%** feasibility in non-convex AC-OPF problems across power networks ranging from 30-node to 118-node configurations, demonstrating the robustness of our approach under complex constraint conditions. Moreover, in optimization scenarios with non-convex constraints, such as AC-OPF, using an iterative solver to solve the standard projection problem does not guarantee the identification of the nearest feasible solutions. The non-convex nature of these problems can trap

Table 4: Performance comparison in non-convex AC-OPF problems.

Method		**Feasibility**			**Optimality**				**Speedup**	
NN Predictor	Post Processing	feas. rate % (↑)	ineq. vio. 1-norm (↓)	eq. vio. 1-norm (↓)	solution error ave. % (↓)	solution error cor. % (↓)	objective error ave. % (↓)	objective error cor. % (↓)	ave. × (↑)	cor. × (↑)
		30-node AC-OPF: $n = 72,\ d = 60,\ n_{\text{eq}} = 60,\ n_{\text{ineq}} = 84$								
Eq-NN-S	–	77.73	0.011	0	1.04	1.07	1.22	-0.26	10^5	–
Eq-NN-S	WS	100	0	0	0.8	0	1.27	0	10.7	2.4
Eq-NN-S	Proj	100	0	0	1.23	1.93	1.34	0.29	9.6	2.1
Eq-NN-S	D-Proj	78.61	0.011	0	1.01	0.97	1.21	-0.29	7.4	1.6
Eq-NN-S	H-Proj	100	0	0	1.05	1.15	1.34	0.28	465	108
		30-node AC-OPF: $n = 72,\ d = 60,\ n_{\text{eq}} = 60,\ n_{\text{ineq}} = 84$								
Eq-NN-U	–	79.69	0.011	0	2.7	2.71	1.46	-0.15	10^5	–
Eq-NN-U	WS	100	0	0	2.15	0	1.49	0	10.8	2.2
Eq-NN-U	Proj	100	0	0	2.74	2.92	1.56	0.38	11.3	2.3
Eq-NN-U	D-Proj	79.88	0.011	0	2.67	2.6	1.45	-0.18	7.8	1.6
Eq-NN-U	H-Proj	100	0	0	2.62	2.34	1.55	0.34	596	128
		57-node AC-OPF: $n = 128,\ d = 114,\ n_{\text{eq}} = 114,\ n_{\text{ineq}} = 302$								
Eq-NN-S	–	95.12	0.021	0	2.17	1.49	1.09	0.31	10^4	–
Eq-NN-S	WS	100	0	0	2.1	0	1.08	0	33.4	1.6
Eq-NN-S	Proj	100	0	0	2.15	1.15	1.1	0.37	32.6	1.6
Eq-NN-S	D-Proj	96.48	0.016	0	2.16	1.41	1.09	0.31	35.5	1.7
Eq-NN-S	H-Proj	100	0	0	2.17	1.42	1.1	0.43	730	40.7
		57-node AC-OPF: $n = 128,\ d = 114,\ n_{\text{eq}} = 114,\ n_{\text{ineq}} = 302$								
Eq-NN-U	–	94.53	0.016	0	3.12	2.73	0.64	0.29	10^4	–
Eq-NN-U	WS	100	0	0	2.97	0	0.63	0	32.6	1.8
Eq-NN-U	Proj	100	0	0	3.1	2.42	0.64	0.34	33.4	1.8
Eq-NN-U	D-Proj	95.02	0.012	0	3.11	2.65	0.64	0.29	31.6	1.7
Eq-NN-U	H-Proj	100	0	0	3.12	2.89	0.66	0.66	620	37.5
		118-node AC-OPF: $n = 344,\ d = 236,\ n_{\text{eq}} = 236,\ n_{\text{ineq}} = 452$								
Eq-NN-S	–	94.92	0.002	0	9.05	9.08	0.69	0.59	10^4	–
Eq-NN-S	WS	100	0	0	8.59	0	0.66	0	29	1.5
Eq-NN-S	Proj	100	0	0	9.13	10.75	0.69	0.59	33.1	1.7
Eq-NN-S	D-Proj	95.41	0.002	0	9.05	9.08	0.69	0.59	24.6	1.3
Eq-NN-S	H-Proj	100	0	0	9.36	15.3	0.78	2.44	370	22.9
		118-node AC-OPF: $n = 344,\ d = 236,\ n_{\text{eq}} = 236,\ n_{\text{ineq}} = 452$								
Eq-NN-U	–	92.87	0.008	0	15.18	15.35	1.15	1.18	10^4	–
Eq-NN-U	WS	100	0	0	14.09	0	1.07	0	22.1	1.7
Eq-NN-U	Proj	100	0	0	15.15	14.99	1.15	1.19	21.1	1.5
Eq-NN-U	D-Proj	92.97	0.008	0	15.18	15.37	1.15	1.18	29.4	2.1
Eq-NN-U	H-Proj	100	0	0	15.37	18.03	1.27	2.84	395	33.1

[1] The non-linear power flow equation constraint is ensured by the reconstruction techniques (Donti et al., 2020).

solvers in local optima or stationary points related to the KKT system. However, homeomorphic projection guarantees finding the feasible solution with a bounded optimality gap given valid INN, as supported by Theorem 10 and validated by the columns of Optimality in the Table.

Regarding speed, although we incorporate equality completion to ensure the feasibility of power flow equations (Donti et al., 2020), our method still shows competitive speed compared to commonly adopted solver-based approaches like WS and Proj. This competi-

tive performance highlights the efficiency of our approach, even when dealing with complex constraints and larger network scales.

8. Conclusions and Future Directions

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 non-convex 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 more complex constraint sets.

Acknowledgments and Disclosure of Funding

This work is supported in part by (i) a General Research Fund from Research Grants Council, Hong Kong (Project No. 11203122), (ii) an InnoHK initiative, The Government of the HKSAR, Laboratory for AI-Powered Financial Technologies, (iii) a Shenzhen-Hong Kong-Macau Science & Technology Project (Category C, Project No. SGDX20220530111203026), (iv) Caltech Resnick Sustainability Institute, and (v) Caltech S2I program. Part of this work has been presented at ICML 2023 (Liang et al., 2023). The authors would also like to thank the anonymous reviewers for their helpful comments.

Appendix A. Handling Equality Constraints

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

$$\mathcal{K}_\theta = \{x \in \mathbb{R}^n \mid \mathbf{h}(x,\theta) = 0, g_1(x,\theta) \leq 0, \cdots, g_{n_{\text{ineq}}}(x,\theta) \leq 0\}, \tag{18}$$

where $\mathbf{h}(\cdot,\cdot) : \mathbb{R}^{n+d} \to \mathbb{R}^r$ is continuous with respect to x and θ, and we denote $\mathbf{h}_\theta(\cdot) = \mathbf{h}(\cdot,\theta)$.

Assuming that the equality constraint maintains a constant rank:

$$\text{rank}(\text{J}_{\mathbf{h}_\theta}(x)) = r, \quad \forall \theta \in \Theta \text{ and } \forall x \in \mathcal{K}_\theta, \tag{19}$$

this implies that $\mathcal{K}_\theta$ has a Euclidean dimension[7] of $n - r$, as per the *Constant-Rank Level Set Theorem* (Lee, 2013).

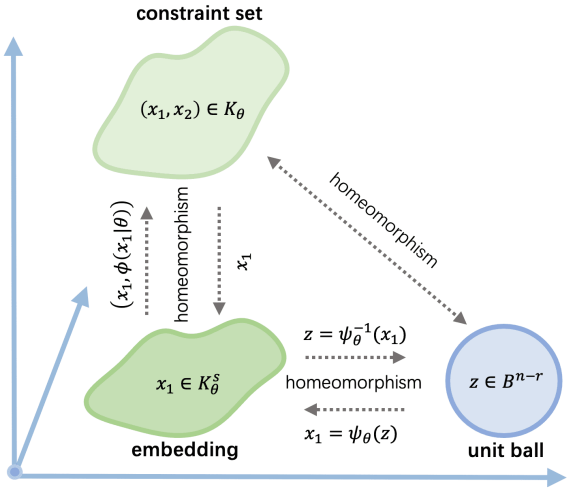

Figure 8: Constraint homeomorphism with equality constraints.

In simpler terms, we can use a subset of decision variables $x_1 \in \mathbb{R}^{n-r}$ and reconstruct the full decision variable set $[x_1, x_2] \in \mathbb{R}^n$ via the equality constraint, as depicted in Fig. 8, where $x_2 = \phi_\theta(x_1)$ and $\mathbf{h}_\theta([x_1, \phi_\theta(x_1)]) = 0$. This process of reconstruction, which ensures the feasibility of the equality constraint, is widely employed in the literature (Pan et al., 2019; Zamzam and Baker, 2020; Donti et al., 2020).

We then denote the equivalent constraint set with equality reconstruction as $\mathcal{K}_\theta^s = \{x \in \mathbb{R}^{n-r} \mid g_1([x_1, \phi_\theta(x_1)], \theta) \leq 0, \cdots, g_{n_{\text{ineq}}}([x_1, \phi_\theta(x_1)], \theta) \leq 0\}$, which is also 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:

$$[x_1, x_2] \in \mathcal{K}_\theta \to x_1 \in \mathcal{K}_\theta^s, \tag{20}$$

$$x_1 \in \mathcal{K}_\theta^s \to [x_1, \phi_\theta(x_1)] \in \mathcal{K}_\theta. \tag{21}$$

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

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

A.1 Linear Equality Constraint

Let's consider an equality constraint defined as $\{x \in \mathbb{R}^n \mid Ax = \theta, A \in \mathbb{R}^{r\times n}, \theta \in \mathbb{R}^r\}$, where x is the decision variable and θ is the input parameter. We can assume, without loss of generality, that the rank of matrix A is $\text{rank}(A) = r$. We then partition the decision variable into two groups: $x_1 \in \mathbb{R}^{n-r}$ and $x_2 \in \mathbb{R}^r$. Correspondingly, matrix A is partitioned as $A = [A_1, A_2]$, with $A_1 \in \mathbb{R}^{r\times(n-r)}$ and $A_2 \in \mathbb{R}^{r\times r}$. Hence, the equality constraint can be represented as $A_1x_1 + A_2x_2 = \theta$.

The reconstruction process essentially implies that we only need a subset of variables, x_1, to ensure the equality constraints, as expressed by:

$$x_2 = \phi_\theta(x_1) = A_2^{-1}(\theta - A_1x_1). \tag{22}$$

There are multiple ways to partition x_1 and x_2 such that A_2 has the full rank of r. Empirically, we select the partition that leads to A_2 having a large determinant. After inverting A_2, x_2 becomes less sensitive to x_1, which is advantageous for alleviating error propagation when training the neural network (NN) predictor and MDH mapping. The Jacobian matrix used for back-propagation is:

$$\mathrm{J}_{\phi_\theta}(x_1) = -A_2^{-1}A_1. \tag{23}$$

A.2 Non-linear Equality Constraint

In the case of a non-linear equality constraint defined as $\{x \in \mathbb{R}^n \mid \mathbf{h}(x, \theta) = 0, \theta \in \mathbb{R}^d, \mathbf{h} : \mathbb{R}^{n+d} \to \mathbb{R}^r\}$, we partition the decision variable into x_1 and x_2 similarly. Under the constant rank assumption, the completion function ϕ_θ exists and is implicitly defined by:

$$\mathbf{h}([x_1, \phi_\theta(x_1)], \theta) = 0. \tag{24}$$

We can use Newton's method to iteratively solve this non-linear equation. The Jacobian matrix required for back-propagation can be derived using the *Implicit Function Theorem*, as follows:

$$\mathrm{J}_{\phi_\theta}(x_1) = -\,\mathrm{J}_{\mathbf{h}_\theta}^{-1}(x_2)\,\mathrm{J}_{\mathbf{h}_\theta}(x_1). \tag{25}$$

In conclusion, reconstruction techniques allow us to exploit the equality constraints and reduce the number of decision variables to predict. As a result, we only need to model the constraint set with inequality constraints with a part of the decision variables. Moreover, this reconstruction process is differentiable, so it can be integrated into the training process. This strategy provides a powerful tool for dealing with equality constraints in machine learning models.

Appendix B. HP for General Constraints

Assumption 1 posits that the constraint set $\mathcal{K}_\theta$ is topologically equivalent to a unit ball. Under this assumption, the constraint set encompasses: (i) all compact convex sets, and (ii) certain classes of compact and simply-connected non-convex sets. We leverage the homeomorphism between these sets to facilitate learning the mapping and conducting projections with reduced computational complexity.

In this study, we select a unit ball as the initial set due to its simplicity in facilitating projection and sampling using quasi-Monte Carlo methods, which are integral to the efficiency of the homeomorphic projection framework.

However, Assumption 1 may not hold for all types of constraints, especially those that are disconnected or manifold-like. To address more general constraint sets, one approach is to identify the corresponding homeomorphism for each specific constraint set, such as transforming a teacup-like set into a torus. Yet, determining the topological properties of a constraint set can be challenging. Alternatively, we can apply ball-homeomorphic projection to learn multiple ball-homeomorphic mappings, such that $\mathcal{K}_\theta = \bigcup_{c=1}^{N} \Phi(\mathcal{B}|\theta, c)$.

For instance, consider the sphere constraint $\mathcal{K}_\theta = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$, which has a constant rank of 1, but it is not homeomorphic to a unit ball. After applying the reconstruction techniques detailed 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, the study of more general constraint sets will form an integral part of future work. The aim will be to extend the relevance and utility of the HP framework to a broader range of optimization problems with complex constraints.

Appendix C. Invertible Neural Network

The Invertible Neural Network (INN) is essentially an invertible NN owing to its design on each invertible layer, such that it is invertible and differentiable in forward and inverse. Thus, it is a homeomorphic mapping with learnable parameters. We introduce several commonly used invertible layers for INN, and refer readers to (Papamakarios et al., 2021) for a more comprehensive introduction.

C.1 Invertible Layers of INN

- **Affine coupling layer** (Dinh et al., 2014)

$$\text{Forward:} \quad [y_1, y_2] = [x_1, \exp(\mathbf{w}(x_1)) \cdot x_2 + \mathbf{b}(x_1)], \tag{26}$$

$$\text{Inverse:} \quad [x_1, x_2] = [y_1, (y_2 - \mathbf{b}(y_1))/\exp(\mathbf{w}(y_1))], \tag{27}$$

  where $x = [x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}]$, $\mathbf{w}(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ and $\mathbf{b}(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ are two regular NNs with learnable parameters, which take x_1 as input and predict the weight and bias respectively 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 either 1 or $\exp(\mathbf{w}(x_1))$. Furthermore, for the conditional invertible layer, we augment the input parameters θ as $\mathbf{w}(x_1, \theta)$ and $\mathbf{b}(x_1, \theta)$, respectively.

  One particular case of the coupling layer is the affine auto-regressive layer (Germain et al., 2015), 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.

- **Bi-Lip affine coupling layer**

$$\text{Forward:} \quad [y_1, y_2] = [x_1, \mathrm{S}(\mathbf{w}(x_1), L) \cdot x_2 + \mathbf{b}(x_1)], \tag{28}$$

$$\text{Inverse:} \quad [x_1, x_2] = [y_1, (y_2 - \mathbf{b}(y_1))/\mathrm{S}(\mathbf{w}(y_1), L)], \tag{29}$$

where the function $S(x, L) := \exp(\log(L) \cdot \tanh(x))$ limits the upper and lower bound of the output weights, such that $S(x, L) \in (1/L, L)$, where $L > 1$ is a predefined parameter. Such an output constraint of weight ensures the bi-Lipschitz property of this layer, which is crucial to bound the distortion.

- **Invertible linear layer** (Kingma and Dhariwal, 2018)

$$\text{Forward:} \quad y = Wx + b, \qquad \text{Inverse:} \quad x = W^{-1}(y - b), \tag{30}$$

where $W \in \mathbb{R}^{n \times n}$ is an invertible and learnable matrix, and $b \in \mathbb{R}^n$ is the learnable bias. Further, by the LU decomposition, the invertible matrix is designed as $W = W_P W_L (W_U + \text{diag(s)})$, where W_P is a fixed permutation matrix, W_L is a lower triangular matrix, W_U is an upper triangular matrix, and $s \in \mathbb{R}^n$ is the diagonal elements. The singular values of the invertible matrix are $|s|$. The invertible linear layer includes Invertible 1×1 convolution and the Actnorm layer (Kingma and Dhariwal, 2018).

- **Invertible residual layer** (Behrmann et al., 2019; Chen et al., 2019)

$$\text{Forward:} \quad y = x + F(x), \qquad \text{Inverse:} \quad x_t \leftarrow y - F(x_{t-1}), \tag{31}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a Lipschitz constrained function where $\text{Lip}(F) < 1$. The inverse process is computed iteratively through a fixed-point iteration scheme. Owning to the Lipschitz constraint, the fixed-point iteration is guaranteed to converge when $t \to \infty$, thus ensuring the invertibility of the residual layer. The log-determinant of this layer can be approximated by the power series (Behrmann et al., 2019). Further, an invertible residual layer with certified strong monotonicity and Lipschitzness has been proposed in (Wang et al., 2024).

- **Neural ODE layer** (Chen et al., 2018; Grathwohl et al., 2018)

$$\text{Forward:} \quad y = x + \int_0^1 F(x, t)\mathrm{d}t, \qquad \text{Inverse:} \quad x = y - \int_1^0 F(x, t)\mathrm{d}t, \tag{32}$$

where $F(\cdot, \cdot) : \mathbb{R}^{n+1} \to \mathbb{R}^n$ represents a time-dependent vector field. The forward and inverse processes are both computed based on integration, ensuring that the system is invertible.

- **Convex potential layer** (Huang et al., 2020)

$$\text{Forward:} \quad y = \nabla F(x) \qquad \text{Inverse:} \quad x = \arg\min_z \{F(z) - y^\top z\}, \tag{33}$$

where $F : \mathbb{R}^n \to \mathbb{R}$ denotes a strongly convex function. The inverse process is computed by iteratively solving the optimization problem. Because of the strictly convex property of F, the solution for the inverse process is unique. Thus, the invertibility of the convex potential layer is guaranteed.

Table 5: Comparison of different INN layers.

INN layer	Free-form structure	Exact forward calc.	Exact inverse calc.	Exact singular val.	$\mathbb{R}^n$ bounded distortion	$\mathcal{H}^n$ universal approximation
Invertible linear layer	✗	✓	✓	✓	σ_1/σ_n	✗
Affine coupling layer	✗	✓	✓	✓	✗	✓
Invertible residual layer	✗	✓	✗	✗	$\frac{1+\text{Lip}(F)}{1-\text{Lip}(F)}$	✓
Neural ODE layer	✓	✗	✗	✗	✗	✓
Convex potential layer	✗	✓	✗	✗	✗	✗
Bi-Lip affine coupling layer	✗	✓	✓	✓	L^2	✓

[1] The $\mathcal{H}^n$ universal approximation of a layer represents that the finite composition of the layer and additional linear layers can approximate any n-dimensional (differentiable) homeomorphic mappings arbitrarily well given enough number of layers (Teshima et al., 2020; Lyu et al., 2022; Zhang et al., 2020; Kong and Chaudhuri, 2021; Ishikawa et al., 2022).

C.2 Computational Issues and Approximation Ability of INN

For our *Homeomorphic Projection* (HP) framework, there are several requirements for the Invertible Neural Network (INN): (i) The inverse needs to be computed in the bisection algorithm as outlined in Alg. 1. This requirement ensures the correct functioning of the algorithm. (ii) The singular values of the Jacobian matrix of the INN need to be calculated. This is crucial for the efficient training of the INN as per Equation (11). (iii) The INN should have bounded distortion to ensure the worst-case performance for homeomorphic projection. (iv) The INN should be a universal approximator of homeomorphic mappings. This enables it to handle complex transformations involving a broad range of constraints.

Based on the summary of INNs in Table 5, we select the coupling layer as a building block of INN in our HP framework.

Appendix D. Proof for Propositions

D.1 Proof for Proposition 7

Proof We first prove the universal approximation ability of the INN Φ with bi-Lipschitz affine coupling layers and invertible linear layers over compact sets with the following steps: (i) the finite composition of bi-Lipschitz affine coupling layers can represent any affine coupling layer over a compact set, (ii) the composition of affine coupling layers and invertible linear layers is the universal approximation of any differentiable homeomorphism, also known as a diffeomorphism.

For the first statement, consider an affine coupling layer $f_a : \mathbb{R}^n \to \mathbb{R}^n$ defined as:

$$[y_1, y_2] = [x_1, \exp(\mathbf{w}_a(x_1)) \cdot x_2 + \mathbf{b}_a(x_1)], \tag{34}$$

where $x = [x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}] \in \mathcal{X}$ and $\mathcal{X}$ is a comapct set. $\mathbf{w}_a(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ and $\mathbf{b}_a(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ are continuous functions.

As the output of a continuous function over a compact set is bounded, the weights for the affine transformation, i.e., $\exp(\mathbf{w}_a(x_1))$, are also bounded, and there exists a constant $c > 1$ such that $1/c \leq \exp(\mathbf{w}_a(x_1)) \leq c$.

On the other hand, the bi-Lip affine coupling layer $f_b : \mathbb{R}^n \to \mathbb{R}^n$ is defined as:

$$[y_1, y_2] = [x_1, \exp(\log(L) \cdot \tanh(\mathbf{w}_b(x_1))) \cdot x_2 + \mathbf{b}_b(x_1)], \tag{35}$$

where $\mathbf{w}_b$ and $\mathbf{b}_b$ are also continuous function. The weights for the affine transformation are within the interval $(1/L, L)$.

Therefore, to represent an affine coupling layer over the compact set, we need $\lceil c/L \rceil$ bi-Lip affine coupling layers, where the weight for each bi-Lip affine coupling layer is given by $\exp(\log(L) \cdot \tanh(\mathbf{w}_b(x_1))) = \exp(\mathbf{w}_a(x_1))^{1/\lceil c/L \rceil}$, and the bias is zero except for the last layer in the composition as $\mathbf{b}_b(x_1) = \mathbf{b}_a(x_1)$.

For the second statement, we leverage the existing results on the universal approximation ability for affine coupling layers based INN by Theorem 2 in (Teshima et al., 2020). The number of INN layers has an order of $\mathcal{O}(n)$, which is given by: (i) the near-identity diffeomorphism representation of any diffeomorphism over a compact set by Proposition 2 in (Teshima et al., 2020), (ii) n single-coordinate diffeomorphism representations of any near-identity diffeomorphism by Proposition 3 in (Teshima et al., 2020), and (iii) the affine coupling layer and linear layer representation of any single-coordinate diffeomorphism by Lemma 13 and 14 in (Teshima et al., 2020).

Let c_1 be the smallest constant bounding the weights within interval $(1/c_1, c_1)$ for each affine coupling layer in the INN constructed in (Teshima et al., 2020). We then substitute them with $\lceil c_1/L \rceil$ bi-Lip affine coupling layers. Finally, we need $\mathcal{O}(n \cdot \lceil c_1/L \rceil)$-layer INN, composed of bi-Lip affine coupling layers and invertible linear layers, to represent any diffeomorphism over the compact set.

Second, we prove the upper bound of distortion over $\mathbb{R}^n$. Let's consider a m-layer INN, denoted as $\Phi = \Phi^m \circ \cdots \circ \Phi^l \circ \cdots \circ \Phi^1$, where Φ^l is an invertible linear layer for $l = 1, 3, \cdots, m-1$ and a bi-Lipschitz affine coupling layer with a hyper-parameter $L > 1$ for $l = 2, 4, \cdots, m$.

The distortion for each bi-Lipschitz affine coupling layer can be bounded by L^2. The distortion of those linear layers is given by the condition number of the invertible matrix, which is input-invariant. Therefore, the distortion for Φ can be bounded as:

$$\mathrm{D}(\Phi, \mathbb{R}^n) \overset{(a)}{\leq} \prod_{l=1}^{m} \mathrm{D}(\Phi^l, \mathbb{R}^n) \overset{(b)}{\leq} \prod_{l=1,3,\cdots,m-1} \frac{\sigma_1(W^l)}{\sigma_n(W^l)} (L^2)^{m/2} = c_2 L^m, \tag{36}$$

where W^l is the invertible weight matrix for the linear layer, $\sigma_1(W^l)$ and $\sigma_n(W^l)$ denote the largest and smallest singular values of the matrix W^l, and c_2 is a constant. Inequality (a) is by item 4 in Lemma 15, and inequality (b) is derived from the singular value representation of the distortion by item 2 in Lemma 15.

Therefore, the Bi-Lipschitz INN has a bounded distribution over $\mathbb{R}^n$.

■

D.2 Proof for Proposition 8

Proof 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, e.g.,

$\mathrm{V}(\mathcal{B}) = \int_{\mathcal{B}} 1\mathrm{d}z$. Because of the containment constraint $\psi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$, the volumes of two sets satisfy $\mathrm{V}(\psi_\theta(\mathcal{B})) \leq \mathrm{V}(\mathcal{K}_\theta)$. 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 when $\psi_\theta(\mathcal{B}) = \mathcal{K}_\theta$, which is the homeomorphism constraint. ■

D.3 Proof for Volume estimation in Proposition 9

Proof Let's denote an m-layer INN as $\Phi_\theta = \Phi_\theta^m \circ \cdots \circ \Phi_\theta^l \circ \cdots \circ \Phi_\theta^1$ and 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 log-volume term $\log \mathrm{V}(\Phi_\theta(\mathcal{B}))$, we first apply the Jensen inequality for $\log \frac{\mathrm{V}(\Phi_\theta(\mathcal{B}))}{\mathrm{V}(\mathcal{B})}$:

$$\log \frac{\mathrm{V}(\Phi_\theta(\mathcal{B}))}{\mathrm{V}(\mathcal{B})} = \log \frac{\int_{\mathcal{B}} |\det \mathrm{J}_{\Phi_\theta}(z)| \mathrm{d}z}{\mathrm{V}(\mathcal{B})} \tag{37}$$

$$= \log \mathbb{E}_{z\sim p(z)}[|\det \mathrm{J}_{\Phi_\theta}(z)|] \tag{38}$$

$$\geq \mathbb{E}_{z\sim p(z)}[\log |\det \mathrm{J}_{\Phi_\theta}(z)|]. \tag{39}$$

Next, the log-determinant term can be expressed by the singular values as:

$$\log |\det \mathrm{J}_{\Phi_\theta}(z)| = \sum_{k=1}^{n} \sum_{l=1}^{m} \log \sigma_k(\mathrm{J}_{\Phi_\theta^l}(z^l)), \tag{40}$$

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

Thus, a lower bound of the log-volume $\log \mathrm{V}(\Phi_\theta(\mathcal{B}))$ is as follows:

$$\log \mathrm{V}(\Phi_\theta(\mathcal{B})) \geq \widehat{\mathrm{V}}(\Phi_\theta(B)) = \frac{1}{\mathrm{V}(\mathcal{B})} \int_{\mathcal{B}} \sum_{k=1}^{n} \sum_{l=1}^{m} \log \sigma_k(\mathrm{J}_{\Phi_\theta^l}(z^l)) \mathrm{d}z + \log \mathrm{V}(\mathcal{B}). \tag{41}$$

The gap for the lower bound is zero when the determinant is invariant, i.e., $\forall z \in \mathcal{B}$, $|\det \mathrm{J}_{\Phi_\theta}(z)| = \frac{\mathrm{V}(\Phi_\theta(\mathcal{B}))}{\mathrm{V}(\mathcal{B})}$. We use this lower bound because of numerical stability and the closed-form expression of singular values for INN, as shown in Appendix C.

Further, we find a connection between the gap of the lower bound and the distortion, which implies that minimizing the distortion also helps reduce the gap of volume estimation.

Let's apply a sharpened version of Jensen's inequality (Liao and Berg, 2018):

$$\frac{\mathrm{Var}(v(z))}{2v_{\max}^2} \leq \log \frac{\mathrm{V}(\Phi_\theta(\mathcal{B}))}{\mathrm{V}(\mathcal{B})} - \mathbb{E}_{z\sim p(z)}[\log |\det \mathrm{J}_{\Phi_\theta}(z)|] \leq \frac{\mathrm{Var}(v(z))}{2v_{\min}^2}, \tag{42}$$

where $v(z) = |\det(\mathrm{J}_{\Phi_\theta}(z))|$ and $v_{\min} \leq v(z) \leq v_{\max}, \forall z \in \mathcal{B}$.

By applying Popoviciu's inequality on variances (Sharma et al., 2010), we have the upper bound of the gap:

$$\frac{\mathrm{Var}(v(z))}{2v_{\min}^2} \leq \frac{(v_{\max} - v_{\min})^2}{8v_{\min}^2} = \frac{(v_{\max}/v_{\min} - 1)^2}{8} \tag{43}$$

Consider the singular-value representation of determinant $v(z) = |\det(\mathrm{J}_{\Phi_\theta}(z))| = \prod_{i=1}^{n} \sigma_i(\mathrm{J}_{\Phi_\theta}(z))$, we can bound the maximum and minimum determinant as:

$$v_{\max} \leq (\sup_{z\in\mathcal{B}}\{\sigma_1(\mathrm{J}_{\Phi_\theta}(z))\})^n, \qquad v_{\min} \geq (\inf_{z\in\mathcal{B}}\{\sigma_n(\mathrm{J}_{\Phi_\theta}(z))\})^n \tag{44}$$

As a result, we have $v_{\max}/v_{\min} \leq (\frac{\sup_{z\in\mathcal{B}}\{\sigma_1(\mathrm{J}_{\Phi_\theta}(z))\}}{\inf_{z\in\mathcal{B}}\{\sigma_n(\mathrm{J}_{\Phi_\theta}(z))\}})^n \overset{(a)}{=} \mathrm{D}(\Phi_\theta, \mathcal{B})^n$, where the equality (a) is according to the second item in Lemma 15.

Therefore, the gap for volume estimation is bounded as:

$$\log \mathrm{V}(\Phi_\theta(\mathcal{B})) - \widehat{\mathrm{V}}(\Phi_\theta(B)) \leq \frac{\mathrm{Var}(v(z))}{2v_{\min}^2} \leq \frac{1}{8}(\mathrm{D}(\Phi_\theta, \mathcal{B})^n - 1)^2 \tag{45}$$

■

D.4 Proof for Distortion estimation in Proposition 9

Proof Let's denote an m-layer INN as $\Phi_\theta = \Phi_\theta^m \circ \cdots \circ \Phi_\theta^l \circ \cdots \circ \Phi_\theta^1$. According to the first item in Lemma 15, the distortion can be expressed as the product of the Lipschitz constants as:

$$\log \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta) = \log \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta) + \log \mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{X}_\theta) \tag{46}$$

$$\overset{(a)}{=} \sup_{z\in\mathcal{Z}_\theta}\{\log \sigma_1(\mathrm{J}_{\Phi_\theta}(z))\} + \sup_{x\in\mathcal{X}_\theta}\{\log \sigma_1(\mathrm{J}_{\Phi_\theta^{-1}}(x))\} \tag{47}$$

$$\overset{(b)}{\leq} \sup_{z^1\in\mathcal{Z}_\theta}\{\log \prod_{l=1}^{m} \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} + \sup_{x^1\in\mathcal{X}_\theta}\{\log \prod_{l=1}^{m} \sigma_1(\mathrm{J}_{\Phi_\theta^{-1,m-l+1}}(x^l))\} \tag{48}$$

$$\overset{(c)}{=} \sup_{z^1\in\mathcal{Z}_\theta}\{\log \prod_{l=1}^{m} \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} - \inf_{z^1\in\mathcal{Z}_\theta}\{\log \prod_{l=1}^{m} \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))\} \tag{49}$$

$$= \sup_{z^1\in\mathcal{Z}_\theta}\{\sum_{l=1}^{m} \log \sigma_1(\mathrm{J}_{\Phi_\theta^l}(z^l))\} - \inf_{z^1\in\mathcal{Z}_\theta}\{\sum_{l=1}^{m} \log \sigma_n(\mathrm{J}_{\Phi_\theta^l}(z^l))\} \tag{50}$$

$$= \widehat{\mathrm{D}}(\Phi_\theta, \mathcal{Z}_\theta) = \widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta) \tag{51}$$

where $z^l = \Phi_\theta^{l-1}(z^{l-1})$ for $l = 2, \cdots, m+1$, the initial value is $z^1 \in \mathcal{Z}_\theta$ and the final value $z^{m+1} = \Phi_\theta(z^1)$. Similarly, $x^l = \Phi_\theta^{-1,m-l+1}(x^{l-1})$ for $l = 2, \cdots, m+1$, the initial value is $x^1 \in \mathcal{X}_\theta$ and the final value $x^{m+1} = \Phi_\theta^{-1}(x^1)$.

The equality (a) employs the spectral norm, which is the largest singular value, to represent the Lipschitz constant. The inequality (b) is derived from the property of the spectral norm of the matrix, as expressed by $\|AB\| \leq \|A\|\|B\|$. This suggests that the largest singular values for the Jacobian matrix of a composite function are restricted by the product of the largest singular values for the Jacobian matrix of each layer. The equality (c) is due to the property of the invertible matrix, expressed as $\sigma_1(\mathrm{J}) = 1/\sigma_n(\mathrm{J}^{-1})$. Lastly, we denote the approximation of the log-distortion term as $\widehat{\mathrm{D}}(\Phi_\theta^{-1}, \mathcal{X}_\theta)$.

For the practical implementation of distortion estimation, we adopt the average in place of the sup and inf operators. In essence, this approximation estimates the average local distortion at each point over each layer of the network. By adopting such an average-case distortion, we simplify the original worst-case formulation, thereby stabilizing the unsupervised training process of the INN. This approach has proven efficient in related Lipschitz regularization schemes, as demonstrated in several studies (Virmaux and Scaman, 2018; Behrmann et al., 2019, 2021).

We choose singular values to represent distortion due to their closed-form expression in certain INN structures, such as the coupling layer. For other INN structures, like i-ResFlow (Behrmann et al., 2019), we can utilize commonly used finite difference estimation techniques to evaluate the distortion (Schmidt et al., 2019; Behrmann et al., 2021; Liu et al., 2022). ■

Appendix E. Proof for Theorem 10

Proof

E.1 Feasibility

According to the Definition 6 of Valid INN, it maps the origin to a feasible point in $\mathcal{K}_\theta$ as $\Phi_\theta(0) \in \mathcal{K}_\theta$. Applying the inverse homeomorphic mapping to both sides, we have $0 \in \Phi_\theta^{-1}(\mathcal{K}_\theta)$.

The bisection process in Alg. 1 aims 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:

$$\alpha^* = \sup_{\alpha \in [0,1]} \{\Phi_\theta(\alpha \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta\}, \tag{52}$$

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

Therefore, the bisection algorithm is guaranteed to return a feasible solution due to the initial feasible point, i.e., $\Phi_\theta(0) \in \mathcal{K}_\theta$. It then searches for better feasible points, moving from the interior to the constraint boundary, as illustrated in Fig. 3. When $k \to \infty$, the returned feasible solution will be located on the constraint boundary.

Next, we will give a lower bound for the optimal α^*. Considering those points over the constraint boundary in homeomorphic space, we have:

$$0 \le r + \inf_{x \in \partial\mathcal{K}_\theta} \{\|\Phi_\theta^{-1}(x)\| - r\} \le \|z\| \le r + \sup_{x \in \partial\mathcal{K}_\theta} \{\|\Phi_\theta^{-1}(x)\| - r\}, \quad \forall z \in \Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \tag{53}$$

where r is the radius for the ball (we use $r = 1$ by default). For convenience, we denote $\delta_{\text{inn}}^- = \inf_{x \in \partial\mathcal{K}_\theta}\{\|\Phi_\theta^{-1}(x)\| - r\}$ and $\delta_{\text{inn}}^+ = \sup_{x \in \partial\mathcal{K}_\theta}\{\|\Phi_\theta^{-1}(x)\| - r\}$.

We then consider the one-side Hausdorff distance between $\partial\mathcal{B}$ and $\Phi_\theta^{-1}(\partial\mathcal{K}_\theta)$, denoted as $\delta_{\text{inn}} = \mathrm{d_h}(\Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \partial\mathcal{B})$. According to the third item in Lemma 16, we have $\mathrm{d_h}(\Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \partial\mathcal{B}) \ge \sup_{x \in \partial\mathcal{K}_\theta}\{|\|\Phi_\theta^{-1}(x)\|_2 - r|\}$, which implies:

$$\delta_{\text{inn}} = \mathrm{d_h}(\Phi_\theta^{-1}(\partial\mathcal{K}_\theta), \partial\mathcal{B}) \ge \max\{|\delta_{\text{inn}}^+|, |\delta_{\text{inn}}^-|\}. \tag{54}$$

Since the bisection process can be view as "projecting" the infeasible point $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$ to constraint boundary in homeomorphic space as $\hat{z}_\theta = \alpha^* \cdot \tilde{z}_\theta \in \Phi_\theta^{-1}(\partial\mathcal{K}_\theta)$, we can bound $\|\alpha^* \cdot \tilde{z}_\theta\|$ according to the inequality in (53):

$$0 \leq r + \delta_{\text{inn}}^- \leq \|\alpha^* \tilde{z}_\theta\| \leq r + \delta_{\text{inn}}^+ \quad \rightarrow \quad 0 \leq \frac{r + \delta_{\text{inn}}^-}{\|\tilde{z}_\theta\|} \leq \alpha^* \leq \frac{r + \delta_{\text{inn}}^+}{\|\tilde{z}_\theta\|}. \tag{55}$$

Therefore, we have the lower bound for α^*, denoted as $\alpha_l^* = \frac{r+\delta_{\text{inn}}^-}{\|\tilde{z}_\theta\|}$. Based on the understanding of it, we can next analyze the worst-case optimality loss under the bisection algorithm.

E.2 Optimality

Let $\hat{x}_\theta^k$ be the feasible solution after the k-step bisection in Alg. 1 and x_θ^* is the optimal solution for problem in (1). To analyze the optimality loss of returned feasible solution $\hat{x}_\theta^k$, we decompose the loss by following terms:

$$\|\hat{x}_\theta^k - x_\theta^*\| \leq \underbrace{\|\hat{x}_\theta^k - \hat{x}_\theta\|}_{\text{bisection error}} + \underbrace{\|\hat{x}_\theta - \tilde{x}_\theta\|}_{\text{projection error}} + \underbrace{\|\tilde{x}_\theta - x_\theta^*\|}_{\text{prediction error}}, \tag{56}$$

where $\hat{x}_\theta$ is the feasible solution by solving $\hat{x}_\theta = \Phi_\theta(\alpha^* \cdot \Phi_\theta^{-1}(\tilde{x}_\theta))$, and $\tilde{x}_\theta$ is the infeasible predicted solution with bounded prediction error ϵ_{pre}.

Next, we will bound the three error terms.

E.2.1 Prediction error

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

$$\|\tilde{x}_\theta - x_\theta^*\| \leq \epsilon_{\text{pre}}. \tag{57}$$

For convenience, we denote the region for all possible infeasible predictions as $\mathcal{X}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\text{pre}})$, such that $\tilde{x}_\theta \in \mathcal{X}_\theta$.

E.2.2 Projection error

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

First, the homeomorphic projection error can then be expressed as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| = \|\Phi_\theta(\hat{z}_\theta) - \Phi_\theta(\tilde{z}_\theta)\|, \tag{58}$$

where $\hat{x}_\theta, \tilde{x}_\theta \in \mathcal{X}_\theta$ and $\hat{z}_\theta, \tilde{z}_\theta \in \mathcal{Z}_\theta = \Phi_\theta^{-1}(\mathcal{X}_\theta)$.

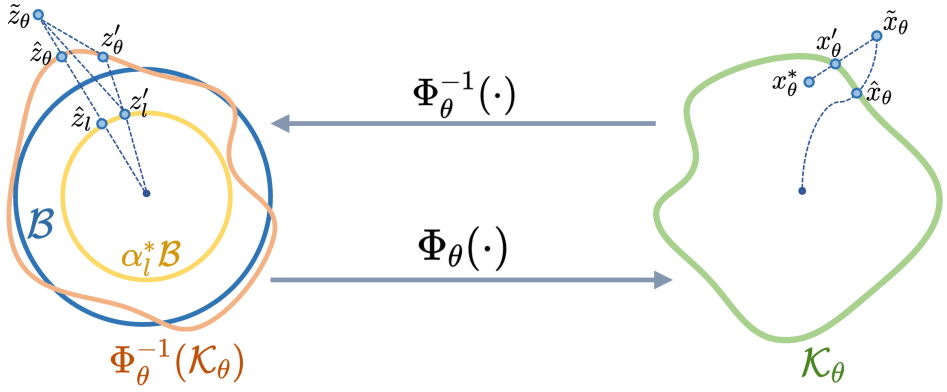

Figure 9: Illustration of Proof for projection error.

Next, based on the Lipschitz constant of mapping Φ_θ over $\mathcal{Z}_\theta$, we can bound the error as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\hat{z}_\theta - \tilde{z}_\theta\| \tag{59}$$

$$\overset{(a)}{=} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha^*\tilde{z}_\theta - \tilde{z}_\theta\| \tag{60}$$

$$\overset{(b)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^*\tilde{z}_\theta - \tilde{z}_\theta\| \tag{61}$$

where equality (a) is according to $\hat{x}_\theta = \Phi_\theta(\hat{z}_\theta) = \Phi_\theta(\alpha^* \cdot \Phi_\theta^{-1}(\tilde{x}_\theta)) = \Phi_\theta(\alpha^* \cdot \tilde{z}_\theta)$, such that $\hat{z}_\theta = \alpha^*\tilde{z}_\theta$. Inequality (b) is due to the lower bound of α^* as shown in (55) and we denote $\hat{z}_l = \alpha_l^*\tilde{z}_\theta$

To proceed, let an auxiliary point as $z'_\theta = \Phi_\theta^{-1}(x'_\theta)$, where $x'_\theta = \lambda x^*_\theta + (1-\lambda)\tilde{x}_\theta \in \partial\mathcal{K}_\theta$ for $\lambda \in [0,1]$.

According to Lemma 17, we can bound the distance as $\|\tilde{z}_\theta - \hat{z}_l\| \leq \|\tilde{z}_\theta - z'_l\|$, where $z'_l = \alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z'_\theta\|} z'_\theta$ as shown in Fig. 9.

Thus, the projection error can be bounded as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z'_\theta\|} z'_\theta - \tilde{z}_\theta\| \tag{62}$$

$$\overset{(a)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)(\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z'_\theta\|} z'_\theta - z'_\theta\| + \|z'_\theta - \tilde{z}_\theta\|) \tag{63}$$

$$\overset{(b)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z'_\theta\|} z'_\theta - z'_\theta\| + \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{X}_\theta)(\|x'_\theta - \tilde{x}_\theta\|) \tag{64}$$

$$\overset{(c)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha_l^* \frac{\|\tilde{z}_\theta\|}{\|z'_\theta\|} z'_\theta - z'_\theta\| + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} \tag{65}$$

where the inequality (a) is by the triangle inequality. The inequality (b) is based on the Lipschitz constant of Φ_θ^{-1} over $\mathcal{X}_\theta$. For the inequality (c), we express the product of Lipschitz constants as the distortion term $\mathrm{D}(\Phi^\theta, \mathcal{Z}_\theta)$ according to the first item in Lemma 15 and bound the distance $\|x'_\theta - \tilde{x}_\theta\|$ by the prediction error ϵ_{pre}.

Next, the projection error can be bounded as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)|\alpha_l^*\|\tilde{z}_\theta\| - \|z'_\theta\|| \cdot \|\frac{z'_\theta}{\|z'_\theta\|}\| + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{66}$$

$$\overset{(a)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)|r + \delta_{\text{inn}}^- - \|z'_\theta\|| + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{67}$$

$$\overset{(b)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)(|r - \|z'_\theta\|| + |\delta_{\text{inn}}^-|) + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{68}$$

where the inequality (a) is by $\|\frac{z'_\theta}{\|z'_\theta\|}\| \leq 1$ and $\alpha_l^* = \frac{r+\delta_{\text{inn}}^-}{\|\tilde{z}_\theta\|}$ as shown in (55). The inequality (b) is by the triangle inequality.

According to the inequality in (54), we have $|r - \|z'_\theta\|| \leq \delta_{\text{inn}}$ and $|\delta_{\text{inn}}^-| \leq \delta_{\text{inn}}$. Thus, the projection error can be bounded as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq 2\mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\delta_{\text{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{69}$$

$$\overset{(a)}{\leq} 2\mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\text{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{70}$$

$$\overset{(b)}{\leq} 2\mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\text{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{71}$$

where the inequality (a) is by the second item in Lemma 16 such that $\delta_{\text{inn}} \leq \mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\text{inn}}$. The inequality (b) is by the third item in Lemma 15.

Let $\mathcal{Y}_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pre}}, \epsilon_{\text{inn}}\})$ such that $\mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}) \subseteq \mathcal{Y}_\theta$. We can bound the projection error as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq 2\mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)\epsilon_{\text{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\text{pre}} \tag{72}$$

Since $\mathcal{K}_\theta \subseteq \mathcal{X}_\theta \subseteq \mathcal{Y}_\theta$ and $\mathcal{Z}_\theta \subseteq \Phi^{-1}(\mathcal{Y}_\theta)$, based on the first and third items in Lemma 15, we can bound the error as:

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pre}}) \tag{73}$$

E.2.3 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 Alg. 1. We set the maximum bisection steps as k, and let the returned solution as $\hat{z}_\theta^k = \alpha^k \tilde{z}_\theta$, where α^k is the k-th iteration of bisection for $\alpha \in [0, 1]$ such that $|\alpha^k - \alpha^*| \leq 2^{-k}$.

Therefore, the bisection error can be bounded as:

$$\|\hat{x}_\theta^k - \hat{x}_\theta\| \overset{(a)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\hat{z}_\theta^k - \hat{z}_\theta\| \tag{74}$$

$$= \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\|\alpha^k \tilde{z}_\theta - \alpha^* \tilde{z}_\theta\| \tag{75}$$

$$= \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)|\alpha^k - \alpha^*|\|\tilde{z}_\theta - 0\| \tag{76}$$

$$\leq \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)2^{-k}\|\tilde{z}_\theta - 0\| \tag{77}$$

where $\hat{x}_\theta^k = \Phi_\theta(\hat{z}_\theta^k)$ and $\hat{x}_\theta = \Phi_\theta(\hat{z}_\theta)$. Since $\hat{x}_\theta^k, \hat{x}_\theta \in \mathcal{K}_\theta$ are feasible solutions, then $\hat{z}_\theta^k, \hat{z}_\theta \in \Phi_\theta^{-1}(\mathcal{K}_\theta) \subseteq \mathcal{Z}_\theta$ and the inequality (a) is based on the Lipschitz constant of Φ_θ over $\mathcal{Z}_\theta$.

Next, we consider mapping the point back to the original space as $\tilde{x}_\theta = \Phi_\theta(\tilde{z}_\theta)$ and bound the error as:

$$\|\hat{x}_\theta^k - \hat{x}_\theta\| \overset{(a)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\frac{1}{2^k}\|\tilde{x}_\theta - \Phi_\theta(0)\| \tag{78}$$

$$\overset{(b)}{\leq} \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\frac{1}{2^k}(\|\tilde{x}_\theta - x_\theta^*\| + \|x_\theta^* - \Phi_\theta(0)\|) \tag{79}$$

$$\overset{(c)}{\leq} \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}_\theta)\frac{1}{2^k}(\epsilon_{\mathrm{pre}} + \mathrm{diam}(\mathcal{K}_\theta)) \tag{80}$$

$$\overset{(d)}{\leq} \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)\frac{1}{2^k}(\epsilon_{\mathrm{pre}} + \mathrm{diam}(\mathcal{K}_\theta)). \tag{81}$$

where $\tilde{x}_\theta \in \mathcal{X}_\theta$ is the infeasible predictions and $\Phi_\theta(0) \in \mathcal{K}_\theta \subseteq \mathcal{X}_\theta$ is owning to the valid INN condition. The inequality (a) is based on the Lipschitz constant of Φ_θ^{-1} over $\mathcal{X}_\theta$. The inequality (b) is based on the first item in Lemma 15 and applies the triangle inequality using an auxiliary point x_θ^*. The inequality (c) bounds $\|\tilde{x}_\theta - x_\theta^*\|$ by the definition of prediction error and bounds $\|x_\theta^* - \Phi_\theta(0)\|$ by the diameter of constraint set $\mathcal{K}_\theta$. The inequality (d) is owning to $\mathcal{X}_\theta \subseteq \mathcal{Y}_\theta$ and the third item in Lemma 15.

E.2.4 Final results for optimality loss

In summary, combining the prediction error, projection error, and bisection error, we have the results:

$$\|\hat{x}_\theta^k - x_\theta^*\| \leq \epsilon_{\mathrm{pre}} + \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}_\theta)(2\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pre}} + \epsilon_{\mathrm{bis}}^k), \tag{82}$$

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

We remark that, in practice, the bisection algorithm may not exactly solve $\alpha^* = \sup_{\alpha\in[0,1]}\{\Phi_\theta(\alpha^* \cdot \tilde{z}_\theta) \in \mathcal{K}_\theta\}$ due to potential multiple intersections of the bisection trajectory with the constraint boundary, resulting in one of several boundary feasible solutions. Nevertheless, the upper bound above on the optimality gap remains valid as we consider the worst-case projected solution (Fig. 9), allowing the projection distance of any boundary feasible solution in the bisection trajectory to be bounded by it.

E.3 Run-time Complexity

The run-time complexity for Alg. 1 involves, (i) inverse calculation of INN for the infeasible predictions as $\tilde{z}_\theta = \Phi_\theta^{-1}(\tilde{x}_\theta)$, (ii) k-step bisection, which includes one forward INN calculation $\hat{x}_n = \Phi_\theta(\hat{z}_n)$ and one verification for the feasibility of $\Phi_\theta(\hat{z}_n)$.

Consider an m-block INN, where each invertible block is a composition of a bi-Lip affine coupling layer and an invertible linear layer. Based on the definitions of the two layers in Appendix C, both forward and inverse computation involve a complexity of $\mathcal{O}(n^2)$. Thus, the run-time complexity for INN is $\mathcal{O}(mn^2)$

For the verification of feasibility, we calculate $\mathbf{g}(\hat{x}, \theta) = [g_i(\hat{x}, \theta), ..., g_{n_{\mathrm{ineq}}}(\hat{x}, \theta)]$ to find if it is a feasible solution. We denote the complexity for calculating $\mathbf{g}(\hat{x}, \theta)$ as G. For instance, $G = n_{\mathrm{ineq}} \cdot n$ for linear constraints, $G = n_{\mathrm{ineq}} \cdot n^2$ for quadratic constraints, and $G = n_{\mathrm{ineq}} \cdot \mathcal{O}(n^{1.5})$ for positive definite matrix constraints

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

■

Table 6: Projection error under different settings.

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

[1] Here $\mathcal{Y}'_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pro}}, \epsilon_{\text{inn}}\})$ and $\mathcal{X}'_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \epsilon_{\text{pro}})$.

Appendix F. Proof for Corollary 11

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 6.

F.1 Valid INN mapping

Proof Following the proof in (62), we select the auxiliary point x'_θ as the point by standard projection, i.e., $x'_\theta = \hat{x}^*_\theta \in \mathrm{Proj}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \arg\min_{y \in \mathcal{K}_\theta}\{\|\tilde{x}_\theta - y\|\}$ 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: $\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}'_\theta)(2\epsilon_{\text{inn}} + \epsilon_{\text{pro}})$, where $\mathcal{Y}'_\theta = \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\text{pro}}, \epsilon_{\text{inn}}\})$. ■

F.2 Valid INN and Outer Approximation

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.

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{Y}'_\theta)(\epsilon_{\text{inn}} + \epsilon_{\text{pro}}). \tag{83}$$

Proof

Since $\Phi_\theta(\mathcal{B}) \supseteq \mathcal{K}_\theta$, after homeomorphic mapping $\mathcal{B} \supseteq \Phi_\theta^{-1}(\mathcal{K}_\theta)$, we have $\delta^-_{\text{inn}} \leq \delta^+_{\text{inn}} \leq 0$ and $\delta_{\text{inn}} = |\delta^-_{\text{inn}}|$.

Following the inequalities from (67) to (73), we have:

$$\begin{aligned}
\|\hat{x}_\theta - \tilde{x}_\theta\| &\le \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)(|r + \delta^-_{\mathrm{inn}} - \|z^b_\theta\|\|) + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (84)\\
&\overset{(a)}{\le} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)(|\delta^+_{\mathrm{inn}} - \delta^-_{\mathrm{inn}}|) + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (85)\\
&\le \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\delta_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (86)\\
&\le \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi^{-1}_\theta, \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (87)\\
&\overset{(b)}{\le} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi^{-1}_\theta, \mathcal{K}_\theta + \mathcal{B}(0, \max\{\epsilon_{\mathrm{pre}}, \epsilon_{\mathrm{inn}}\}))\epsilon_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (88)\\
&\le \mathrm{D}(\Phi^{-1}_\theta, \mathcal{Y}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pre}}), && (89)
\end{aligned}$$

where the inequality (a) is by $\delta^-_{\mathrm{inn}} \le \|z'_\theta\| - r \le \delta^+_{\mathrm{inn}}$ according to inequalities (53) and applying the triangle inequality. The inequality (b) is owning to $\mathcal{K}_\theta \subseteq \Phi_\theta(\mathcal{B}) \subseteq \mathcal{Y}_\theta$.

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: $\|\hat{x}_\theta - \tilde{x}_\theta\| \le \mathrm{D}(\Phi^{-1}_\theta, \mathcal{Y}'_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pro}})$.

■

F.3 Valid INN and 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 $\Phi_\theta(0) \in \mathcal{K}_\theta$.

$$\|\hat{x}_\theta - \tilde{x}_\theta\| \le \mathrm{D}(\Phi^{-1}_\theta, \mathcal{X}'_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pro}}). \tag{90}$$

Proof

Since $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta$, after homeomorphic mapping $\mathcal{B} \subseteq \Phi^{-1}_\theta(\mathcal{K}_\theta)$, we have $0 \le \delta^-_{\mathrm{inn}} \le \delta^+_{\mathrm{inn}} = \delta_{\mathrm{inn}}$.

Following the inequalities from (67) to (73), we have:

$$\begin{aligned}
\|\hat{x}_\theta - \tilde{x}_\theta\| &\le \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)|r + \delta^-_{\mathrm{inn}} - \|z'_\theta\|\| + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (91)\\
&\overset{(a)}{\le} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)(|\delta^+_{\mathrm{inn}} - \delta^-_{\mathrm{inn}}|) + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (92)\\
&\le \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\delta_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (93)\\
&\le \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi^{-1}_\theta, \mathcal{X}_\theta \cup \mathcal{K}_\theta \cup \Phi_\theta(\mathcal{B}))\epsilon_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (94)\\
&\overset{(b)}{\le} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}_\theta)\mathrm{Lip}(\Phi^{-1}_\theta, \mathcal{X}_\theta)\epsilon_{\mathrm{inn}} + \mathrm{D}(\Phi_\theta, \mathcal{Z}_\theta)\epsilon_{\mathrm{pre}} && (95)\\
&= \mathrm{D}(\Phi^{-1}_\theta, \mathcal{X}_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pre}}), && (96)
\end{aligned}$$

where the inequality (a) is by $\delta^-_{\mathrm{inn}} \le \|z'_\theta\| - r \le \delta^+_{\mathrm{inn}}$ according to inequalities (53) and applying the triangle inequality. The inequality (b) is owning to $\Phi_\theta(\mathcal{B}) \subseteq \mathcal{K}_\theta \subseteq \mathcal{X}_\theta$.

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: $\|\hat{x}_\theta - \tilde{x}_\theta\| \le \mathrm{D}(\Phi^{-1}_\theta, \mathcal{X}'_\theta)(\epsilon_{\mathrm{inn}} + \epsilon_{\mathrm{pro}})$.

■

F.4 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: $\hat{x}_\theta = \mathrm{HP}^{\Phi_\theta}_{\mathcal{K}_\theta}(\tilde{x}_\theta) = \Phi_\theta(\mathrm{Proj}_{\mathcal{B}}(\Phi_\theta^{-1}(\tilde{x}_\theta)))$. 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 $\epsilon_{\text{pro}} = \|\hat{x}^*_\theta - \tilde{x}_\theta\|$.

The homeomorphic projection error is bounded as:

$$\epsilon_{\text{pro}} \leq \|\hat{x}_\theta - \tilde{x}_\theta\| \leq \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}'_\theta)\epsilon_{\text{pro}}. \tag{97}$$

Proof

By definition of homeomorphic projection operator under feasible mapping, the homeomorphic projection error is bounded as:

$$\begin{aligned}
\|\hat{x}_\theta - \tilde{x}_\theta\| &= \|\Phi_\theta(\mathrm{Proj}_{\mathcal{B}}(\Phi_\theta^{-1}(\tilde{x}_\theta))) - \Phi_\theta(\Phi_\theta^{-1}(\tilde{x}_\theta))\| && (98)\\
&\overset{(a)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}'_\theta)\|\mathrm{Proj}_{\mathcal{B}}(\Phi_\theta^{-1}(\tilde{x}_\theta)) - \Phi_\theta^{-1}(\tilde{x}_\theta)\| && (99)\\
&\overset{(b)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}'_\theta)\|\Phi_\theta^{-1}(\hat{x}^*_\theta) - \Phi_\theta^{-1}(\tilde{x}_\theta)\| && (100)\\
&\overset{(c)}{\leq} \mathrm{Lip}(\Phi_\theta, \mathcal{Z}'_\theta)\mathrm{Lip}(\Phi_\theta^{-1}, \mathcal{X}'_\theta)\|\hat{x}^*_\theta - \tilde{x}_\theta\| && (101)\\
&= \mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}'_\theta)\epsilon_{\text{pro}} && (102)
\end{aligned}$$

where the inequality (a) is based on the Lipschitz constant of Φ_θ over $\mathcal{Z}'_\theta = \Phi_\theta^{-1}(\mathcal{X}'_\theta)$. The inequality (b) is based on the property of standard projection operation. The inequality (c) is based on the Lipschitz constant of Φ_θ^{-1} over $\mathcal{X}'_\theta$ and the definition of projection distance ϵ_{pro}. The inequality (d) is based on the first item in Lemma 15 and the definition of projection error.

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 13 *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 $\mathrm{D}(\Phi_\theta^{-1}, \mathcal{X}'_\theta) = 1$. The homeomorphic projection distance is exactly the same as the standard projection distance.

■

Appendix G. Proof for Theorem 12

Proof

First, we define the ϵ-covering dataset as follows:

Definition 14 (ϵ-covering dataset) *Let $\mathcal{X} \subset \mathbb{R}^n$ be a compact set. A finite-samples dataset $\mathcal{D} = x_i{}_{i=1}^N \subset \mathcal{X}$ is an ϵ-cover of $\mathcal{X}$ if $\forall y \in \mathcal{X}$, $\exists x \in \mathcal{D}$ such that $|x - y| \leq \epsilon$. The size of the dataset is of order $N = |\mathcal{D}| = \mathcal{O}((\text{diam}(\mathcal{X})/\epsilon)^n)$.*

Given that $\mathcal{D}_1$ is an r_c-covering dataset for Θ, it implies that for any $\theta \in \Theta$, there exists $\theta_0 \in \mathcal{D}_1$ such that $|\theta - \theta^0| \leq r_c$.

First, we can bound the variation of the constraint boundary for a pair (θ, θ^0) as follows:

$$\mathrm{d_H}(\partial\mathcal{K}_\theta, \partial\mathcal{K}_{\theta^0}) \leq \|\theta - \theta^0\| C_0 \leq C_0 r_c \tag{103}$$

where $C_0 = \sup_{\theta_1,\theta_2\in\Theta,\theta_1\neq\theta_2}\{\mathrm{d_H}(\partial\mathcal{K}_{\theta_1}, \partial\mathcal{K}_{\theta_2})/\|\theta_1 - \theta_2\|\}$ represents the "Lipschitz" of the mapping from input parameters to the constraint set. Therefore, $C_0 r_c$ provides an upper bound for the variation of the constraint boundary with respect to the variation of the input parameter.

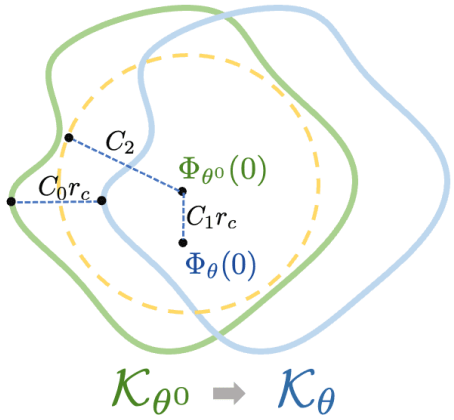

Figure 10: Illustration of $C_0 r_c$, $C_1 r_c$, and C_2.

Next, we can bound the variation of the mapped center $\Phi_\theta(0)$ for a pair of (θ, θ^0) as follows:

$$\|\Phi(0, \theta) - \Phi(0, \theta^0)\| \leq \|\theta - \theta^0\| \mathrm{Lip}(\Phi(0, \cdot), \Theta) \leq r_c C_1, \tag{104}$$

where $C_1 = \mathrm{Lip}(\Phi(0, \cdot), \Theta)$ denotes the Lipschitz constant of the trained INN mapped center over the input parameters. Consequently, $C_1 r_c$ provides an upper bound for the variation of the mapped center in relation to the variation of the input parameter.

Given that the INN is valid for the interior of the constraint set under the r_c-covering dataset, i.e., for any $\theta^0 \in \mathcal{D}_1$, $\Phi_{\theta^0}(0) \in \mathcal{K}^\circ_{\theta^0}$, there exists $r > 0$ such that $\mathcal{B}(\Phi_{\theta^0}(0), r) \subseteq \mathcal{K}_{\theta^0}$. This represents an inner approximation for the constraint set.

We can then denote the largest radius of the inner ball as follows:

$$C_2 = \arg\sup_{r>0}\{\mathcal{B}(\Phi_{\theta^0}(0), r) \subseteq \mathcal{K}_{\theta^0},\ \forall \theta^0 \in \mathcal{D}_1\} \tag{105}$$

By this definition, we have $\mathcal{B}(\Phi_{\theta^0}(0), C_2) \subseteq \mathcal{K}_{\theta^0}$ for any $\theta^0 \in \mathcal{D}_1$.

If the condition $C_1 r_c + C_0 r_c \leq C_2$ holds, then the following relations will also hold:

$$(i) \qquad \Phi_\theta(0) \overset{(a)}{\in} \mathcal{B}(\Phi_{\theta_0}(0), C_1 r_c) \overset{(b)}{\subseteq} \mathcal{B}(\Phi_{\theta_0}(0), C_2) \overset{(c)}{\subseteq} \mathcal{K}_{\theta_0} \tag{106}$$

The relation (a) is due to $\|\Phi_\theta(0) - \Phi_{\theta_0}(0)\| \leq C_1 r_c$ as shown in (104). The relation (b) is validated by the inequality $C_1 r_c \leq C_2$. The relation (c) is established by the definition given in equation (105). Therefore, if the condition $C_1 r_c + C_0 r_c \leq C_2$ holds, we can conclude that $\Phi_\theta(0)$ is an interior point of the constraint set $\mathcal{K}_{\theta_0}$.

Next, we have:

$$(ii) \qquad \mathcal{B}(\Phi_\theta(0), C_0 r_c) \overset{(a)}{\subseteq} \mathcal{B}(\Phi_\theta(0), C_2 - C_1 r_c) \overset{(b)}{\subseteq} \mathcal{B}(\Phi_{\theta_0}(0), C_2) \subseteq \mathcal{K}_{\theta_0} \tag{107}$$

The relation (a) is derived from the condition $C_2 - C_1 r_c \geq C_0 r_c$. The relation (b) is due to $\|\Phi_\theta(0) - \Phi_{\theta_0}(0)\| \leq C_1 r_c$ as shown in equation (104). Consequently, if the condition $C_2 - C_1 r_c \geq C_0 r_c$ holds, we have an inner approximation ball $\mathcal{B}(\Phi_\theta(0), C_0 r_c)$ for the constraint set $\mathcal{K}_{\theta_0}$. This also implies that the distance between $\Phi_\theta(0)$ and the constraint boundary $\partial\mathcal{K}_{\theta_0}$ is at least $C_0 r_c$, i.e., $\mathrm{d_h}(\Phi_\theta(0), \partial\mathcal{K}_{\theta_0}) \geq C_0 r_c$.

To proceed, let's define a set as $\mathcal{X} = \mathcal{K}_{\theta_0} \setminus (\bigcup_{x \in \partial\mathcal{K}_{\theta_0}} \mathcal{B}(x, C_0 r_c))$. This represents a subset of $\mathcal{K}_{\theta_0}$ such that $\mathcal{X} + \mathcal{B}(0, C_0 r_c) = \mathcal{K}_{\theta_0}$. The set $\mathcal{X}$ is not empty since $\Phi_\theta(0)$ is an interior point for $\mathcal{K}_{\theta_0}$ and $\mathrm{d_h}(\Phi_\theta(0), \partial\mathcal{K}_{\theta_0}) \geq C_0 r_c$. Therefore, we can assert that $\Phi_\theta(0) \in \mathcal{X}$.

On the other hand, given that $\mathrm{d_H}(\partial\mathcal{K}_\theta, \partial\mathcal{K}_{\theta^0}) \leq C_0 r_c$, the boundary $\partial\mathcal{K}_\theta$ is contained in set as $\partial\mathcal{K}_\theta \subseteq \partial\mathcal{K}_{\theta_0} + \mathcal{B}(0, C_0 r_c)$. It implies $\mathcal{K}_{\theta_0} \setminus (\bigcup_{x \in \partial\mathcal{K}_{\theta_0}} \mathcal{B}(x, C_0 r_c)) \subseteq \mathcal{K}_\theta \subseteq \mathcal{K}_{\theta_0} + \mathcal{B}(0, C_0 r_c)$. As a result, $\Phi_\theta(0)$ is also an interior point of $\mathcal{K}_\theta$, as $\Phi_\theta(0) \in \mathcal{X} \subseteq \mathcal{K}_\theta$.

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

■

Appendix H. Technical Lemmas

Lemma 15 (Property of distortion) *The distortion for a homeomorphic mapping $\psi \in \mathcal{H}^n$ over the compact set $\mathcal{Z}$ is denoted as $\mathrm{D}(\psi, \mathcal{Z})$. Then the following properties hold:*

(1) $\mathrm{D}(\psi, \mathcal{Z}) = \mathrm{Lip}(\psi, \mathcal{Z})\mathrm{Lip}(\psi^{-1}, \mathcal{X})$, *where* $\mathcal{X} = \psi(\mathcal{Z})$, $\mathrm{Lip}(\cdot, \cdot)$ *indicates the Lipschitz constant of mapping over a set.*

(2) $\mathrm{D}(\psi, \mathcal{Z}) = \mathrm{D}(\psi^{-1}, \mathcal{X}) = \frac{\sup_{z \in \mathcal{Z}} \{\sigma_1(\mathrm{J}_\psi(z))\}}{\inf_{z \in \mathcal{Z}} \{\sigma_n(\mathrm{J}_\psi(z))\}} = \frac{\sup_{x \in \mathcal{X}} \{\sigma_1(\mathrm{J}_{\psi^{-1}}(x))\}}{\inf_{x \in \mathcal{X}} \{\sigma_n(\mathrm{J}_{\psi^{-1}}(x))\}}$, *where* $\sigma_1(\cdot) \geq \cdots \geq \sigma_n(\cdot)$ *denotes the sorted singular values of the Jacobian matrix* J *for mapping at a certain point.*

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

(4) Let $\psi_1 : \mathcal{Z} \to \mathcal{X}$ *and* $\psi_2 : \mathcal{X} \to \mathcal{Y}$ *are homeomorphic mappings, then* $\mathrm{D}(\psi_2 \circ \psi_1, \mathcal{Z}) \leq \mathrm{D}(\psi_1, \mathcal{Z})\mathrm{D}(\psi_2, \mathcal{X})$.

Proof (1) According to Definition 4, the distortion of mapping ψ over set $\mathcal{Z}$ is defined as $\mathrm{D}(\psi, \mathcal{Z}) = \kappa_2/\kappa_1$, where κ_2 is also the definition of Lipschitz constant of ψ as

$$\kappa_2 = \sup_{z_1, z_2 \in \mathcal{Z}, z_1 \neq z_2} \{\|\psi(z_1) - \psi(z_2)\|/\|z_1 - z_2\|\} = \mathrm{Lip}(\psi, \mathcal{Z}), \tag{108}$$

and κ_1 is defined as:

$$1/\kappa_1 = 1/\inf_{z_1, z_2 \in \mathcal{Z}, z_1 \neq z_2} \{\|\psi(z_1) - \psi(z_2)\|/\|z_1 - z_2\|\} \tag{109}$$

$$= \sup_{z_1, z_2 \in \mathcal{Z}, z_1 \neq z_2} \{\|z_1 - z_2\|/\|\psi(z_1) - \psi(z_2)\|\} \tag{110}$$

$$= \sup_{x_1, x_2 \in \mathcal{X}, x_1 \neq x_2} \{\|\psi^{-1}(x_1) - \psi^{-1}(x_2)\|/\|x_1 - x_2\|\} = \mathrm{Lip}(\psi^{-1}, \mathcal{X}). \tag{111}$$

Therefore, the distortion can be represented as the multiplication of the Lipschitz constant of forward and inverse mappings as: $\mathrm{D}(\psi, \mathcal{Z}) = \mathrm{Lip}(\psi, \mathcal{Z})\mathrm{Lip}(\psi^{-1}, \mathcal{X})$.

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

$$\mathrm{Lip}(\psi, \mathcal{Z}) = \sup_{z \in \mathcal{Z}} \{\sigma_1(\mathrm{J}_\psi(z))\}, \qquad \mathrm{Lip}(\psi^{-1}, \mathcal{X}) = \sup_{x \in \mathcal{X}} \{\sigma_1(\mathrm{J}_{\psi^{-1}}(x))\}. \tag{112}$$

Because of the invertibility of homeomorphic mapping, such that the Jacobian matrix is also invertible as $\mathrm{J}_\psi^{-1}(z) = \mathrm{J}_{\psi^{-1}}(x)$ for $x = \psi(z)$. According to the property of invertible matrix, we can represent the singular values as $\sigma_1(\mathrm{J}_\psi(z)) = 1/\sigma_n(\mathrm{J}_{\psi^{-1}}(x))$ for $x = \psi(z)$.

Thus, the distortion can be represented as:

$$\mathrm{D}(\psi, \mathcal{Z}) = \frac{\sup_{z \in \mathcal{Z}}\{\sigma_1(\mathrm{J}_\psi(z))\}}{\inf_{z \in \mathcal{Z}}\{\sigma_n(\mathrm{J}_\psi(z))\}} = \frac{\sup_{x \in \mathcal{X}}\{\sigma_1(\mathrm{J}_{\psi^{-1}}(x))\}}{\inf_{x \in \mathcal{X}}\{\sigma_n(\mathrm{J}_{\psi^{-1}}(x))\}} = \mathrm{D}(\psi^{-1}, \mathcal{X}). \tag{113}$$

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

(3) According to Eq. (113), if $\mathcal{Z}_1 \subseteq \mathcal{Z}_2$, then:

$$\mathrm{D}(\psi, \mathcal{Z}_1) = \frac{\sup_{z \in \mathcal{Z}_1}\{\sigma_1(\mathrm{J}_\psi(z))\}}{\inf_{z \in \mathcal{Z}_1}\{\sigma_n(\mathrm{J}_\psi(z))\}} \leq \frac{\sup_{z \in \mathcal{Z}_2}\{\sigma_1(\mathrm{J}_\psi(z))\}}{\inf_{z \in \mathcal{Z}_2}\{\sigma_n(\mathrm{J}_\psi(z))\}} = \mathrm{D}(\psi, \mathcal{Z}_2). \tag{114}$$

(4) According to the Lipschitz representation of distortion, we have:

$$\mathrm{D}(\psi_2 \circ \psi_1, \mathcal{Z}) = \mathrm{Lip}(\psi_2 \circ \psi_1, \mathcal{Z})\mathrm{Lip}(\psi_1^{-1} \circ \psi_2^{-1}, \mathcal{Y}) \tag{115}$$

$$\overset{(a)}{\leq} \mathrm{Lip}(\psi_1, \mathcal{Z})\mathrm{Lip}(\psi_2, \mathcal{X})\mathrm{Lip}(\psi_1^{-1}, \mathcal{X})\mathrm{Lip}(\psi_2^{-1}, \mathcal{Y}) \tag{116}$$

$$\overset{(b)}{\leq} \mathrm{D}(\psi_1, \mathcal{Z})\mathrm{D}(\psi_2, \mathcal{X}) \tag{117}$$

where inequality (a) is by the property of Lipschitz of Function composition. Equality is due to the Lipschitz representation of distortion.

■

Lemma 16 (Property of Hausdroff distance) *The Hausdroff distance between sets is denoted as* $\mathrm{d_H}(\mathcal{X},\mathcal{Y}) = \max\{\mathrm{d_h}(\mathcal{X},\mathcal{Y}), \mathrm{d_h}(\mathcal{Y},\mathcal{X})\}$*, where the one-side Hausdroff distance is defined as:* $\mathrm{d_h}(\mathcal{X},\mathcal{Y}) = \sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}}\{\|x-y\|\}$*. Then the following properties hold:*

(1) $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0,r)$ *if and only if* $\mathrm{d_h}(\mathcal{Y},\mathcal{X}) \le r$*; if* $\mathcal{Y} = \mathcal{X} + \mathcal{B}(0,r)$ *then* $\mathrm{d_H}(\mathcal{Y},\mathcal{X}) \le r$*;*

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

(3) Let $\mathcal{B} = \{z \in \mathbb{R}^n \mid \|z\|_2 \le r\}$*, then* $\mathrm{d_h}(\partial\mathcal{X},\partial\mathcal{B}) \ge \sup_{x\in\partial\mathcal{X}}\{|\|x\| - r|\}$*.*

Proof (1) First, to prove if $\mathrm{d_h}(\mathcal{Y},\mathcal{X}) \le r$ then $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0,r)$, we have:

$$\inf_{x\in\mathcal{X}}\{\|x-y\|\} \le \sup_{y\in\mathcal{Y}} \inf_{x\in\mathcal{X}}\{\|x-y\|\} = \mathrm{d_h}(\mathcal{Y},\mathcal{X}) \le r, \tag{118}$$

such that $\forall y \in \mathcal{Y}$, the distance $\inf_{x\in\mathcal{X}}\{\|x-y\|\} \le r$. 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 if $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0,r)$ then $\mathrm{d_h}(\mathcal{Y},\mathcal{X}) \le r$, 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

$$\begin{aligned}\mathrm{d_h}(\mathcal{Y},\mathcal{X}) &= \sup_{y\in\mathcal{Y}} \inf_{x\in\mathcal{X}}\{\|x-y\|\} \le \sup_{y\in\mathcal{Y}'} \inf_{x\in\mathcal{X}}\{\|x-y\|\} &(119)\\ &= \sup_{x'\in\mathcal{X},\vec{r}} \inf_{x\in\mathcal{X}}\{\|x-(x'+\vec{r})\|\} \le \sup_{x'\in\mathcal{X},\vec{r}}\{\|x'-(x'+\vec{r})\|\} = r. &(120)\end{aligned}$$

Therefore, we conclude $\mathcal{Y} \subseteq \mathcal{X} + \mathcal{B}(0,r)$ if and only if $\mathrm{d_h}(\mathcal{Y},\mathcal{X}) \le r$.

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:

$$\mathrm{d_h}(\mathcal{X},\mathcal{Y}) = \sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}}\{\|x-y\|\} = \sup_{x\in\mathcal{X}} \inf_{x'\in\mathcal{X},\vec{r}}\{\|x-(x'+\vec{r})\|\} \le \sup_{x\in\mathcal{X}}\{\|x-(x+\vec{r})\|\} = r. \tag{121}$$

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

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

$$\begin{aligned}\mathrm{d_h}(\psi(\mathcal{X}),\psi(\mathcal{Y})) &= \sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}}\{\|\psi(x)-\psi(y)\|\} &(122)\\ &\le \mathrm{Lip}(\psi,\mathcal{X}\cup\mathcal{Y}) \sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}}\{\|x-y\|\} &(123)\\ &= \mathrm{Lip}(\psi,\mathcal{X}\cup\mathcal{Y})\mathrm{d_h}(\mathcal{X},\mathcal{Y}) \le \mathrm{Lip}(\psi,\mathcal{X}\cup\mathcal{Y})\mathrm{d_H}(\mathcal{X},\mathcal{Y}), &(124)\end{aligned}$$

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

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

$$\begin{aligned}\mathrm{d_h}(\partial\mathcal{X},\partial\mathcal{B}) &= \sup_{x\in\partial\mathcal{X}} \inf_{z\in\partial\mathcal{B}}\{\|x-z\|\} = \sup_{x\in\partial\mathcal{X}}\{\|x - \mathrm{Proj}_{\partial\mathcal{B}}(x)\|\} &(125)\\ &\ge \sup_{x\in\partial\mathcal{X}}\{|\|x\| - \|\mathrm{Proj}_{\partial\mathcal{B}}(x)\||\} &(126)\\ &= \sup_{x\in\partial\mathcal{X}}\{|\|x\| - r|\}. &(127)\end{aligned}$$

■

Lemma 17 *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:

$$\|\lambda a - b\|_2^2 - \|\lambda a - a\|_2^2 = \lambda^2\|a\|_2^2 + \|b\|_2^2 - 2\lambda\|a\|_2\|b\|_2\cos(\beta) - \lambda^2\|a\|_2^2 - \|a\|_2^2 + 2\lambda\|a\|_2^2 \tag{128}$$

$$= 2\lambda\|a\|_2^2 - 2\lambda\|a\|_2\|b\|_2\cos(\beta) \tag{129}$$

$$= 2\lambda\|a\|_2^2(1 - \cos(\beta)) \geq 0. \tag{130}$$

■

Appendix I. Experiment Settings

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

I.1 Formulation of Constrained Optimization Problem

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

$$\begin{aligned} \textbf{QP}: \ \min_{x\in\mathbb{R}^n} \ & \frac{1}{2}x^\mathsf{T}Qx + p^\mathsf{T}x \\ \text{s.t.} \ \ & Ax = \theta, \ Gx \leq h, \ l \leq x \leq u, \end{aligned} \tag{131}$$

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

$$\begin{aligned} \textbf{Convex QCQP}: \ \min_{x\in\mathbb{R}^n} \ & \frac{1}{2}x^\mathsf{T}Qx + p^\mathsf{T}x \\ \text{s.t.} \ \ & x^\mathsf{T}H_ix + g_i^\mathsf{T}x \leq h_i, \ i = 1, ..., n_{\text{ineq}} \\ & Ax = \theta, \ l \leq x \leq u, \end{aligned} \tag{132}$$

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

$$\begin{aligned} \textbf{SOCP}: \ \min_{x\in\mathbb{R}^n} \ & \frac{1}{2}x^\mathsf{T}Qx + p^\mathsf{T}x \\ \text{s.t.} \ \ & \|G_ix + h_i\|_2 \leq c_i^\mathsf{T}x + d_i, \ i = 1, ..., n_{\text{ineq}} \\ & Ax = \theta, \ l \leq x \leq u, \end{aligned} \tag{133}$$

where $G_i \in \mathbb{R}^{m\times n}$, $h_i \in \mathbb{R}^m$, $c_i \in \mathbb{R}^n$, and $d_i \in \mathbb{R}$ for $i = 1, ..., n_{\text{ineq}}$.

$$\begin{aligned} \textbf{SDP}: \quad & \min_{X\in \text{Sym}_n} \quad \text{tr}(CX) \\ & \text{s.t.} \quad \text{tr}\left(A_i X\right) = \theta_i,\ X \succeq 0,\ L \le X \le U, \end{aligned} \tag{134}$$

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

A compact formulation of the AC-OPF problem with complex variables is as follows.

$$\textbf{AC-OPF}: \ \min \quad p_g^{\mathsf{T}} Q p_g + b^{\mathsf{T}} p_g \tag{135}$$

$$\text{s.t.} \quad p_g^{\min} \le p_g \le p_g^{\max}, \tag{136}$$

$$q_g^{\min} \le q_g \le q_g^{\max}, \tag{137}$$

$$v_m^{\min} \le |v| \le v_m^{\max}, \tag{138}$$

$$v_a^{\min} \le \angle v_i - \angle v_j \le v_a^{\max} \tag{139}$$

$$|v_i(\bar{v}_i - \bar{v}_j)\bar{w}_{ij}| \le S_{ij}^{max}, \tag{140}$$

$$(p_g - p_d) + (q_g - q_d)\,\mathrm{i} = \text{diag}(v)\bar{W}\bar{v}, \tag{141}$$

$$\text{var.} \quad p_g \in \mathbb{R}^n, q_g \in \mathbb{R}^n, v \in \mathbb{C}^n \tag{142}$$

For a power network comprised of n nodes, the vectors $p_d, q_d \in \mathbb{R}^n$ represent the real and reactive power demand at each node, the vectors $p_g, q_g \in \mathbb{R}^n$ denote the real and reactive power generation, and the vector $v \in \mathbb{C}^n$ is the nodal voltage variable. The admittance matrix $W \in \mathbb{C}^{n\times n}$ characterizes the physical properties and topology of the power network, with $\bar{W}$ denoting its complex conjugate transpose. The generation cost is represented by a quadratic function with matrix $Q \in \mathbb{R}^{n\times n}$ and vector $b \in \mathbb{R}^n$. The constraints include generation limits $(p_g^{\min}, p_g^{\max}, q_g^{\min}, q_g^{\max})$, voltage magnitude bounds $(v_m^{\min}, v_m^{\max})$, voltage angle limits $(v_a^{\min}, v_a^{\max})$, line flow limits $(S_{ij}^{\max})$, and power flow balance equations.

I.2 Structure and Parameters for NN Predictor

We adopt the fully connected NN with ReLU activation and residual connection, denoted as $F(\cdot)$, to predict the optimal solution for constrained optimization problems given the input parameters. To enforce simple upper/lower bound on the predicted solution, we adopt a scaled sigmoid function in the last layer to enforce the limits. The parameters are in Table 7.

For supervised training schemes, we collect the optimal input-solution pairs under different input parameters using an iterative solver. Subsequently, we aim to minimize the mean square error (MSE) between the solution predicted by the NN predictor and the optimal solutions, as well as the constraint violation for predicted solutions. For unsupervised training schemes, we only sample the input parameters and minimize the objective value regularized by the constraint violation for NN-predicted solutions. Two training schemes are represented as:

$$\text{Supervise}: \ \min_F \mathbb{E}_{\theta, x_\theta^*}[\|F(\theta) - x_\theta^*\|_2^2 + \lambda_1^s \|\text{ReLU}(\mathbf{g}(F(\theta), \theta))\|_1 + \lambda_2^s \|(\mathbf{h}(F(\theta), \theta))\|_1],$$

$$\text{Unsupervise}: \ \min_F \mathbb{E}_\theta[f(F(\theta), \theta) + \lambda_1^u \|\text{ReLU}(\mathbf{g}(F(\theta), \theta))\|_1 + \lambda_2^u \|(\mathbf{h}(F(\theta), \theta))\|_1],$$

where the second term stands for the penalty for inequality constraint violation, and the third term is the penalty for equality constraint violation. By adjusting the penalty coefficient λ, we can train the NN predictor to achieve different qualities in terms of solution feasibility and optimality. For the equality constraint, we can utilize the variable selection and completion techniques elaborated in Appendix A to ensure its feasibility.

Table 7: Structure of NN predictor in experiments

Parameter	Value
NN structure	
dimension of input and output layers	(d, n)
activation function	ReLU$(\cdot)$
number of layer	3
last layer	Sigmoid$(\cdot)$
NN training parameters	
number of training samples	10,000
number of testing samples	1,024
optimizer	Adam
learning rate	0.001
the coefficient for objective function	0.01
the coefficient for penalty function	0.1

I.3 Structure and Parameters for INN Mapping

Inspired by the generative flow with the invertible 1x1 Convolutions (GLOW) model (Kingma and Dhariwal, 2018), we construct an Invertible Neural Network (INN) composed of three basic layers. These include the Actnorm layer, the invertible convolution layer, and the bi-Lipschitz autoregressive layer. Notably, the first two layers are specific instances of the invertible linear layer, while the third one is a special case of the bi-Lip affine Coupling layer. To train the MDH mapping for different examples following Alg. 2, we select the parameters shown in Table 8.

Table 8: Structure of INN mapping in experiments

Parameter	Value
INN structure	
dimension of input and output layers	$(n + d, n)$
basic INN block	$\left\{\begin{matrix}\text{Actnorm, Inv. Conv.} \\ \text{Bi-Lip Autoregressive}\end{matrix}\right\}$
number of INN block	3
the last layer	Sigmoid$(\cdot)$
INN training parameters	
number of QMC samples $z \in \mathcal{B}$	10,000
number of uniform samples $\theta \in \Theta$	10,000
optimizer	Adam
learning rate	0.0001
the coefficient for penalty	1
the coefficient for distortion	0.1

References

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

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

Lars Ahlfors and Lipman Bers. Riemann's mapping theorem for variable metrics. *Annals of Mathematics*, pages 385–404, 1960.

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

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

Sogol Babaeinejadsarookolaee, Adam Birchfield, Richard D Christie, Carleton Coffrin, Christopher DeMarco, Ruisheng Diao, Michael Ferris, Stephane Fliscounakis, Scott Greene, Renke Huang, et al. The power grid library for benchmarking ac optimal power flow algorithms. *arXiv preprint arXiv:1908.02788*, 2019.

Kyri Baker. Learning warm-start points for ac optimal power flow. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021.

Daniel Bienstock and Abhinav Verma. Strong np-hardness of ac power flows feasibility. *Operations Research Letters*, 47(6):494–501, 2019.

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

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Tom Brown, Jonas Hörsch, and David Schlachtberger. Pypsa: Python for power system analysis. *arXiv preprint arXiv:1707.09913*, 2017.

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

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

Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *arXiv preprint arXiv:2103.12828*, 2021b.

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

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

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

Frederik Diehl. Warm-starting ac optimal power flow with graph neural networks. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 1–6, 2019.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

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

Mahyar Fazlyab, Manfred Morari, and George J Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, 67(1):1–15, 2020.

Mahyar Fazlyab, Manfred Morari, and George J Pappas. An introduction to neural network analysis via semidefinite programming. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6341–6350. IEEE, 2021.

Ferdinando Fioretto, Terrence WK Mak, and Pascal Van Hentenryck. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 630–637, 2020.

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

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

Stefan Geschke. 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.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Neel Guha, Zhecheng Wang, Matt Wytock, and Arun Majumdar. Machine learning for ac optimal power flow. *arXiv preprint arXiv:1910.08842*, 2019.

Howard Heaton, Samy Wu Fung, Aviv Gibali, and Wotao Yin. Feasibility-based fixed point networks. *Fixed Point Theory and Algorithms for Sciences and Engineering*, 2021(1): 1–19, 2021.

Howard Heaton, Samy Wu Fung, Alex Tong Lin, Stanley Osher, and Wotao Yin. Wasserstein-based projections with applications to inverse problems. *SIAM Journal on Mathematics of Data Science*, 4(2):581–603, 2022.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Chin-Wei Huang, Ricky TQ Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2020.

Wanjun Huang, Xiang Pan, Minghua Chen, and Steven H Low. Deepopf-v: Solving ac-opf problems efficiently. *IEEE Transactions on Power Systems*, 37(1):800–803, 2021.

Wanjun Huang, Minghua Chen, and Steven H Low. Unsupervised learning for solving ac optimal power flows: Design, analysis, and experiment. *IEEE Transactions on Power Systems*, 2024.

Isao Ishikawa, Takeshi Teshima, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Universal approximation property of invertible neural networks. *arXiv preprint arXiv:2204.07415*, 2022.

Elias Khalil, Pierre Le Bodic, Le Song, George Nemhauser, and Bistra Dilkina. Learning to branch in mixed integer programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

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

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

Frederic Koehler, Viraj Mehta, and Andrej Risteski. Representational aspects of depth and conditioning in normalizing flows. In *International Conference on Machine Learning*, pages 5628–5636. PMLR, 2021.

Zhifeng Kong and Kamalika Chaudhuri. Universal approximation of residual flows in maximum mean discrepancy. *arXiv preprint arXiv:2103.05793*, 2021.

James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder. End-to-end constrained optimization learning: A survey. *arXiv preprint arXiv:2103.16378*, 2021.

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

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

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

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

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

JG Liao and Arthur Berg. Sharpening jensen's inequality. *The American Statistician*, 2018.

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

Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. *arXiv preprint arXiv:2202.08345*, 2022.

Junlong Lyu, Zhitang Chen, Chang Feng, Wenjing Cun, Shengyu Zhu, Yanhui Geng, Zhijie Xu, and Yongwei Chen. Universality of parametric coupling flows over parametric diffeomorphisms. *arXiv preprint arXiv:2202.02906*, 2022.

Deanna Needell, William Swartworth, and David P Woodruff. Testing positive semidefiniteness using linear measurements. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 87–97. IEEE, 2022.

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

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

Xiang Pan, Tianyu Zhao, and Minghua Chen. 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.

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

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

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Seonho Park, Wenbo Chen, Terrence WK Mak, and Pascal Van Hentenryck. Compact optimization learning for ac optimal power flow. *IEEE Transactions on Power Systems*, 2023.

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

Rajiv Sambharya, Georgina Hall, Brandon Amos, and Bartolomeo Stellato. End-to-end learning to warm-start for real-time quadratic optimization. In *Learning for Dynamics and Control Conference*, pages 220–234. PMLR, 2023.

Rajiv Sambharya, Georgina Hall, Brandon Amos, and Bartolomeo Stellato. Learning to warm-start fixed-point optimization algorithms. *Journal of Machine Learning Research*, 25(166):1–46, 2024.

Patrick Schmidt, Janis Born, Marcel Campen, and Leif Kobbelt. Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019.

Rajesh Sharma, Madhu Gupta, and Girish Kapoor. Some better bounds on the variance with applications. *Journal of Mathematical Inequalities*, 4(3):355–363, 2010.

Stephen Smale. On the structure of manifolds. *American Journal of Mathematics*, 84(3): 387–399, 1962.

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

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

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.

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

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

Jussi Väisälä. A proof of the mazur-ulam theorem. *The American mathematical monthly*, 110(7):633–635, 2003.

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

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

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Ruigang Wang, Krishnamurthy Dvijotham, and Ian R Manchester. Monotone, bi-lipschitz, and polyak-\l {} ojasiewicz networks. *arXiv preprint arXiv:2402.01344*, 2024.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

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

Ahmed S Zamzam and Kyri Baker. 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)*, pages 1–6. IEEE, 2020.

Han Zhang, Xi Gao, Jacob Unterman, and Tom Arodz. Approximation capabilities of neural odes and invertible residual networks. In *International Conference on Machine Learning*, pages 11086–11095. PMLR, 2020.

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

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

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

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

Ray D Zimmerman, Carlos E Murillo-Sánchez, and Deqiang Gan. Matpower. *PSERC.[Online]. Software Available at: http://www. pserc. cornell. edu/matpower*, 1997.