EFFICIENT BISECTION PROJECTION TO ENSURE NN SOLUTION FEASIBILITY FOR OPTIMIZATION OVER GENERAL SET

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks (NNs) have shown promise in solving constrained optimization problems in real-time. However, ensuring that NN-generated solutions strictly adhere to constraints is challenging due to NN prediction errors. Recent methods have achieved feasibility guarantees over ball-homeomorphic sets with low complexity and bounded optimality loss, yet extending these guarantees to more general sets remains largely open. In this paper, we develop *Bisection Projection*, an efficient approach to ensure NN solution feasibility for optimization over general compact sets with non-empty interiors, irrespective of their ball-homeomorphic properties. Our method begins by identifying multiple interior points (IPs) within the constraint set, chosen based on their eccentricity modulated by the NN infeasibility region. We utilize another unsupervised-trained NN (called IPNN) to map inputs to these interior points, thereby reducing the complexity of computing these IPs in run-time. For NN solutions initially deemed infeasible, we apply a bisection procedure that adjusts these solutions towards the identified interior points, ensuring feasibility with minor projection-induced optimality loss. We prove the feasibility guarantee and bound the optimality loss of our approach under mild conditions. Extensive simulations, including non-convex optimal power flow problems in large-scale networks, demonstrate that bisection projection outperforms existing methods in solution feasibility and computational efficiency with comparable optimality losses.

1 INTRODUCTION

Constrained Optimization (CO) plays an essential role in various engineering fields, such as supply chain management, transportation, and power systems. To solve CO problems, iterative algorithms, such as interior point methods, have been developed and embedded within commercial solvers like Gurobi. These tools are designed to tackle CO with high precision, providing exact or approximated solutions. However, they can be slow for real-time applications with tight time constraints.

Recent advancements in machine learning (ML) have introduced innovative strategies for solving real-time CO problems, including the end-to-end (E2E) solution mapping (Amo22), the learning-to-optimize (L2O) iterative scheme (CCC$^+$21), and hybrid approaches (KFVHW21). One powerful idea is to leverage the universal approximation ability of neural networks (NNs) (HSW89; LLPS93) to predict high-quality solutions given input parameters, significantly reducing computation time compared to traditional iterative solvers. For instance, by employing this idea, NNs have been trained to solve critical optimal power flow problems in modern grid operations, achieving a 2-4 orders of magnitude speedup over iterative solvers (PZC19; GWWM19; PZCZ20; FMVH20; ZB20; DRK20).

Despite the minor optimality loss and significant speedup of NN-based methods, guaranteeing the feasibility of NN solutions with respect to problem constraints remains a challenge due to inherent NN prediction errors. While recent advances have managed to ensure NN solution feasibility within *ball-homeomorphic* sets with low complexity and bounded optimality loss (LCL23; LCL24), establishing similar guarantees for general sets remains an open problem. A discussion on related works is provided in Section 2, and a summary is presented in Table 1.

In this paper, we develop *Bisection Projection* (BP) as an efficient scheme to recover infeasible NN solutions with bounded optimality loss, with respect to general compact sets beyond ball-homeomorphic ones under mild conditions. Our main contributions are as follows:

▷ In Sec. 4, we introduce the BP framework for ensuring NN solution feasibility. It comprises two steps: (i) pinpointing multiple interior points (IPs) in the constraint set with the minimized eccentricity modulated by the NN infeasibility region and (ii) employing a bisection algorithm to "project" infeasible solutions to constraint boundary with minor optimality loss.

We also leverage an unsupervised trained NN, called IPNN, to predict those IPs, significantly cutting down the run-time complexity in real-time operation. We highlight the applicability of our approach to optimization over general sets that are not necessarily ball-homeomorphic.

▷ In Sec. 5, we prove that BP can recover infeasible solutions with bounded optimality loss related to the eccentricity of used IPs. We also analyze its run-time and training complexities under mild conditions. Our analysis and results are general and go beyond the ball-homeomorphism setting.

▷ In Sec. 6, we carry out extensive experiments over convex and non-covnex problems to evaluate the performance of BP. The results show that it outperforms existing methods in feasibility and run-time complexity while achieving similar optimality losses.

To our knowledge, this is the first work to guarantee NN solution feasibility over general compact sets with bounded optimality loss and low run-time complexity under mild conditions.

2 Related Work

Recently, ML-driven optimization has been an active research field (PZC19; KFVHW21; CCC$^+$21; Amo22). A core challenge is ensuring the NN prediction feasibility over input-dependent constraints. Researchers have developed various methods to enhance solution feasibility, summarized in Table 1.

Equality constraints. Linear equations and certain non-linear equations with constant ranks, can be embedded as neural network layers by predicting a subset of variables and solving for the remaining variables to satisfy the equality constraints (Aba69; Lee13; PZC19; DRK20; LCL23; DWDS23).

Projection approach. To enforce solution feasibility, orthogonal/L2 projection is often employed. However, solving the projection problem either by iterative solver or equivalent optimization layers (AK17; AAB$^+$19; CDB$^+$21; WZG$^+$23) is computationally intensive in real-time. Alternative strategies include gradient-based methods (e.g., DC3 (DRK20)) and L2O models that mimic iterative projection procedures to adjust infeasible solutions (HWFGY21; HFL$^+$22). However, those projection-analogous approaches do not guarantee feasibility for general input-adaptive constraints.

Warm-start approach. The NN predictions can serve as warm-start points for iterative solvers, potentially reducing the number of iterations required to reach the optimal solution (Die19; Bak19; SHAS23; SHAS24). However, it may still be computationally intensive for real-time operations.

Penalty approach. To reduce constraint violations in predicted solutions, researchers have incorporated various penalty functions, such as quadratic penalties, into the NN loss function (COMB19; PZCZ20; ZB20; FMVH20). Additionally, integration of the Karush–Kuhn–Tucker (KKT) conditions as equality constraints has been explored to refine NN performance (NC21b; NC21a; ZCZ21). However, these methods do not consistently ensure solution feasibility due to NN prediction errors.

Sampling approach. To guarantee feasibility, an inner approximation of the original constraint set can be constructed. For linear constraints, a convex combination of sampled vertexes and rays is adopted to ensure feasibility (FNC20; ZSRZ21). For general compact but fixed constraint sets, sampling-based methods are theoretically studied in (KZLD21). However, scalability remains a challenge due to the exponential growth in required samples with increasing problem dimensionality.

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

Gauge function. These works utilize gauge functions (BM08) to constrain the NN. A closed-form bijection, known as gauge mapping, between a hypercube and a polytope is used to bound the NN output within the polytope (TZ22a; TZ22b; LKM23). For fixed convex constraints, RAYEN and several works apply analytic expressions for gauge functions to find feasible boundary solutions (THH23; KU23; LM23). However, these approaches only work for specific convex sets.

Table 1: Existing work for ensuring NN solution feasibility for continuous constrained optimization problems.

Existing Work (see Sec. 2 for ref.)	Constraint Set			Performance Guarantee		
	Input dependent	Non-linear equality	General inequality	Feasibility ensuring	Optimality bound	Low run-time
Orthogonal Proj./Warm-start	✓	✓	✓	✓	✓	✗
Penalty method	✓	✓	✓	✗	✗	✓
Sampling approach	✗	✓	✓	✓	✓	✗
RAYEN	✗	✗ (linear)	✗ (convex)	✓	✓	✓
Preventive learning	✓	✗ (linear)	✗ (linear)	✓	✗	✓
Gauge mapping	✓	✗ (linear)	✗ (linear)	✓	✗	✓
DC3	✓	✓ (CR)	✓	✗	✗	✓
Homeomorphic Projection	✓	✓ (CR)	✗ (BH)	✓	✓	✓
Bisection Projection	✓	✓ (CR)	✓	✓	✓	✓

[1] CR indicates the Jacobian of equality functions is of a constant rank, which includes linear equality and a part of non-linear equality.
[2] BH indicates the constraint set is homeomorphic to a unit ball, which includes all compact convex sets and a part of non-convex sets.

Homeomorphic Projection has been proposed to ensure the NN solution feasibility over *ball-homeomorphic* constraint (LCL23; LCL24). It applies invertible NN to construct a homeomorphism between the constraint set and a unit ball, such that the projection operation can be efficiently conducted over the ball through simple bisection. Nevertheless, the ball-homeomorphism assumption limits its application for optimization problems over general constraint sets.

In summary, existing works either incur high run-time complexity or have limited applicable constraints. In this work, we propose *Bisection Projection* as an efficient scheme to guarantee NN solution feasibility over (fairly) general compact sets with bounded optimality loss under mild conditions.

3 Setting and Open Issue

We consider the following continuous optimization problem over a compact constraint set:

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

where $x \in \mathcal{C}_\theta \subset \mathbb{R}^n$ is the decision variable and $\theta \in \Theta \subset \mathbb{R}^d$ is the input contextual parameter. Without loss of generality, we assume the input domain Θ and constraint set $\mathcal{C}_\theta$ are compact. The objective function $f(x,\theta)$ is continuous and can be non-convex. The optimal solution is denoted as $x^*_\theta \in \arg\min_{x\in\mathcal{C}_\theta}\{f(x,\theta)\}$. The constraint set $\mathcal{C}_\theta$ is specified by inequalities: $\mathcal{C}_\theta = \{x\in\mathbb{R}^n \mid g(x,\theta)\le 0\}$, where $g:\mathbb{R}^{n+d}\to\mathbb{R}^{n_{\text{ineq}}}$ is continuous. While we do not explicitly consider equality constraints in the formulation, we remark that certain equality constraints can be embedded as NN layers and be satisfied (Aba69; PZC19; DRK20). We do carry out simulations for problems with linear/nonlinear equality constraints in Sec. 6 and provide detailed discussion in Appendix A.1.

We further specify the constraint set as follows, beyond those discussed in related works in Sec. 2.

Assumption 1. $\forall\theta\in\Theta$, any interior point of the **compact** set $\mathcal{C}_\theta$ has a neighborhood of **positive** measure within $\mathcal{C}_\theta$.

We **remark** this assumption ensures that the constraint set has a non-empty interior, and it circumvents unusual compact sets in continuous optimization (e.g., vertices of a hypercube or Cantor set) so that our algorithm design and theoretical proof later go through rigorously. Nevertheless, the constraint set under Assumption 1 is very general, covering linear, convex, and ball-homeomorphic sets in all existing works (LCL23; THH23; LCL24). Such a set is also called a "fat" set in measure theory (Lew88). We also discuss the applicability of our approach beyond this assumption in Appendix A.2.

Open issue. As discussed in Sec. 2, various NN-based methods have been developed to solve CO problems with low run-time complexity and minor optimality gap. Denote one such trained NN predictor as $F(\theta):\mathbb{R}^d\to\mathbb{R}^n$, which has a prediction error as $\epsilon_{\text{pre}} = \sup_{\theta\in\Theta}\{\|F(\theta)-x^*_\theta\|\}$. Due to the error ϵ_{pre}, ensuring NN solution feasibility is non-trivial. As illustrated in Fig. 1, an optimal solution often lies on the constraint boundary, such that any positive error may push the NN solution outside the constraint set. Existing approaches, as summarized in Table 1, are either computationally intensive or fail to provide performance guarantees over general input-dependent constraint sets beyond ball-homeomorphic ones.

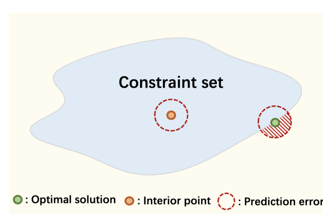


Figure 1: NN predicting optimal solution (on the boundary) incurs infeasibility. NN predicting interior points accommodates errors.

To date, ensuring NN solutions feasibility for CO in (1) under Assumption 1, while maintaining bounded optimality loss and low computational complexity, remains an open and pressing challenge.

4 The Bisection Projection framework

We propose *Bisection Projection* (BP) to "project" infeasible NN solutions onto the constraint set with low run-time complexity and minor optimality loss. As detailed in Section 4.1, this framework applies bisection to iteratively narrow the gap between infeasible points and interior points (IPs) to identify feasible solutions. In Sec. 4.2, to reduce the optimality loss induced by bisection, we introduce the concept of eccentricity for IPs and establish its connection to projection distance. In Sec. 4.3, to reduce the inference time for finding IPs under varying inputs, we train another NN, denoted as IPNN, to obtain IPs fast in run-time.

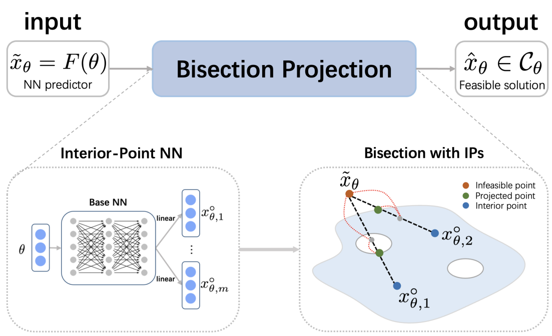


Figure 2: The Bisection Projection framework.

Algorithm 1 Bisection for feasibility.

Input: an IP $x^\circ_\theta \in \mathcal{C}_\theta$ and a prediction $\tilde{x}_\theta \notin \mathcal{C}_\theta$
Output: feasible solution $\hat{x}_\theta \in \mathcal{C}_\theta$

set total iteration K, $\alpha_l = 0$, and $\alpha_u = 1$
for $n = 1 : K$ **do**
  bisection: $\alpha_m = (\alpha_l + \alpha_u)/2$
  if $x^\circ_\theta + \alpha_m \cdot (\tilde{x}_\theta - x^\circ_\theta) \in \mathcal{C}_\theta$ **then**
    increase lower bound: $\alpha_l \leftarrow \alpha_m$
  else
    decrease upper bound: $\alpha_u \leftarrow \alpha_m$
  end if
end for
Return: $\hat{x}_\theta = \alpha_l \cdot (\tilde{x}_\theta - x^\circ_\theta) + x^\circ_\theta \in \mathcal{C}_\theta$

4.1 Bisection with Interior Points

Given an infeasible NN prediction $\tilde{x}_\theta \notin \mathcal{C}_\theta$ and an IP $x^\circ_\theta \in \mathcal{C}_\theta$, we "project" $\tilde{x}_\theta$ to $\mathcal{C}_\theta$ as:

$$\hat{x}_\theta = \mathrm{BP}(\tilde{x}_\theta, x^\circ_\theta) \triangleq \alpha^* \cdot (\tilde{x}_\theta - x^\circ_\theta) + x^\circ_\theta, \tag{2}$$

where $\alpha^* \in [0,1]$ leads to $\hat{x}_\theta \in \partial\mathcal{C}_\theta$ and $\partial\mathcal{C}_\theta$ is the boundary of $\mathcal{C}_\theta$. As depicted in Fig. 2, the "projected" solution $\hat{x}_\theta$ is located on the straight line segment connecting the infeasible solution $\tilde{x}_\theta$ and an IP x°_θ. We note that there could be multiple α^* and corresponding $\hat{x}_\theta$, given a pair of $\tilde{x}_\theta$ and x°_θ. To determine one such α^*, we employ the bisection method, as elaborated in Alg. 1. We initiate by drawing a straight line connecting $\tilde{x}_\theta$ with an IP x°_θ. This segment is guaranteed to intersect the boundary of the feasible region at least once. Subsequently, we apply the bisection algorithm to iteratively pinpoint **one** feasible solution along this segment toward the constraint boundary.

We **remark** on the applicability and efficiency of the bisection method as follows: (i) The bisection is applicable to general compact sets with non-empty interior as required in Assumption 1, beyond existing studies on convex or ball-homeomorphic sets; (ii) The bisection achieves a linear convergence rate, and each iteration is computationally light, primarily evaluating solution feasibility.

Further, the bisection method can be executed in batch for multiple interior points $X^\circ_{\theta,m} := \{x_{\theta,k}\}_{k=1}^m \subset \mathcal{C}_\theta$, and we select the projected point as the one with minimum deviation, defined as:

$$\hat{x}_\theta = \mathrm{BP}(\tilde{x}_\theta, X^\circ_{\theta,m}) \triangleq \arg\min_{\hat{x}_{\theta,k}}\{\|\hat{x}_{\theta,k} - \tilde{x}_\theta\|\}, \tag{3}$$

where $\hat{x}_{\theta,k} = \mathrm{BP}(\tilde{x}_\theta, x^\circ_{\theta,k})$ is the returned feasible point by bisection w.r.t. the k-th IP $x^\circ_{\theta,k} \in X^\circ_{\theta,m}$.

Despite the applicability and efficiency of the BP method, there are several critical challenges to applying it to recover infeasible NN solutions: (i) The projection distance induced by the bisection may be substantial if inappropriate interior points are chosen; (ii) Obtaining different IPs for input-dependent constraint sets in the online inference stage may still be time-consuming.

To address these two challenges, we first introduce the eccentricity of interior points and establish its connection to the bisection-induced projection distance in Sec. 4.2. We then employ another NN, called IPNN, to efficiently predict IPs in run-time in Sec. 4.3 and present the performance analysis for it in Sec. 5.

4.2 Minimum-Eccentricity IPs for Bisection

We first define the eccentricity of IPs, crucial for bounding the bisection-induced projection distance.

Definition 4.1 (Eccentricity of IPs)**.** For a compact set $\mathcal{X}$ satisfying Assumption 1 with non-empty interior, the eccentricity of a set of IPs $X_m^\circ := \{x_k^\circ\}_{k=1}^m \subset \mathcal{X}$ with respect to a compact subset of boundary $\Gamma \subseteq \partial\mathcal{X}$ is defined as:

$$\mathcal{E}(X_m^\circ, \Gamma) \triangleq \max_{y\in\Gamma} \|\mathrm{d}(y, X_m^\circ)\| - \min_{y\in\Gamma} \|\mathrm{d}(y, X_m^\circ)\|, \tag{4}$$

where $\mathrm{d}(y, X_m^\circ) = \min_{1\le k\le m}\{\|y - x_k^\circ\|\}$ is the point-to-set distance.

We make the following remarks regarding the eccentricity:

- When $m = 1$ and $\Gamma = \partial\mathcal{X}$, the eccentricity evaluates the gap between the largest and smallest point-to-boundary distances and defines the "centrality" of the IP. This concept is closely related to classic definitions of Chebyshev center and Incenter shown in Appendix B. However, it is uniquely designed to bound the worst-case bisection-induced projection distance under our setting.
- When $m > 1$ and $\Gamma = \partial\mathcal{X}$, the eccentricity is extended to multiple IPs. This can be viewed as the multiple IPs dividing the boundary into their nearest pieces and then evaluating the maximum and minimum point-to-boundary distances. Further, the minimum eccentricity, $\min_{X_m^\circ \subset \mathcal{X}} \mathcal{E}(X_m^\circ, \Gamma)$, decreases to zero as m increases, proven in Sec. 5.3.
- When $\Gamma \subset \partial\mathcal{X}$ is a subset of the boundary, it provides a local view of eccentricity in a region of interest, e.g., Γ is a local region of boundary containing all possible projected infeasible NN solutions with a bounded prediction error. As will become clear later, such a local view will shed light on efficient training in Sec. 4.3 and tight theoretical bounds in Sec. 5.3.

Next, we establish the connection between eccentricity and the bisection-induced projection distance.

Proposition 4.1. *Let $\tilde{x}_\theta = F(\theta)$ be an infeasible NN prediction with bounded prediction error as $\|F(\theta) - x_\theta^*\| \le \epsilon_{\mathrm{pre}}$; $\hat{x}_\theta = \mathrm{BP}(\tilde{x}_\theta, X_{\theta,m}^\circ)$ be the projected solution with m interior points $X_{\theta,m}^\circ \subset \mathcal{C}_\theta$; Then, the worst-case projection distance is upper bounded as:*

$$\max_{\tilde{x}_\theta \in \mathcal{B}(x_\theta^*, \epsilon_{\mathrm{pre}})} \|\tilde{x}_\theta - \mathrm{BP}(\tilde{x}_\theta, X_{\theta,m}^\circ)\| \le \epsilon_{\mathrm{pre}} + \mathcal{E}(X_{\theta,m}^\circ, \Gamma_\theta), \tag{5}$$

where $\mathcal{B}(x_\theta^*, \epsilon_{\mathrm{pre}})$ represents the NN prediction region, enclosing all infeasible NN predictions with prediction error ϵ_{pre}, and $\Gamma_\theta = \{\mathrm{BP}(\tilde{x}_\theta, X_{\theta,m}^\circ),\ \forall \tilde{x}_\theta \in \mathcal{B}(x_\theta^*, \epsilon_{\mathrm{pre}}) \text{ and } \tilde{x}_\theta \notin \mathcal{C}_\theta\}$ defines a subset of the constraint boundary containing all projected NN solutions from the NN infeasibility region. We call $\mathcal{E}(X_{\theta,m}^\circ, \Gamma_\theta)$ the eccentricity of $X_{\theta,m}^\circ$ modulated by the NN infeasibility region.

Informed by Prop. 4.1, we find those IPs with minimized eccentricity (MEIPs) modulated by the NN infeasibility region to directly bound the **worst-case** projection distance for infeasible NN prediction.

- When the initial prediction error is larger than the diameter of the constraint set, or we are only informed of the information about the constraint set, we seek to find IPs with minimized eccentricity with respect to the entire constraint boundary as $\Gamma_\theta = \partial\mathcal{C}_\theta$. Such MEIPs are prediction-agnostic and only depend on the geometry of $\mathcal{C}_\theta$.
- More typically, boosted by the universal approximation ability of the NN predictor, the initial prediction error is small. Γ_θ is only a local subset of the boundary. In this case, we find IPs with minimal eccentricity over such a local boundary modulated by the NN infeasibility region, avoiding the global and computationally intensive eccentricity computation.

Further, we note that the applicability of this bound applies to general compact sets under Assumption 1 beyond those in existing works, such as ball-homeomorphism (LCL23; LCL24). Thus, the MEIP-based bisection projection has wide applicability and achieves a non-trivial performance guarantee.

However, solving MEIPs is a challenging task due to the non-convexity, particularly when swift responses are essential for real-time decision-making. To address this issue, we propose a learning-based strategy in the subsequent section, which trains another NN (called IPNN) offline to predict the

IPs with low eccentricity given input parameters, mitigating the run-time complexities involved in finding IPs during online real-time applications.

4.3 Interior Points Neural Network (IPNN)

We train another feed-forward NN with ReLU activation, denoted as IPNN $\psi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{m \cdot n}$, to predict the IPs with low eccentricity modulated by the NN infeasibility region. The loss function to be minimized during training is designed as follows:

$$\mathcal{L}(\psi(\theta)) = \mathcal{P}(\psi(\theta)) + \lambda \cdot \hat{\mathcal{E}}(\psi(\theta), \Gamma_\theta) \cdot \mathbf{1}_{\psi(\theta) \subset \mathcal{C}_\theta}. \tag{6}$$

The first term is the penalty for constraint violation of IPNN outputs, i.e., a set of IPs for an input θ; The second term computes the eccentricity of valid IPs only, and $\mathbf{1}_{\psi(\theta) \subset \mathcal{C}_\theta}$ is the indicator function capturing whether the IPs outputted by IPNN is feasible or not. $\lambda > 0$ is a positive coefficient chosen based on empirical experience.

For the penalty term, we apply adversarial training techniques to keep the IPs away from the boundary, making them robust to IPNN prediction errors, as visualized in Fig. 1:

$$\mathcal{P}(\psi(\theta)) = \mathbb{E}_{z_1,\ldots,z_m}\left[\sum_{k=1}^{m} \|\mathrm{ReLU}(g(\psi_k(\theta) + z_k, \theta))\|\right], \tag{7}$$

where $z_1, \ldots, z_m \sim \mathcal{N}(0, \sigma^2 I)$ are independent Gaussian noise with variance σ^2, and the loss evaluates the constraint violation for Gaussian-perturbed IP predictions by $\psi(\theta)$. Such an adversarial loss has been widely used to enhance NN robustness (CRK19; LSF19). Under our setting, it helps to keep the IP predictions away from the constraint boundary, thus safeguarding the IP predictions under NN generalization errors after finite-sample training. We further present two sufficient conditions for IPNN to predict feasible IPs over arbitrary input after training with finite samples in Sec. 5.1.

For the non-smooth eccentricity term defined in (4), to enable efficient gradient-based optimization in training, we apply the following sample-based and smoothed eccentricity loss used in (6):

$$\hat{\mathcal{E}}(\psi(\theta), \Gamma_\theta) = \mathrm{LSE}_\beta(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b) - \mathrm{LSE}_{-\beta}(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b), \tag{8}$$

where $d_{\theta,j,k} = \|y_j - \psi_k(\theta)\|$ is the point-to-boundary distance for boundary samples $\{y_j\}_{j=1}^b \subset \Gamma_\theta$ and predicted IPs $\{\psi_k(\theta)\}_{k=1}^m$. The log-sum-exp operator is defined as $\mathrm{LSE}_\beta(\{x_i\}_{i=1}^n) := \frac{1}{\beta} \log \sum_{i=1}^n \exp(\beta x_i)$, which is a smooth maximum/minimum operator when β is positive/negative. The following proposition characterizes the approximation gap for the LSE-smoothed loss:

Proposition 4.2. *The sample-based eccentricity, defined as* $\overline{\mathcal{E}}(\psi(\theta), \Gamma_\theta) := \max\limits_{1 \le j \le b} \min\limits_{1 \le k \le m} d_{\theta,j,k} - \min\limits_{1 \le j \le b} \min\limits_{1 \le k \le m} d_{\theta,j,k}$, *with boundary samples* $\{y_j\}_{j=1}^b \subset \Gamma_\theta$, *can be well approximated by the LSE-smoothed eccentricity as* β *goes to infinity:*

$$\hat{\mathcal{E}}(\psi(\theta), \Gamma_\theta) - \log(mb^2)/\beta \le \overline{\mathcal{E}}(\psi(\theta), \Gamma_\theta) \le \hat{\mathcal{E}}(\psi(\theta), \Gamma_\theta) + \log(m)/\beta. \tag{9}$$

For efficient IPNN training, we need (i) samples from the input domain to evaluate the adversarial penalty term and minimize it to guide IPNN to find IPs; (ii) samples from the (local) constraint boundary Γ_θ containing all possible projected infeasible NN predictions with error ϵ_{pre} to evaluate the smoothed eccentricity. During training, the second eccentricity loss becomes active once IPNN outputs IPs under the first penalty loss. We derive boundary samples through bisection projection in Alg. 1 for infeasible solutions, constructed by adding noise to NN predictions. This **objective-aware** training focuses on the local region around the optimal solution, providing a smaller eccentricity and optimality gap. We also consider an **objective-agnostic** setting in Appendix D, optimizing the IPNN with eccentricity over the entire constraint boundary $\partial\mathcal{C}_\theta$. This approach can train one IPNN for multiple optimization problems with different objectives over the same constraints; however, it incurs higher complexity due to sampling the entire constraint boundary.

Finally, to optimize the average performance across different input parameters, we uniformly sample θ and minimize the total loss as $\mathcal{L}(\psi) = \mathbb{E}_\theta[\mathcal{L}(\psi(\theta))]$.

At the end of the training, we obtain an IPNN for predicting a set of IPs for an input θ. For the performance analysis later, we define the validness of the obtained IPNN as follows:

Definition 4.2 (Valid IPNN)**.** An IPNN is valid if it outputs feasible IPs for all the θ in the training set; Furthermore, it is **universally valid** if it can output feasible IPs for all the θ in the input region.

5 Performance Analysis

In this section, we present a comprehensive analysis of the BP framework: (i) the universally valid conditions for IPNN under varying inputs in Sec. 5.1 and (ii) the optimality loss and run-time complexity for the bisection operation in Sec. 5.2. We also discuss the training complexity and impacts of hyperparameters in Sec. 5.3 and reveal the connection to existing approaches (TZ22b; LCL23; THH23) in Appendix C.

5.1 Universally Valid Condition for IPNN

Theorem 1. *Suppose the trained IPNN, is valid (over the training dataset $\mathcal{D} = \{\theta_i\}_{i=1}^N \subseteq \Theta$), i.e., $\forall \theta_i \in \mathcal{D},\ \psi(\theta_i) \subset \mathcal{C}_{\theta_i}$. Then the IPNN is universally valid, i.e., it will output feasible IPs for any input θ in the compact input domain Θ, if one of the following conditions is met:*

(i) Sample-based Condition. Let dataset $\mathcal{D}$ be an r_c-covering set for Θ, i.e., $\forall \theta \in \Theta$, $\exists \theta_i \in \mathcal{D}$ such that $\|\theta - \theta_i\| \leq r_c$. Given the valid condition over training dataset $\mathcal{D}$, if $(C_0 + C_1) r_c \leq C_2$, then $\forall \theta \in \Theta,\ \psi(\theta) \subset \mathcal{C}_\theta$. Here $C_0 = \sup_{\theta_1, \theta_2 \in \Theta, \theta_1 \neq \theta_2} \{\frac{\mathrm{d_H}(\partial \mathcal{C}_{\theta_1}, \partial \mathcal{C}_{\theta_2})}{\|\theta_1 - \theta_2\|}\}$, $C_1 = \max_k \{\mathrm{Lip}(\psi_k(\cdot), \Theta)\}$ and $C_2 = \min_{k, \theta \in \mathcal{D}} \{\min_{z \in \partial \mathcal{C}_\theta} \{\|z - \psi_k(\theta)\|\}\}$. Detailed explanation in Appendix E.4.

(ii) Verification-based Condition. Consider calculating the worst-case constraint violation for the IPNN prediction $\psi(\cdot)$ over the input domain Θ, defined as $\{\max_{\theta \in \Theta, 1 \leq k \leq m} \|g(x_k, \theta)\|_\infty,\ \text{s.t. } x_k = \psi_k(\theta)\}$. If an upper bound of the optimal objective value is non-positive, then $\forall \theta \in \Theta,\ \psi(\theta) \subset \mathcal{C}_\theta$.

Theorem 1 establishes sufficient conditions for the trained IPNN to be universally valid, a premise for the subsequent bisection applied to infeasible predictions under unseen input parameters. First, the IPNN needs to be valid, i.e., predict feasible IPs over finite training samples. Empirical studies in Sec. 6 show that this condition is easily achieved under the designed adversarial penalty loss in (6), as the constraint violation is minimized for perturbed IP predictions. To generalize the valid condition to any input parameter $\theta \in \Theta$, the following sufficient conditions are provided:

The **Sample-based** condition indicates that a smaller covering radius r_c is needed for "thin" constraint sets (small C_2), highly variable constraint geometries (large C_0), and IPNN with large Lipschitz constants (large C_1). Additionally, Assumption 1 also ensures $C_2 > 0$ for a "fat" set. A small covering radius implies a larger number of training samples (N), scaling as $\mathcal{O}((\mathrm{diam}(\Theta)/r_c)^d)$, to cover the input space for a universally feasible IPNN. However, C_0 may be unbounded for constraint sets that change discontinuously, and obtaining closed-form expressions for these constants may be computationally challenging.

The **Verification-based** condition aims to compute the worst-case constraint violation (or its upper bound) to verify the IPNN's feasibility (QOB+19). Exact verification can be NP-hard due to ReLU activation or non-convex constraint functions. However, an upper bound of the constraint violation, obtained by employing convex relaxation for the activation function (TXT17; RSL18) and the non-convex constraint function (LNDT19; NS06), can be used as a sufficient condition for verification. This reduces the verification problem to a convex programming task, which can be solved efficiently. However, the relaxation techniques are problem-dependent and may not be universal for arbitrary constraints. More details and examples are provided in Appendix E.5 and F.2.

5.2 Optimality and Run-time Complexity for Bisection

Theorem 2. *Given constraint set $\mathcal{C}_\theta$ under Assumption 1, an infeasible NN prediction $\tilde{x}_\theta$ with bounded error to the optimal solution x_θ^* as $\|\tilde{x}_\theta - x_\theta^*\| \leq \epsilon_{\text{pre}}$, and multiple IPs $X_{\theta,m}^\circ$ produced by a universally-valid IPNN, after executing K steps of bisection shown in Alg. 1. We obtain a solution $\hat{x}_\theta^K$ satisfying the following:*

(i) it is guaranteed to be feasible, i.e., $\hat{x}_\theta^K \in \mathcal{C}_\theta$;

(ii) it has a bounded optimality gap as $\|\hat{x}_\theta^K - x_\theta^*\| \leq 2\epsilon_{\text{pre}} + \mathcal{E}(X_{\theta,m}^\circ, \Gamma_\theta) + 2^{-K}(\epsilon_{\text{pre}} + \mathrm{diam}(\mathcal{C}_\theta)) + \epsilon_g$, where $\epsilon_g = 0$ if $m = 1$ or $K \geq \log_2(C_3)$ (ϵ_g and C_3 are constants detailed in Appendix E.6);

(iii) the run-time complexity is $\mathcal{O}(mKG)$, where G is the complexity of checking if a given solution satisfies the constraints of the optimization problem.

First, given multiple IPs, The bisection in Alg. 1 consistently returns a feasible solution. The optimality loss of the returned feasible solution is mainly bounded by three factors: the initial NN prediction error for the optimal solution, the eccentricity measure of predicted IPs, and the error due to finite-step bisection. (i) The initial prediction error is typically small, thanks to the NN's universal approximation capabilities. (ii) The eccentricity measure represents the upper bound of the maximum deviation caused by employing bisection with multiple IPs, which should be minimized during IPNN training, as highlighted by the loss function in (6). (iii) The error from finite bisection decreases exponentially with each additional iteration after a small threshold.

The algorithm's run-time complexity, i.e., the number of arithmetic operations, is primarily affected by the number of bisection steps (K) and the complexity of verifying inequality constraints at each step (G). Notably, in practical applications, the run-time escalates linearly with the number of bisection steps but not with the number of IPs by employing batch processing.

5.3 Discussions

IPNN training complexity. As discussed in Sec. 4.3, IPNN training requires sampling over (local) boundary Γ_θ to evaluate the eccentricity. Although accurately approximating the eccentricity over general compact sets in the worst case requires a covering dataset (Zho02), we use stochastic training by sampling a small batch of boundary points at each iteration and optimizing the smoothed eccentricity loss. Our numerical studies show that this approach guides IPNN in finding IPs with low eccentricity, leading to minor bisection-induced optimality loss.

IPNN sampling complexity. To obtain boundary samples, we apply the bisection projection in Alg. 1 with runtime complexity $\mathcal{O}(KG)$, where K is the number of bisection steps and G is the complexity of checking constraint satisfaction. This algorithm provides linear convergence to find boundary samples and can be applied to a batch of infeasible predictions, enabling low-complexity IPNN training, as demonstrated in Sec. 6.2. The detailed pseudocode for training and sampling is provided in Appendix G.3.

IPNN convergence. Despite the empirical success of training the IPNN to be valid over the training dataset, a premise on the generalization condition in Theorem 1, we remark that the theoretical convergence guarantee on IPNN training is non-trivial, similar to regular NN training, which largely depends on the optimizer, data coverage, and NN initialization, and requires future exploration.

Impact of number of IPs. As we used multiple IPs to reduce the bisection-induced projection distance, we analyzed the trade-off between the projection distance and the number of used IPs.

Proposition 5.1. *Given a compact set $\mathcal{C}_\theta$ for input θ under Assumption 1, the minimum eccentricity modulated by the NN infeasibility region for m IPs is upper bounded as:* $\min_{X^\circ_{\theta,m}\subset\mathcal{C}_\theta}\mathcal{E}(X^\circ_{\theta,m},\Gamma_\theta)\leq\min\{\epsilon_{\text{pre}},\mathcal{O}\left(m^{-1/(n-1)}\right)\}$*, where we recall n is the number of decision variables.*

We provide constructive proof for Prop. 5.1 in Appendix E.3. The developed upper bound is tight for general compact sets, as the eccentricity of multiple IPs is connected to the covering complexity (KZLD21), where the complexity order is consistent for compact sets under different inputs. Since we focus on eccentricity modulated by the NN infeasibility region, it can be directly bounded by the corresponding NN prediction error. The second term in the upper bound indicates that the minimum eccentricity monotonically decreases as the number of IPs increases. This proposition justifies the design of (i) employing multiple IPs in projection, compared to using only one IP as in previous studies (discussed in Appendix C), and (ii) focusing on eccentricity modulated by the NN infeasibility region instead of eccentricity over the entire constraint set. Although empirical experiments suggest that 1-2 IPs lead to minor optimality loss, the exact lower bound on the number of IPs remains largely open and warrants future exploration.

6 Numerical Experiments

We carry out simulations to (i) visualize and comprehend the proposed MEIPs and IPNN training in the 2-dimensional non-convex constraint set, as detailed in Appendix F.1; (ii) validate the performance of BP against existing state-of-the-art methods on various constrained optimization problems in Sec. 6.1; (iii) corroborate the efficacy of key components and parameters in the BP framework through sensitivity analysis in Sec. 6.2; (iv) demonstrate the verification condition for IPNN feasibility over convex and non-convex sets, as elaborated in Appendix F.2. The problem formulation and hyperparameter setting are provided in Appendix G.

6.1 Ensuring NN solution feasibility for Constrained Optimization Problems

Table 2: Performance comparison for four constrained optimization problems.

	Feasibility			**Optimality**				**Speedup**	
	feasible rate (%)	inequality vio. (1-norm)	equality vio. (1-norm)	solution error ave. (%)	solution error cor. (%)	objective gap ave. (%)	objective gap cor. (%)	NN+post. ave. (×)	NN+post. cor. (×)
	Convex QCQP: $n = 400,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
NN	80.08	0.502	0	9.09	8.99	6.03	4.73	964243	–
NN+WS	100	0	0	7.3	0	5.09	0	5.2	1
NN+Proj	100	0	0	9.09	8.98	6.04	4.75	5.8	1.2
NN+D-Proj	84.28	0.079	0	9.08	8.97	6.04	4.74	37.6	7.5
NN+H-Proj	100	0	0	9.18	9.45	6.24	5.75	4128	824
NN+B-Proj	100	0	0	9.15	9.32	6.23	5.71	54236	11449
	SOCP: $n = 400,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$								
NN	83.79	0.69	0	17.41	17.26	6.46	5.38	698523	–
NN+WS	100	0	0	14.62	0	5.59	0	6.1	1
NN+Proj	100	0	0	17.41	17.25	6.48	5.5	7.5	1.2
NN+D-Proj	93.36	0.003	0	17.43	17.37	6.47	5.45	34	5.5
NN+H-Proj	100	0	0	17.43	17.38	6.61	6.28	4025	655
NN+B-Proj	100	0	0	17.44	17.39	6.60	6.23	168708	36058
	JCC-IM: : $n = 400,\ d = 100,\ n_{\text{eq}} = 0,\ n_{\text{ineq}} = 10100$								
NN	77.15	0.03	0	1.71	1.74	1.17	1.18	551239	–
NN+WS	100	0	0	1.32	0	0.9	0	7.7	1.8
NN+Proj	100	0	0	1.7	1.74	1.17	1.8	1	0.1
NN+D-Proj	78.52	0.01	0	1.71	1.74	1.17	1.18	73	1.7
NN+H-Proj	100	0	0	4.34	13.24	4.02	13.64	378	86
NN+B-Proj	100	0	0	1.84	2.3	1.33	1.87	857	196
	AC-OPF: $n = 476,\ d = 400,\ n_{\text{eq}} = 400,\ n_{\text{ineq}} = 1042$								
NN	82.81	0.001	0	0.32	0.27	0.05	0.05	631	–
NN+WS	100	0	0	0.28	0	0.04	0	18.5	1.3
NN+Proj	100	0	0	0.6	2.62	0.05	0.07	4.3	0.5
NN+D-Proj	84.28	0.001	0	0.32	0.27	0.05	0.05	11.4	2
NN+H-Proj	100	0	0	0.41	0.42	0.08	0.15	51	15
NN+B-Proj	100	0	0	0.34	0.38	0.06	0.11	258	52

[1] d and n represent the dimensions for input parameter θ and output decision x, respectively. n_{eq} and n_{ineq} denote the number of equality and inequality constraints, respectively.

[2] All post-processing approaches are adopted for infeasible NN predictions only. Those metrics are separately evaluated for all predictions (ave. metric) and corrected infeasible predictions (cor. metric).

[3] For the joint chance constraint (JCC) without tractable reformulation, we employ the scenario-based approach to generate the approximated ground truth for NN training. Consequently, the inequality grows linearly with sampled scenarios (PAS09). In the absence of an analytical expression for the JCC, the feasibility metric is evaluated over i.i.d samples.

[4] We **remark** on the complicated constraint geometry for two non-convex problems, as visualized in Fig. 9 and 10.

We apply the BP framework to two benchmark convex cases (convex QCQP, and SOCP) and two non-convex real-world scenarios, including AC-OPF problems in grid operation and joint chance-constrained problems in inventory management (JCCIM). We first train an NN predictor to learn the mapping from input parameters to the optimal solutions in existing works (DRK20), where the training and testing data are generated by randomly sampling the input parameter and solve the corresponding optimal solutions through iterative solvers as ground truth (DRK20; LCL23). For convex optimization, we use MOSEK to solve the optimal solution. For AC-OPF problems, we adopt PYPOWER as the specialized solver (ZMSG97).

To ensure the feasibility of equality constraints, we utilize predict-then-reconstruct techniques (PZCZ20; DRK20), as detailed in Appendix A.1. For infeasible NN predictions with respect to the inequality constraints, we compare our BP framework with different post-processing approaches: (i) solver-based approaches, such as orthogonal projection (Proj.) and warm-start methods (WS.), (ii) gradient-descent methods (D-Proj.) (DRK20), and (iii) homeomorphic projection (H-Proj.) (LCL23). Note that some baselines shown in Table 1 are not included due to their limited applicability, and the details of those compared baselines are provided in Appendix G.1.

The performances over four constraint optimization problems are shown in Table 2. We have the following observations for the experiment results. The BP framework achieves **100%** feasibility rate for infeasible NN predictions, with the best speedup and similar optimality loss. Solver-based methods like warm-start or projection ensure feasibility and minimal optimality loss but suffer from poor speeds compared to direct problem-solving. The gradient-based D-Proj method, while versatile for different constraint sets, does not guarantee feasibility and is highly sensitive to step size choices. H-Proj also achieves 100% feasibility with similar optimality loss but is less efficient than B-Proj, especially in such high-dimensional scenarios, due to the demanding invertible NN calculations required at each bisection step. Overall, for those convex and non-convex constraint sets, BP outperforms existing methods in feasibility or run-time complexity with similar optimality loss.

6.2 Sensitivity Analysis for Bisection Projection Framework

This section explores the impact of key components and parameters in the BP framework to validate its design, conducting comparisons across convex and non-convex problems. We investigate (i) the impact of eccentricity on the bisection-induced optimality gap, and (ii) the relationship between the number of employed IPs and the optimality loss.

As illustrated in Table 3, in alignment with our observations in Sec. 6.1, the IPNN predicts valid IPs across both constraint sets under unseen testing parameters, substantiating the design of the adversarial penalty loss. By minimizing the eccentricity, the ME-IPNN identifies better IPs for conducting bisection, resulting in a minor projection distance. It is noteworthy that ME-IPNN with a single IP surpasses the performance of IPNN with 8 IP predictors, corroborating the efficiency of the eccentricity loss design. Moreover, as the number of IPs increases, the bisection-induced projection distance decreases, which aligns with Prop. 5.1.

Table 3: Sensitivity analysis for BP framework.

	Feasibility feas. rate (%)	ineq. vio. (1-norm)	**Optimality** sol. err. (%)	obj. gap (%)	**Run-time** total (s)	**Training** per iteration (s)
	Convex QCQP: : $n = 400,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 100$					
NN Solution Predictor	0.00%	11.05	3.10%	3.03%	0.001	0.006
H-Proj	100%	0.00	6.02%	3.99%	1.13	0.01
IPNN 1	100%	0.00	8.55%	6.67%	0.13	0.007
IPNN 2	100%	0.00	6.23%	4.15%	0.27	0.006
IPNN 4	100%	0.00	5.62%	3.66%	0.65	0.008
IPNN 8	100%	0.00	5.32%	3.43%	0.82	0.01
ME-IPNN 1	100%	0.00	5.18%	3.63%	0.10	0.02
ME-IPNN 2	100%	0.00	5.01%	3.46%	0.34	0.02
ME-IPNN 4	100%	0.00	4.87%	3.30%	0.39	0.03
ME-IPNN 8	100%	0.00	4.80%	3.28%	0.65	0.03
	Non-convex JCC-IM: : $n = 400,\ d = 100,\ n_{\text{eq}} = 0,\ n_{\text{ineq}} = 10100$					
NN Solution Predictor	6.54%	0.18	2.68%	2.66%	0.001	0.06
H-Proj	100%	0.00	8.39%	10.99%	1.10	0.07
IPNN 1	100%	0.00	6.22%	5.78%	0.39	0.08
IPNN 2	100%	0.00	5.47%	4.94%	0.46	0.07
IPNN 4	100%	0.00	5.21%	4.65%	0.48	0.07
IPNN 8	100%	0.00	5.01%	4.43%	0.62	0.07
ME-IPNN 1	100%	0.00	4.74%	4.20%	0.42	0.77
ME-IPNN 2	100%	0.00	4.29%	3.74%	0.41	0.71
ME-IPNN 4	100%	0.00	3.91%	3.24%	0.50	0.69
ME-IPNN 8	100%	0.00	3.68%	2.97%	0.54	0.70

[1] We train an NN predictor without penalty loss to obtain an approximate solution to the optimization problem (with substantial constraint violation). We then feed those infeasible NN predictions to different IPNNs to recover a feasible solution.
[2] IPNN indicates that it is trained solely with penalty terms for constraint violation. ME-IPNN signifies that it is jointly trained with penalty terms and eccentricity measures.
[3] Run time denotes the time required for executing the bisection algorithm to recover feasibility for infeasible cases out of 1,024 test instances.
[4] Training time represents the average per-iteration training cost, encompassing boundary sampling time and forward-backward propagation.

7 Conclusion and Limitation

We introduce the *Bisection Projection* framework, an efficient scheme to recover feasible solutions from infeasible NN predictions over a general compact constraint set through simple bisection operation and with minor optimality loss. Extensive experiments in this paper demonstrate the general applicability and efficiency of the approach beyond existing methods.

Meanwhile, there are several limitations of the framework that can be viewed as important future directions, as discussed in Sec. 5.3. (i) In addition to the covering-based upper bound in Prop. 5.1, the lower bound of the number of required IPs for specific sets is useful and subject to further investigation. (ii) The convergence guarantee for training IPNN to learn feasible solutions over the training dataset under the overparameterized NN framework will also be an interesting future direction. (iii) The two sufficient conditions in Theorem 1 do not universally work for arbitrary constraint sets; more general and computationally efficient conditions should be explored in the future.

References

[AAB+19] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. Advances in neural information processing systems, 32, 2019.

[Aba69] Jean Abadie. Generalization of the wolfe reduced gradient method to the case of nonlinear constraints. Optimization, pages 37–47, 1969.

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

[Amo22] Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. arXiv preprint arXiv:2202.00665, 2022.

[Bak19] Kyri Baker. Learning warm-start points for ac optimal power flow. In 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), 2019.

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

[CCC+21] 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, 2021.

[CDB+21] Bingqing Chen, Priya L Donti, Kyri Baker, J Zico Kolter, and Mario Bergés. Enforcing policy feasibility constraints through differentiable projection for energy optimization. In Proceedings of the Twelfth ACM International Conference on Future Energy Systems, pages 199–210, 2021.

[COMB19] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 3387–3395, 2019.

[CRK19] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In international conference on machine learning, pages 1310–1320. PMLR, 2019.

[Die19] 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.

[DRK20] Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. In International Conference on Learning Representations, 2020.

[DWDS23] Shutong Ding, Jingya Wang, Yali Du, and Ye Shi. Reduced policy optimization for continuous control with hard constraints. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[FMP20] 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.

[FMVH20] 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.

[FNC20] Thomas Frerix, Matthias Nießner, and Daniel Cremers. Homogeneous linear inequality constraints for neural network activations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 748–749, 2020.

[GWWM19] Neel Guha, Zhecheng Wang, Matt Wytock, and Arun Majumdar. Machine learning for ac optimal power flow. arXiv preprint arXiv:1910.08842, 2019.

[HFL+22] 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.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

[HWFGY21] 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.

[KFVHW21] James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder. End-to-end constrained optimization learning: A survey. arXiv preprint arXiv:2103.16378, 2021.

[KU23] Andrei V Konstantinov and Lev V Utkin. A new computationally simple approach for implementing neural networks with output hard constraints. In Doklady Mathematics, volume 108, pages S233–S241. Springer, 2023.

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

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

[LCL23] Enming Liang, Minghua Chen, and Steven H. Low. Low complexity homeomorphic projection to ensure neural-network solution feasibility for optimization over (non-)convex set. In International Conference on Machine Learning. PMLR, 2023.

[LCL24] Enming Liang, Minghua Chen, and Steven H. Low. Homeomorphic projection to ensure neural-network solution feasibility for constrained optimization. Journal of Machine Learning Rsearch, 2024.

[Lee13] John M Lee. Smooth manifolds. In Introduction to smooth manifolds, pages 1–31. Springer, 2013.

[Lew88] John L Lewis. Uniformly fat sets. Transactions of the American Mathematical Society, 308(1):177–196, 1988.

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

[LLPS93] 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.

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

[LNDT19] Dongchan Lee, Hung D Nguyen, Krishnamurthy Dvijotham, and Konstantin Turitsyn. Convex restriction of power flow feasibility sets. IEEE Transactions on Control of Network Systems, 6(3):1235–1245, 2019.

[LSF19] Alexander Levine, Sahil Singla, and Soheil Feizi. Certifiably robust interpretation in deep learning. arXiv preprint arXiv:1905.12105, 2019.

[MH+19] Daniel K Molzahn, Ian A Hiskens, et al. A survey of relaxations and approximations of the power flow equations. 2019.

[NC21a] Rahul Nellikkath and Spyros Chatzivasileiadis. Physics-informed neural networks for ac optimal power flow. arXiv preprint arXiv:2110.02672, 2021.

[NC21b] Rahul Nellikkath and Spyros Chatzivasileiadis. Physics-informed neural networks for minimising worst-case violations in dc optimal power flow. In 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pages 419–424. IEEE, 2021.

[NS06] Arkadi Nemirovski and Alexander Shapiro. Scenario approximations of chance constraints. Probabilistic and randomized methods for design under uncertainty, pages 3–47, 2006.

[PAS09] Bernardo K Pagnoncelli, Shabbir Ahmed, and Alexander Shapiro. Sample average approximation method for chance constrained programming: theory and applications. Journal of optimization theory and applications, 142(2):399–416, 2009.

[PCZL22] Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H Low. Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems. IEEE Systems Journal, pages 42–47, 2022.

[PZC19] 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.

[PZCZ20] Xiang Pan, Tianyu Zhao, Minghua Chen, and Shengyu Zhang. Deepopf: A deep neural network approach for security-constrained dc optimal power flow. IEEE Transactions on Power Systems, 36(3):1725–1735, 2020.

[QOB+19] Chongli Qin, Brendan O'Donoghue, Rudy Bunel, Robert Stanforth, Sven Gowal, Jonathan Uesato, Grzegorz Swirszcz, Pushmeet Kohli, et al. Verification of non-linear specifications for neural networks. arXiv preprint arXiv:1902.09592, 2019.

[RSL18] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. Advances in neural information processing systems, 31, 2018.

[SHAS23] 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.

[SHAS24] 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.

[SJK+24] Zhouxing Shi, Qirui Jin, Zico Kolter, Suman Jana, Cho-Jui Hsieh, and Huan Zhang. Neural network verification with branch-and-bound for general nonlinearities. arXiv preprint arXiv:2405.21063, 2024.

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

[TXT17] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. arXiv preprint arXiv:1711.07356, 2017.

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

[TZ22b] Daniel Tabas and Baosen Zhang. Safe and efficient model predictive control using neural networks: An interior point approach. In 2022 IEEE 61st Conference on Decision and Control (CDC), pages 1142–1147. IEEE, 2022.

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

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

[WZG+23] Runzhong Wang, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang, and Junchi Yan. Linsatnet: the positive linear satisfiability neural networks. In International Conference on Machine Learning, pages 36605–36625. PMLR, 2023.

[WZX+21] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. Advances in Neural Information Processing Systems, 34, 2021.

[ZB20] 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.

[ZCZ21] 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.

[Zho02] Ding-Xuan Zhou. The covering number in learning theory. Journal of Complexity, 18(3):739–767, 2002.

[ZMS11] Ray D Zimmerman and Carlos E Murillo-Sanchez. Matpower 4.1 user's manual. Power Systems Engineering Research Center (PSERC), 20, 2011.

[ZMSG97] Ray D Zimmerman, Carlos E Murillo-Sánchez, and Deqiang Gan. Matpower. PSERC.[Online]. Software Available at: http://www. pserc. cornell. edu/matpower, 1997.

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

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

[ZSRZ21] Liyuan Zheng, Yuanyuan Shi, Lillian J Ratliff, and Baosen Zhang. Safe reinforcement learning of control-affine systems with vertex networks. In Learning for Dynamics and Control, pages 336–347. PMLR, 2021.

Appendices

Contents

A Assumptions of Constraint Sets

A.1 Equality Constraint

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

$$\mathcal{C}_\theta = \{x \in \mathbb{R}^n \mid h(x,\theta) = 0,\ g(x,\theta) \leq 0\}, \tag{10}$$

where the functions $h(\cdot,\cdot) : \mathbb{R}^{n+d} \to \mathbb{R}^{n_{\text{eq}}}$ and $g(\cdot,\cdot) : \mathbb{R}^{n+d} \to \mathbb{R}^{n_{\text{ineq}}}$ are continuous with respect to x and θ. For simplicity, we use $h_\theta(\cdot) = h(\cdot,\theta)$.

Assuming the equality constraint maintains a constant rank:

$$\text{rank}(\mathrm{J}_{h_\theta}(x)) = r, \quad \forall\theta \in \Theta \text{ and } \forall x \in \mathcal{C}_\theta, \tag{11}$$

this condition implies that $\mathcal{C}_\theta$ has a Euclidean dimension[1] of $n - r$, as per the *Constant-Rank Level Set Theorem* (Lee13).

In simpler terms, we can utilize a subset of decision variables $x_1 \in \mathbb{R}^{n-r}$ and reconstruct the complete set of decision variables $[x_1, x_2] \in \mathbb{R}^n$ by solving $x_2 = \phi_\theta(x_1)$, such that $h_\theta([x_1, \phi_\theta(x_1)]) = 0$. Note that such a parametrization are not necessarily held globally for non-linear equality constraints. This method of reconstruction, which ensures the feasibility of the equality constraint, is extensively used in optimization literature (Aba69; PZC19; ZB20; DRK20; LCL23; THH23; LM23; DWDS23).

We then denote the reduced constraint set as

$$\mathcal{C}_\theta^s = \{x \in \mathbb{R}^{n-r} \mid g([x_1, \phi_\theta(x_1)], \theta) \leq 0\} \tag{12}$$

[1] 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 $\mathbb{R}^n$ (Lee13).

This set $\mathcal{C}_\theta^s$ is not only equivalent to the original constraint set $\mathcal{C}_\theta$ but also homeomorphic to it, implying a one-to-one, continuous, and bicontinuous correspondence between the two sets. The forward and inverse mappings of this homeomorphism are described by the following transformations:

$$[x_1, x_2] \in \mathcal{C}_\theta \to x_1 \in \mathcal{C}_\theta^s, \tag{13}$$
$$x_1 \in \mathcal{C}_\theta^s \to [x_1, \phi_\theta(x_1)] \in \mathcal{C}_\theta. \tag{14}$$

Let's consider two examples to illustrate this equality completion/reconstruction process:

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 $\text{rank}(A) = r$.

To facilitate the reconstruction process, we partition the decision variable x into two groups: $x_1 \in \mathbb{R}^{n-r}$ and $x_2 \in \mathbb{R}^r$. Accordingly, we also partition matrix A into $A = [A_1, A_2]$, where $A_1 \in \mathbb{R}^{r\times(n-r)}$ and $A_2 \in \mathbb{R}^{r\times r}$. Hence, the equality constraint can be represented as $A_1x_1 + A_2x_2 = \theta$. The reconstruction process indicates that we can determine x_2 using only the subset of variables x_1, with the explicit relationship given by:

$$x_2 = \phi_\theta(x_1) = A_2^{-1}(\theta - A_1x_1). \tag{15}$$

Here, we choose the partition of x_1 and x_2 such that A_2 has the full rank of r.

The relevant Jacobian matrix for back-propagation in this context is:

$$\mathrm{J}_{\phi_\theta}(x_1) = -A_2^{-1}A_1. \tag{16}$$

Non-linear equality constraint

For a non-linear equality constraint defined as $\{x \in \mathbb{R}^n \mid h(x,\theta) = 0, \theta \in \mathbb{R}^d, h : \mathbb{R}^{n+d} \to \mathbb{R}^r\}$, we partition the decision variable into $x_1 \in \mathbb{R}^{n-r}$ and $x_2 \in \mathbb{R}^r$ in a similar fashion to the linear case. Under the assumption that the Jacobian matrix of h with respect to x_2 has a constant rank, the completion function ϕ_θ is well-defined and satisfies:

$$h([x_1, \phi_\theta(x_1)], \theta) = 0. \tag{17}$$

To solve for $\phi_\theta(x_1)$ when h is non-linear, we can employ an iterative technique such as Newton's method. The necessary Jacobian matrix for back-propagation can be computed using the *Implicit Function Theorem*, which provides the derivative of the implicitly defined function ϕ_θ. The Jacobian matrix is given by:

$$\mathrm{J}_{\phi_\theta}(x_1) = -\,\mathrm{J}_{h_\theta}^{-1}(x_2)\,\mathrm{J}_{h_\theta}(x_1). \tag{18}$$

Note that ϕ_θ for such a non-linear constraint may not be single-valued globally and depends on the initial value for the iterative algorithm, which may bring potential convergence issues.

In conclusion, reconstruction techniques utilizing equality constraints allow for a reduction in the dimensionality of the decision variable space. By modeling only a subset of the decision variables, we can focus on the inequality constraints and use the equality constraints to implicitly define the remaining variables. This process is differentiable, making it suitable for integration into the training of machine learning models, hence providing a powerful tool for incorporating equality constraints into such models (Aba69; PZC19; PCZL22; DRK20; DWDS23).

A.2 Inequality Constraint

After exploiting and eliminating the equality constraints without compromising optimality, we introduce Assumption 1 pertaining to the constraint set. It is important to note that this assumption is made to circumvent extreme cases in compact set spaces, such as discrete constraint sets and certain continuous ones, which may limit the applicability of our framework.

As illustrated in Figure 3, the constraint set $\mathcal{C}_\theta$, represented by the green curve, resides in a two-dimensional space. However, it possesses a one-dimensional Euclidean structure and is measure-zero (zero volume) in $\mathbb{R}^2$. When applying the bisection projection over this compact constraint set, the

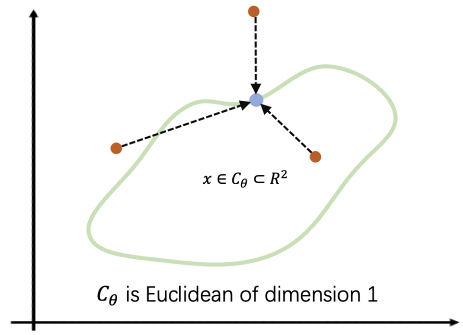


Figure 3: An extreme case for continuous constraint set (green curve).

feasibility of the projected solution is guaranteed. Nevertheless, nearly all projected solutions will converge to the given feasible point. Even when conducting bisections with multiple points, all infeasible points will be projected back to their nearest predefined feasible points. Under such circumstances, the number of required IPs reaches the upper bound established in Prop. 5.1. In other words, we require the exact covering dataset of IPs to achieve the worst-case performance guarantee. In this context, our methods reduce to the sampling-based techniques presented in (KZLD21), which, although theoretically feasible, are less practical.

To circumvent such measure-zero constraint sets, which typically arise from equality constraints, we first employ the technique described in A.1 to remove them and embed the high-dimensional measure-zero constraint set into a non-measure-zero subspace. Furthermore, to avoid certain extreme cases under inequality constraints, we introduce Assumption 1 to further refine the constraint set of interest.

B Comparison of Different Centers

Defining the centers of a set is a classic problem in mathematics, which involves various definitions tailored to serve specific purposes. Each definition captures a unique aspect of "centrality" depending on the application or theoretical requirements. Here, as shown in Table 4, we discuss several classic definitions including the proposed minimum-eccentricity interior point (MEIP) in our work.

Table 4: Comparison of different definitions of center for a set

Name	Definition	Description
MEIP	$x^\circ = \arg\min_{x\in\mathcal{X}} \left(\max_{y\in\partial\mathcal{X}} \|x-y\| - \min_{y\in\partial\mathcal{X}} \|x-y\| \right)$	Minimizes the discrepancy between the maximum and minimum distances from the point to the boundary of the set.
Chebyshev Center	$x^\circ = \arg\min_{x\in\mathcal{X}} \max_{y\in\partial\mathcal{X}} \|x-y\|$	Minimizes the maximum distance from the point to the boundary of the set.
Incenter	$x^\circ = \arg\max_{x\in\mathcal{X}} \min_{y\in\partial\mathcal{X}} \|x-y\|$	Maximizes the minimum distance from the point to the boundary of the set.
Analytical Center	$x^\circ = \arg\max_{x\in\mathcal{X}} \left(\sum_{i=1}^{n_{\text{ineq}}} \log(-g_i(x))\right)$	Maximizes the logarithmic barrier of the inequality residuals ($g_i(x) \le 0$), ensuring the point is centrally located within the feasible region.
Centroid	$x^\circ = \frac{1}{n}\sum_{i=1}^{n} x_i$	Calculates the average position of all points in the set, often used for geometric shapes and spatial data.
Barycenter	$x^\circ = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$	Calculates the weighted average position of all points in the set, where each point has an associated weight w_i.

- The MEIP, Chebyshev Center, and Incenter focus on geometric properties of sets, specifically distances to the boundary. The MEIP minimizes the disparity between the maximum and minimum distances to the boundary. The Chebyshev Center minimizes the maximum distance to the boundary, while the Incenter maximizes the minimum distance. We propose the MEIP for bisection operation, justified by the performance guarantee in Proposition 4.1.

- The Analytical Center uses optimization techniques, particularly barrier methods, to maintain a central position within a feasible region defined by constraints. This approach is crucial for solving linear and nonlinear programming problems. However, maximizing the log-residual of the inequality function does not directly reflect the point-to-boundary distance for general constraint sets, which may result in a large deviation for the bisection operation.
- The Centroid and Barycenter represent statistical approaches to defining centrality by calculating averages of point sets. The Centroid computes a simple arithmetic mean, suitable for applications in statistics and machine learning. The Barycenter incorporates weights, allowing for differentiated influence among points. Despite their simplicity, these centers are not necessarily interior points for general non-convex sets $\mathcal{X}$, making them unsuitable for our bisection operation.

C Connection to Existing Works

As discussed in Sec. 2, the proposed framework is conceptually related to the homeomorphic projection (LCL23; LCL24) and gauge function based methods (TZ22b; THH23).

We first introduce the basic definitions for gauge function and homeomorphic projection.

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

The Gauge/Minkowski function is a generalized definition of the norm function. For example, if the compact convex set is a p-norm ball, the gauge function is the p-norm as: $\varphi_{\mathcal{B}}(x, 0) = \|x\|_p$. It satisfies the properties of the norm function, such as non-negative, positively homogeneous, and sub-additive. Thus, the Gauge function is also convex for x.

In a simple manner, the gauge function is the ratio between the norm of x and the IP-to-boundary distance along direction $x - x^\circ$, i.e.,

$$\varphi_{\mathcal{C}}(x, x^\circ) \begin{cases} = 0, & x = 0 \\ = 1, & x \in \partial(\mathcal{C} - x^\circ) \\ > 1, & x \notin \mathcal{C} - x^\circ \end{cases} \tag{19}$$

Based on the gauge function, we can construct the following bijection between two compact convex sets:

Definition C.2 (Gauge mapping (TZ22b))**.** The gauge mapping between two convex sets $\mathcal{Z}$ and $\mathcal{C}$ is defined as $\Phi(z) = \frac{\varphi_{\mathcal{Z}}(z - z^\circ, z^\circ)}{\varphi_{\mathcal{C}}(z - z^\circ, x^\circ)}(z - z^\circ) + x^\circ$ for $z \in \mathcal{Z}$ or equivalentlx $\Phi^{-1}(x) = \frac{\varphi_{\mathcal{C}}(x - x^\circ, x^\circ)}{\varphi_{\mathcal{Z}}(x - x^\circ, z^\circ)}(x - x^\circ) + z^\circ$ for $x \in \mathcal{C}$.

The gauge mapping is also a homeomorphic mapping between two convex sets such that $\mathcal{C} = \Phi(\mathcal{Z})$ and $\mathcal{Z} = \Phi^{-1}(\mathcal{C})$ (TZ22b).

Definition C.3 (Homeomorphic projection (LCL23))**.** For ball-homeomorphic set $\mathcal{K}$, consider a homeomorphic mapping between $\mathcal{K}$ and $\mathcal{B}$ such that $\mathcal{K} = \Phi(\mathcal{B})$. For an infeasible point $\tilde{x} \notin \mathcal{K}$, it can be projected back to the constraint set as:

$$\hat{x} = \Phi(\mathrm{Proj}_{\mathcal{B}}(\Phi^{-1}(\tilde{x}))) \tag{20}$$

Definition C.4 (RAYEN method (THH23))**.** For convex set $\mathcal{C}$ with interior point x°, given an infeasible prediction $\tilde{x}$, it can be projected to the constraint boundary as $\hat{x} = \gamma \cdot (\tilde{x} - x^\circ) + x^\circ$, where $\gamma = \frac{1}{\varphi_{\mathcal{C}}(\tilde{x} - x^\circ, x^\circ)}$.

The following observation reveals the connection between the bisection projection framework and some existing schemes.

Proposition C.1. *The homeomorphic projection (LCL23) and the RAYEN (THH23) approach are special cases of our bisection projection with only one IP, over convex constraint set.*

Thus, the bisection projection framework provides a unified view for some existing projection-analogous approaches over convex sets. Meanwhile, we highlight the theoretical analysis and application scenario for BP works on general compact sets under Assumption 1 beyond those in the existing studies, further exploring the projection-based design and achieving substantially better performance in feasibility, optimality loss, and speedup as shown in Sec. 6.

Proof. Let's consider applying the gauge mapping to the homeomorphic projection for a compact convex set. Then we can simplify the homeomorphic projection operator as:

$$\hat{x} = \Phi(\mathrm{Proj}_{\mathcal{B}}(\Phi^{-1}(\tilde{x}))) = \Phi(\mathrm{Proj}_{\mathcal{B}}(\frac{\varphi_{\mathcal{C}}(\tilde{x} - x^\circ, x^\circ)}{\|\tilde{x} - x^\circ\|}(\tilde{x} - x^\circ))) \tag{21}$$

$$= \Phi(\frac{\frac{\varphi_{\mathcal{C}}(\tilde{x}-x^\circ, x^\circ)}{\|\tilde{x}-x^\circ\|}(\tilde{x} - x^\circ)}{\|\frac{\varphi_{\mathcal{C}}(\tilde{x}-x^\circ, x^\circ)}{\|\tilde{x}-x^\circ\|}(\tilde{x} - x^\circ)\|}) = \Phi(\frac{\tilde{x} - x^\circ}{\|\tilde{x} - x^\circ\|}) \tag{22}$$

$$= \frac{\|\frac{\tilde{x}-x^\circ}{\|\tilde{x}-x^\circ\|}\|}{\varphi_{\mathcal{C}}(\frac{\tilde{x}-x^\circ}{\|\tilde{x}-x^\circ\|}, x^\circ)}(\frac{\tilde{x} - x^\circ}{\|\tilde{x} - x^\circ\|}) + x^\circ \tag{23}$$

$$= \frac{\tilde{x} - x^\circ}{\varphi_{\mathcal{C}}(\tilde{x} - x^\circ, x^\circ)} + x^\circ \tag{24}$$

When considering $\tilde{x}$ as an infeasible point, we need scale down $\tilde{x}$ of $\frac{1}{\varphi_{\mathcal{C}}(\tilde{x}-x^\circ, x^\circ)}$ such that the $\frac{1}{\varphi_{\mathcal{C}}(\tilde{x}-x^\circ, x^\circ)}(\tilde{x} - x^\circ) + x^\circ$ will be located in the boundary. Therefore, we take $\alpha^* = \frac{1}{\varphi_{\mathcal{C}}(\tilde{x}-x^\circ, x^\circ)}$, the homeomorphic projection operator is indeed the bisection projection operator in (2). It is also equivalent to the RAYEN methods by its definition in C.4.

□

D Unsupervised Training for IPNN

To train the IPNN, we minimize the loss function defined in (6), which incorporates an adversarial penalty term and a smoothed eccentricity term, as delineated in Alg. 2. The IPNN is initialized randomly with m IP predictors. During the training process, Gaussian noise is introduced to perturb the IP predictions, and the penalty term is computed to ensure the identification of valid IPs. For IPs deemed valid, the smoothed eccentricity is calculated using boundary points sampled from a specified region of interest. We consider the following training scenarios with different boundary sampling approaches.

D.1 Objective-aware training

In this scenario, we are provided with a trained NN predictor $F(\cdot)$ or a dataset containing optimal solutions x_θ^* under various inputs. Our objective is to identify IPs with minimized eccentricity in a local region of the boundary. As illustrated in Alg. 3, we initially sample an infeasible solution by introducing noise to the NN prediction or the optimal solution; subsequently, we apply the bisection projection described in Alg. 1 to determine the projected feasible solution in the local boundary.

This training scheme emphasizes the local region surrounding the optimal solution, taking into account the objective function. Consequently, it has the potential to yield a smaller eccentricity and overall optimality gap.

D.2 Objective-agnostic training

In this scenario, we do not have access to any prior information about the optimal solution or a trained NN predictor. Our objective is to minimize the eccentricity with respect to the entire constraint boundary, denoted as $\mathcal{E}(\psi(\theta), \partial\mathcal{C}_\theta)$. To achieve this, we propose a bisection-based boundary sampling algorithm that significantly enhances the efficiency of the training process.

During the training phase, when IPNN outputs an interior point x_θ°, we can sample a set of boundary points using the following steps: (i) sample a unit vector $v \in \mathbb{R}^n$ and (ii) identify a boundary point

Algorithm 2 Unsupervised Training for IPNN

set m, the number of IPs to predict
initialize IPNN $\psi : \mathbb{R}^d \to \mathbb{R}^{m \cdot n}$
sampling input parameter data $\mathcal{D} = \{\theta\}_{i=1}^N \subset \Theta$.
while in training epoch **do**
    batch sampling: $\{\theta\}_{i=1}^B \subset \mathcal{D}$
    Gaussian noise sampling: $\{z_k\}_{k=1}^m \sim \mathcal{N}(0, \sigma^2 I)$
    IPs prediction: $X^\circ_{\theta_i, m} = \psi(\theta_i)$
    adversarial loss for constraint violation: $\mathcal{P}(\psi(\theta_i)) = \sum_{k=1}^m \|\mathrm{ReLU}(g(\psi_k(\theta_i) + z_k, \theta_i))\|$
    if $\psi(\theta_i) \subset \mathcal{C}_{\theta_i}$ **then**
        constraint boundary sampling $\{y_j\}_{j=1}^b$ via Algorithm 3
        computing point-to-boundary distance: $d_{i,j,k} = \|y_j - \psi_k(\theta_i)\|$
        sample-based smoothed eccentricity:

$$\hat{\mathcal{E}}(\psi(\theta_i)) = \mathrm{LSE}_\beta(\{\mathrm{LSE}_{-\beta}(\{d_{i,j,k}\}_{k=1}^m)\}_{j=1}^b) - \mathrm{LSE}_{-\beta}(\{\mathrm{LSE}_{-\beta}(\{d_{i,j,k}\}_{k=1}^m)\}_{j=1}^b)$$

    else
        $\hat{\mathcal{E}}(\psi(\theta_i)) = 0$
    end if
    average loss function: $\mathcal{L}(\psi) = \frac{1}{B}\sum_{i=1}^B \mathcal{P}(\psi(\theta_i))) + \lambda(\hat{\mathcal{E}}(\psi(\theta_i))$
    update IPNN parameter: $\psi \leftarrow \mathrm{Adam}(\mathcal{L}(\psi))$
end while
return ψ

Algorithm 3 Bisection-based Objective-Agnostic/Aware Boundary Sampling

Input: an interior point $x^\circ_\theta \in \mathcal{C}_\theta$.
Output: a feasible point near the boundary.
Objective-Agnostic Sampling
sample Gaussian noise $v \sim \mathcal{N}(0, I)$
normalize to unit vector: $v \leftarrow v/\|v\|$
initialize $\gamma_l = 0$, $\gamma_u = 1$, $n = 0$
while $n \leq K$ **do**
    if $x^\circ_\theta + \gamma_u \cdot v \in \mathcal{C}_\theta$ **then**
        increase lower bound: $\gamma_l \leftarrow \gamma_u$
        increase upper bound: $\gamma_u \leftarrow 2 \cdot \gamma_u$
    else
        decrease upper bound: $\gamma_u \leftarrow (\gamma_l + \gamma_u)/2$
    end if
    $n \leftarrow n + 1$
end while
boundary point: $y = x^\circ_\theta + \gamma_l \cdot v$
Objective-Aware Sampling
set infeasible region radius ϵ
sample Gaussian noise $v \in \mathcal{N}(0, I)$
normalize it as: $v \leftarrow v/\|v\| \cdot \epsilon$
sample infeasible solution as $\tilde{x}_\theta = x^*_\theta + v$ or $\tilde{x}_\theta = F(\theta) + v$
boundary point by bisection projection: $y = \mathrm{BP}(\tilde{x}_\theta, x^\circ_\theta)$
return y

$\hat{y}_\theta \in \partial\mathcal{C}_\theta$ along the direction of v starting from x°_θ as:

$$\hat{y}_\theta = x^\circ_\theta + \gamma^* \cdot v, \tag{25}$$

where $\gamma^* > 0$ leading to $\hat{y}_\theta \in \partial\mathcal{C}_\theta$. The parameter γ can be efficiently determined using a bisection method, as described in Alg. 3. We **remark** that, analogous to the bisection projection described in Alg. 1, this bisection-based boundary sampling algorithm boasts efficiency in per-step computations. Additionally, the algorithm is amenable to batch processing for multiple sampled unit vectors $\{v_i\}_{i=1}^M$, thus further streamlining the training procedure. We also note that analytical solutions for γ^* are available for common convex sets—including linear, convex quadratic, second-order cone (SOC), and linear matrix inequality (LMI) constraints (THH23), which further boost the sampling efficiency for the IPNN training under those constraints.

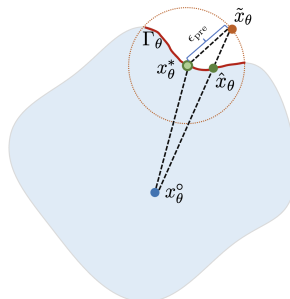


Figure 4: A geometric illustration of eccentricity and the proof.

This training approach is applicable to a wide range of optimization problems with different objective functions but with the same constraint formulation, albeit with higher complexity due to the entire boundary sampling.

E PROOF FOR MAIN RESULTS

E.1 PROOF FOR PROPOSITION 4.1

Proof. We start from a single-IP setting as $x^\circ_\theta \in \mathcal{C}_\theta$.

For an infeasible solution $\tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\text{pre}})$, recall the definition of the projected solution $\hat{x}_\theta$ as:

$$\hat{x}_\theta \triangleq \text{BP}(\tilde{x}_\theta, x^\circ_\theta) = \alpha^* \cdot (\tilde{x}_\theta - x^\circ_\theta) + x^\circ_\theta, \tag{26}$$

where $\alpha^* \in [0, 1]$ leads to $\hat{x}_\theta \in \partial\mathcal{C}_\theta$.

The projection distance for an infeasible prediction $\tilde{x}_\theta$ is bounded as:

$$\begin{aligned}
\|\tilde{x}_\theta - \hat{x}_\theta\| &\overset{(a)}{=} \|\tilde{x}_\theta - x^\circ_\theta\| - \|\hat{x}_\theta - x^\circ_\theta\| &(27)\\
&\overset{(b)}{\leq} \|\tilde{x}_\theta - x^*_\theta\| + \|x^*_\theta - x^\circ_\theta\| - \|\hat{x}_\theta - x^\circ_\theta\| &(28)\\
&\overset{(c)}{\leq} \epsilon_{\text{pre}} + \|x^*_\theta - x^\circ_\theta\| - \|\hat{x}_\theta - x^\circ_\theta\| &(29)\\
&\overset{(d)}{\leq} \epsilon_{\text{pre}} + \max_{x\in\Gamma_\theta} \|x - x^\circ_\theta\| - \min_{x\in\Gamma_\theta} \|x - x^\circ_\theta\| &(30)\\
&\overset{(e)}{=} \epsilon_{\text{pre}} + \mathcal{E}(x^\circ_\theta, \Gamma_\theta) &(31)
\end{aligned}$$

Equality (a) is by three points, $\tilde{x}_\theta$, x°_θ, and $\hat{x}_\theta$, exist in the same straight line. Inequality (b) is by the triangle inequality with auxiliary point x^*_θ. Inequality (c) is by $\tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\text{pre}})$. Inequality (d) is by taking the maximum and minimum point over local boundary $\Gamma_\theta = \{\text{BP}(\tilde{x}_\theta, x^\circ_\theta),\ \forall \tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\text{pre}})$ and $\tilde{x}_\theta \notin \mathcal{C}_\theta\}$. Equality (e) is by the definition of eccentricity in Def. 4.1.

We then complete the proof for a single IP setting as: $\max_{\tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\text{pre}})} \|\tilde{x}_\theta - \hat{x}_\theta\| \leq \epsilon_{\text{pre}} + \mathcal{E}(x^\circ_\theta, \Gamma_\theta)$

For the multiple IPs setting $X^\circ_{\theta,m} = \{x^\circ_{\theta,k}\}_{k=1}^m \subset \mathcal{C}_\theta$, we derive the upper bound as follows:

$$\begin{aligned}
\min_{1\leq k\leq m} \|\tilde{x}_\theta - \hat{x}_{\theta,k}\| &\overset{(a)}{\leq} \min_{1\leq k\leq m} \{\epsilon_{\text{pre}} + \|x^*_\theta - x^\circ_{\theta,k}\| - \|\hat{x}_{\theta,k} - x^\circ_{\theta,k}\|\} &(32)\\
&\overset{(b)}{\leq} \epsilon_{\text{pre}} + \min_{1\leq k\leq m} \|x^*_\theta - x^\circ_{\theta,k}\| - \min_{1\leq k\leq m} \|\hat{x}_{\theta,k} - x^\circ_{\theta,k}\| &(33)\\
&\overset{(c)}{\leq} \epsilon_{\text{pre}} + \max_{x\in\Gamma_\theta} \min_{1\leq k\leq m} \|x - x^\circ_{\theta,k}\| - \min_{x\in\Gamma_\theta} \min_{1\leq k\leq m} \|x - x^\circ_{\theta,k}\| &(34)\\
&\overset{(d)}{=} \epsilon_{\text{pre}} + \mathcal{E}(X^\circ_{\theta,m}, \Gamma_\theta) &(35)
\end{aligned}$$

Inequality (a) is by the bound for the single-IP setting above. Inequality (b) is by the minimization of the joint term, which is smaller than the minimization separately. Inequality (c) is by taking the maximum and minimum point over the local boundary $\Gamma_\theta = \partial\mathcal{C}_\theta \cap \{\mathrm{BP}(\tilde{x}_\theta, X^\circ_{\theta,m}),\ \forall \tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\mathrm{pre}})$ and $\tilde{x}_\theta \notin \mathcal{C}_\theta\}$

Thus, we complete the proof as follows:

$$\max_{\tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\mathrm{pre}})} \min_{1\le k\le m} \|\tilde{x}_\theta - \hat{x}_{\theta,k}\| \le \epsilon_{\mathrm{pre}} + \mathcal{E}(X^\circ_{\theta,m}, \Gamma_\theta) \tag{36}$$

□

E.2 Proof for Proposition 4.2

Recall the definition of the log-sum-exp (LSE) operator as:

$$\mathrm{LSE}_\beta(\{x_j\}_{j=1}^b) = \frac{1}{\beta}\log\sum_{j=1}^b \exp \beta x_j \tag{37}$$

where β is the temperature parameter such that $\mathrm{LSE}_\infty(\{x_j\}_{j=1}^b) = \max(\{x_j\}_{j=1}^b)$ and $\mathrm{LSE}_{-\infty}(\{x_j\}_{j=1}^b) = \min(\{x_j\}_{j=1}^b)$. Further, we have the following basic inequalities for the LSE operator:

$$\max(\{x_j\}_{j=1}^b) \le \mathrm{LSE}_\beta(\{x_j\}_{j=1}^b) \le \max(\{x_j\}_{j=1}^b) + \frac{\log(b)}{\beta} \tag{38}$$

$$\min(\{x_j\}_{j=1}^b) - \frac{\log(b)}{\beta} \le \mathrm{LSE}_{-\beta}(x_1, \cdots, x_b) \le \min(\{x_j\}_{j=1}^b) \tag{39}$$

Recall the definition of sample-based eccentric distance:

$$\overline{\mathcal{E}}(X^\circ_{\theta,m}, \Gamma_\theta) := \max_{1\le j\le b}\min_{1\le k\le m} d_{\theta,j,k} - \min_{1\le j\le b}\min_{1\le k\le m} d_{\theta,j,k}.$$

where the point-to-boundary distance as $d_{\theta,j,k} = \|y_j - \psi_k(\theta)\|$ is the point-to-boundary distance for boundary samples $\{y_j\}_{j=1}^b \subset \Gamma_\theta$ and predicted IPs $X^\circ_{\theta,m} = \psi(\theta)$.

The sample approximated eccentric distance can be bounded as:

$$\overline{\mathcal{E}}(X^\circ_{\theta,m}, \Gamma_\theta) = \max_{1\le j\le b}\{\min_{1\le k\le m}\{d_{\theta,j,k}\}_{k=1}^m\}_{j=1}^b - \min_{1\le j\le b}\{\min_{1\le k\le m}\{d_{\theta,j,k}\}_{k=1}^m\}_{j=1}^b \tag{40}$$

$$\le \max_{1\le j\le b}\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m) + \frac{\log(m)}{\beta}\}_{j=1}^b - \min_{1\le j\le b}\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b \tag{41}$$

$$\le \mathrm{LSE}_\beta(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b) - \mathrm{LSE}_{-\beta}(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b) + \frac{\log(m)}{\beta} \tag{42}$$

$$= \hat{\mathcal{E}}(X^\circ_{\theta,m}, \Gamma_\theta) + \frac{\log(m)}{\beta} \tag{43}$$

where the smoothed sample-based eccentric distance is as: $\hat{\mathcal{E}}(X^\circ_{\theta,m}, \Gamma_\theta) = \mathrm{LSE}_\beta(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b) - \mathrm{LSE}_{-\beta}(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b)$.

Similarly, the lower bound can be derived as:

$$\overline{\mathcal{E}}(X^\circ_{\theta,m}, \Gamma_\theta) = \max_{1\le j\le b}\{\min_{1\le k\le m}\{d_{\theta,j,k}\}_{k=1}^m\}_{j=1}^b - \min_{1\le j\le b}\{\min_{1\le k\le m}\{d_{\theta,j,k}\}_{k=1}^m\}_{j=1}^b \tag{44}$$

$$\ge \max_{1\le j\le b}\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b - \min_{1\le j\le b}\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m) + \frac{\log(m)}{\beta}\}_{j=1}^b \tag{45}$$

$$\ge \mathrm{LSE}_\beta(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b) - \frac{\log(b)}{\beta} - (\mathrm{LSE}_{-\beta}(\{\mathrm{LSE}_{-\beta}(\{d_{\theta,j,k}\}_{k=1}^m)\}_{j=1}^b) + \frac{\log(bm)}{\beta}) \tag{46}$$

$$= \hat{\mathcal{E}}(X^\circ_{\theta,m}, \Gamma_\theta) - \frac{\log(mb^2)}{\beta} \tag{47}$$

Therefore, the sample-based eccentric distance can be smoothly approximated by the LSE operator when $\beta \to \infty$.

$$\hat{\mathcal{E}}(X^{\circ}_{\theta,m}, \Gamma_\theta) - \frac{\log(mb^2)}{\beta} \leq \overline{\mathcal{E}}(X^{\circ}_{\theta,m}, \Gamma_\theta) \leq \hat{\mathcal{E}}(X^{\circ}_{\theta,m}, \Gamma_\theta) + \frac{\log(m)}{\beta} \tag{48}$$

E.3 Proof for Proposition 5.1

Proof. Given a compact constraint set $\mathcal{C}_\theta \subset \mathbb{R}^n$ under Assumption 1, which has a non-empty interior or is Euclidean of dimension n. Its boundary $\partial\mathcal{C}_\theta$ is also compact and Euclidean of dimension $n-1$.

Thus, there exists a δ-covering set, denoted as $\mathcal{D}$, for $\partial\mathcal{C}_\theta$ as $\{y_k\}_{k=1}^m \subset \partial\mathcal{C}_\theta$, such that for any $z \in \partial\mathcal{C}_\theta$, there exists $y \in \mathcal{D}$ such that $\|y - z\| \leq \delta$.

Next, for every boundary covering points $y \in \mathcal{D}$, we construct its corresponding IP as $x^\circ = y + v$ where $\|v\| > 0$ is strictly positive but it can be arbitrarily small such that $x^\circ \in \mathcal{C}_\theta$. Subsequently, the eccentricity measure under those constructed IPs $X^{\circ}_{\theta,m} = \{x^\circ_\theta = y_k + v_k \in \mathcal{C}_\theta \mid y_k \in \mathcal{D},\ \|v_k\| > 0\}$ is bounded as follows:

$$\begin{aligned}
\mathcal{E}(X^{\circ}_{\theta,m}, \partial\mathcal{C}_\theta) &= \max_{z\in\partial\mathcal{C}_\theta} \min_{x^\circ \in X^{\circ}_{\theta,m}} \|z - x^\circ\| - \min_{z\in\partial\mathcal{C}_\theta} \min_{x^\circ \in X^{\circ}_{\theta,m}} \|z - x^\circ\| && (49)\\
&\overset{(a)}{=} \max_{z\in\partial\mathcal{C}_\theta} \min_{x^\circ \in X^{\circ}_{\theta,m}} \|x^\circ - z\| && (50)\\
&\overset{(b)}{=} \max_{z\in\partial\mathcal{C}_\theta} \min_{y\in\mathcal{D}} \|y + v - z\| && (51)\\
&\overset{(c)}{=} \max_{z\in\partial\mathcal{C}_\theta} \|y(z) + v - z\| && (52)\\
&\overset{(d)}{\leq} \max_{z\in\partial\mathcal{C}_\theta} \|y(z) - z\| + \|v\| && (53)\\
&\overset{(e)}{\leq} \delta + \|v\| && (54)
\end{aligned}$$

Inequality (a) is by dropping the negative term. Equality (b) is by the definition of constructed IP given $y \in \mathcal{D}$. Equality (c) is by defining the optimal solution for the second minimization term as $y(z) = \arg\min_{y'\in\mathcal{D}}\{\|y' + v - z\|\}$. Inequality (d) is by the triangle inequality. Inequality (e) is by the property of covering the dataset such that for any $z \in \partial\mathcal{C}_\theta$ there exists $y \in \mathcal{D}$ leading to $\|z - y\| \leq \delta$.

Since the constructed $\|v\|$ for each boundary point $y \in \mathcal{D}$ can be arbitrarily small, to achieve the δ eccentricity measure, we can adopt the δ-covering data set D for $\partial\mathcal{C}_\theta$ and the constructed IPs with same size of $\mathcal{D}$, where the covering number is with an order of $\mathcal{O}(\delta^{-(n-1)})$. In other words, to reach the δ eccentricity measure, we may use a δ-covering dataset to construct the set of IPs, and the minimum number of IPs is upper bounded by this construction. Equivalently, the minimum eccentricity measure under m IPs is also upper bounded by $\mathcal{O}(m^{-1/(n-1)})$

Further, if we also incorporate the constant-rank equality constraint, where the Jacobian of the equality function is of a constant rank r. Then, as per the *Constant-Rank Level Set Theorem* (Lee13), the ambient dimension of $\mathcal{C}_\theta$ can be reduced to $n-r$ such that the covering number for its boundary can be reduced to $\mathcal{O}(\delta^{-(n-r-1)})$). A similar setting of low-dimensional constraint set embedding is also considered in (KZLD21).

On the other hand, when we focus on the eccentricity $\mathcal{E}(X^{\circ}_{\theta,m}, \Gamma_\theta)$ modulated by the NN infeasibility region, we constructed those IPs in the same point as $x^\circ = x^* + v$, where $\|v\| > 0$ is strictly positive but it can be arbitrarily small such that $x^\circ \in \mathcal{C}_\theta$. Subsequently, the local boundary is as $\Gamma_\theta = \{\mathrm{BP}(\tilde{x}_\theta, X^{\circ}_{\theta,m}),\ \forall \tilde{x}_\theta \in \mathcal{B}(x^*_\theta, \epsilon_{\text{pre}})$ and $\tilde{x}_\theta \notin \mathcal{C}_\theta\}$

$$\begin{aligned}
\mathcal{E}(x^\circ, \Gamma_\theta) &= \max_{z\in\Gamma_\theta} \|z - x^\circ\| - \min_{z\in\Gamma_\theta} \|z - x^\circ\| && (55)\\
&\overset{(a)}{\leq} \max_{z\in\Gamma_\theta} \|x^\circ - z\| && (56)
\end{aligned}$$

$$\overset{(b)}{\leq} \epsilon_{\text{pre}} + \|v\| \tag{57}$$

inequality (a) is by dropping the negative term. Inequality (b) is by taking the point as along the line $z - x^\circ$ until reaching the infeasibility boundary of $\mathcal{B}(x^*_\theta, \epsilon_{\text{pre}})$. Since the constructed $\|v\|$ can be arbitrarily small, we have the local eccentricity modulated by the NN infeasibility region, which can be directly bounded by the prediction error.

Combining the two cases with constructed interior points, we have minimum eccentricity is bounded as:

$$\min_{X^\circ_{\theta,m} \subset \mathcal{C}_\theta} \mathcal{E}(X^\circ_{\theta,m}, \Gamma_\theta) \leq \min\{\epsilon_{\text{pre}}, \mathcal{O}\left(m^{-1/(n-1)}\right)\} \tag{58}$$

□

E.4 Proof for Sample-based Condition in Theorem 1

Proof. The sample-based condition is derived from Theorem 2 in (LCL23). This condition uses a dataset $\mathcal{D} = \{\theta_i, i = 1, \ldots, N\} \subseteq \Theta$ as an r_c-covering training set. This implies that for any $\theta \in \Theta$, there exists a $\theta_i \in \mathcal{D}$ such that $\|\theta - \theta_i\| \leq r_c$.

Without loss of generality, we focus on an IPNN with one IP predictor $\psi_k(\cdot)$. The feasibility of the predictor under new input parameters is maintained if the following condition holds:

$$r_c C_0 + r_c C_1^k \leq C_2^k, \tag{59}$$

where

- $C_0 = \sup\limits_{\theta_1,\theta_2 \in \Theta, \theta_1 \neq \theta_2} \{\frac{\mathrm{d_H}(\partial\mathcal{C}_{\theta_1}, \partial\mathcal{C}_{\theta_2})}{\|\theta_1 - \theta_2\|}\}$ represents the largest rate of variation for the constraint boundary under different inputs. This allows us to capture the change in the constraint boundary within an r_c neighborhood of any $\theta \in \mathcal{D}$.
- $C_1^k = \mathrm{Lip}(\psi_k(\cdot), \Theta)$ denotes the Lipschitz constant of the k-th IP predictor over the input domain Θ, and $r_c C_1^k$ calculates the variation of the predicted IP within the r_c neighborhood for any $\theta \in \mathcal{D}$.
- $C_2^k = \min\limits_{\theta \in \mathcal{D}}\{\min\limits_{z \in \partial\mathcal{C}_\theta} \{\|z - \psi_k(\theta)\|\}\}$ is the minimum distance from the IP to the boundary of the constraint set for all $\theta \in \mathcal{D}$. This term essentially provides a "buffer" to account for the variation of both the IP and the constraint boundary.

When the specified condition is satisfied, it ensures that the predicted IP under any new input parameter remains feasible despite the variations in the constraint boundary and the IP itself.

Therefore, for the valid condition of whole set IP predictions, we define the constants and conditions as:

$$r_c C_0 + r_c C_1 \leq C_2, \tag{60}$$

where $C_1 = \max_{1 \leq k \leq m}\{C_1^k\}$ and $C_2 = \min_{1 \leq k \leq m}\{C_2^k\}$.

□

E.5 Proof for Verification-based Condition in Theorem 1

Proof. Without loss of generality, we focus on an IPNN with one IP predictor $\psi_k(\cdot)$ and one constraint function $g_i(x, \theta)$. The verification of other IP predictors or constraint functions can be solved independently. The feasibility verification problem is formulated as:

$$\textbf{P0}: \quad \max\ g_i(x, \theta) \tag{61}$$

$$\text{s.t. } x = \psi_k(\theta) \tag{62}$$

$$\text{var. } \theta \in \Theta \tag{63}$$

The optimal objective t^* represents the worst-case constraint violation for the NN output given arbitrary input $\theta \in \Theta$. This problem is generally non-convex and NP-hard due to the NN structure or non-convex constraint function. To derive an upper bound of the worst-case constraint violation as a sufficient condition for verifying NN feasibility, we apply convex relaxation techniques to reformulate the NP-hard problem into a convex one:

- We apply relaxation techniques (e.g., linear or SDP relaxation) for the ReLU-based NN (RSL18; FMP20; WZX$^+$21). For example, the linear relaxation of the ReLU unit is defined as:

$$\mathrm{CR}(\mathrm{ReLU(x)}) \begin{cases} = x & \text{if } L \geq 0 \\ = 0 & \text{if } U \leq 0 \\ \in \{y \mid y \geq 0, y \geq x, y \leq \frac{U}{U-L}(x-L)\} & \text{otherwise} \end{cases} \tag{64}$$

  where input variable $x \in [L, U]$ is bounded, and the lower/upper bound for the per-layer output can be computed separately by solving a similar but smaller verification problem (TXT17; ZPCL23). This pre-computing operation significantly tightens the relaxation for the ReLU-type neural network, making the upper bound close to the optimal value. End-to-end verification packages such as α, β-CROWN (WZX$^+$21; SJK$^+$24) can also be directly applied to NN feasibility verifications without manual design.

- Second, the constraint function $g_i(x, \theta)$ can be non-linear; we discuss the following cases: (i)If $g_i(x, \theta)$ is linear or concave, the maximization problem becomes a convex optimization after NN relaxation. (ii) For other constraint functions, we treat $g_i(x, \theta)$ as an equality constraint $t = g_i(x, \theta)$ and apply convex relaxation to obtain $t \in \mathrm{CR}(g_i(x, \theta))$ (QOB$^+$19). This convex relaxation is problem-dependent (see examples in Appendix F.2).

The relaxed verification problem can be formulated as the following convex programming task, enabling polynomial-time algorithms (e.g., interior point methods) to find optimal solutions:

$$\begin{aligned} \textbf{P1}: \quad & \max t && (65) \\ & \text{s.t. } t \in \mathrm{CR}(g_i(x, \theta)) && (66) \\ & \quad x \in \mathrm{CR}(\psi_k(\theta)) && (67) \\ & \text{var. } \theta \in \Theta && (68) \end{aligned}$$

The optimal solution $\hat{t}^*$ is an upper bound of the worst-case constraint violation. The tightness of the relaxation depends on both NN reformulation and convex relaxation, which may be problem-dependent in empirical experiments. □

E.6 Proof for Theorem 2

Proof. First, the feasibility of the solution returned through bisection is guaranteed due to the bisection trajectory connecting an infeasible point and an interior point, which must intersect the constraint boundary. Thus, the bisection algorithm can always find a feasible solution by scaling down the infeasible solution along the line segment. We remark that for general non-convex sets, the line segment between an infeasible point and an interior point may intersect the constraint boundary multiple times, causing our bisection algorithm to converge to one of the multiple feasible solutions.

Without loss of generality, let's $\hat{x}_{\theta,k} \in \partial \mathcal{C}_\theta$ be the converged boundary feasible solution given infinite bisection with an interior point $x^\circ_{\theta,k}$, which also satisfies the definition of projected solution in Eq. (2). And $\hat{x}_\theta = \arg\min_{\hat{x}_{\theta,k}} \{\|\hat{x}_{\theta,k} - \tilde{x}_\theta\|\}$ is the best projected solution.

We divide the optimality gap by the following three terms:

$$\|\hat{x}^K_\theta - x^*_\theta\| \leq \underbrace{\|x^*_\theta - \tilde{x}_\theta\|}_{\text{prediction error}} + \underbrace{\|\tilde{x}_\theta - \hat{x}_\theta\|}_{\text{projection error}} + \underbrace{\|\hat{x}_\theta - \hat{x}^K_\theta\|}_{\text{bisection error}} \tag{69}$$

The prediction error is determined by the provided NN predictor, and we denoted it as $\epsilon_{\text{pre}} = \sup_{\theta \in \Theta}\{\|F(\theta) - x^*_\theta\|\}$, where $F(\cdot)$ is the NN predictor to predict the optimal solution.

The projection error from bisection-projection, as proved in Prop. 4.1, can be bonded by the eccentricity measure related term as:

$$\|\tilde{x}_\theta - \hat{x}_\theta\| = \min_{1\le k\le m} \|\tilde{x}_\theta - \hat{x}_{\theta,k}\| \tag{70}$$

$$\le \max_{y\in\mathcal{B}(x_\theta^*,\epsilon_{\rm pre})} \min_{1\le k\le m} \|\tilde{x}_\theta - \hat{x}_{\theta,k}\| \tag{71}$$

$$\le \epsilon_{\rm pre} + \mathcal{E}(X_{\theta,m}^\circ, \Gamma_\theta) \tag{72}$$

Since $\hat{x}_{\theta,k}$ is the converged boundary feasible solution under the bisection algorithm, the bisection error induced by finite step iteration can be derived as:

$$\|\hat{x}_{\theta,k} - \hat{x}_{\theta,k}^K\| \le \|\alpha_k^* \cdot (\tilde{x}_\theta - x_\theta^\circ) + x_{\theta,k}^\circ - (\alpha_k^K \cdot (\tilde{x}_\theta - x_\theta^\circ) + x_{\theta,k}^\circ)\| \tag{73}$$

$$= (\alpha_k^* - \alpha_k^K)\|\tilde{x}_\theta - x_{\theta,k}^\circ\| \tag{74}$$

$$\le 2^{-K}\|\tilde{x}_\theta - x_\theta^* + x_\theta^* - x_{\theta,k}^\circ\| \tag{75}$$

$$\le 2^{-K}(\|\tilde{x}_\theta - x_\theta^*\| + \|x_\theta^* - x_{\theta,k}^\circ\|) \tag{76}$$

$$\le 2^{-K}(\epsilon_{\rm pre} + D) \tag{77}$$

where $D = \mathrm{diam}(\mathcal{C}_\theta)$ denote the diameter of a compact set. This bound holds for every interior point $1 \le k \le m$.

However, we remark that the defined $\hat{x}_\theta = \arg\min_{\hat{x}_{\theta,k}}\{\|\hat{x}_{\theta,k} - \tilde{x}_\theta\|\}$ may not share the same optimal index k with $\hat{x}_\theta^K = \arg\min_{\hat{x}_{\theta,k}^K}\{\|\hat{x}_{\theta,k}^K - \tilde{x}_\theta\|\}$, due to finite-step bisection execution error. Further, the optimal index for the interior point may not be unique for both cases. This makes the distance between $\|\hat{x}_\theta - \hat{x}_\theta^K\|$ may not monotonically decrease with bisection steps K.

Let's denote $s_1 \in \arg\min_{1\le k\le m}\{\|\hat{x}_{\theta,k} - \tilde{x}_\theta\|\}$ and $s_2 \in \arg\min_{1\le k\le m}\{\|\hat{x}_{\theta,k}^K - \tilde{x}_\theta^K\|\}$, δ_g be the gap between the minimum and strictly second minimum of the projection distance $\|\tilde{x}_\theta - \hat{x}_{\theta,k}\|$, and we make the following classification and discussion.

- when $m = 1$, there is only one interior point such that $s_1 = s_2$. It is straightforward to conclude $\|\hat{x}_\theta - \hat{x}_\theta^K\| \le 2^{-K}(\epsilon_{\rm pre} + \mathrm{diam}(\mathcal{C}_\theta))$.
- when $m > 1$, there are multiple interior points. If $\delta_g = 0$, which means all projection distances $\|\hat{x}_{\theta,k} - \tilde{x}_\theta\|$ are equal, we then select $s_1 = s_2$ such that $\|\hat{x}_\theta - \hat{x}_\theta^K\| \le 2^{-K}(\epsilon_{\rm pre} + D)$.
- when $m > 1$, there are multiple interior points. If $\delta_g > 0$, to align s_1 with s_2, we consider the condition as $2^{-K}(\epsilon_{\rm pre} + D) \le \delta_g$ such that the finite bisection execution error will not affect the choice of minimum solution and make $s_2 \in \arg\min_{1\le k\le m}\{\|\hat{x}_{\theta,k} - \tilde{x}_\theta\|\}$. Therefore, we can select $s_1 = s_2$ such that $\|\hat{x}_\theta - \hat{x}_\theta^K\| \le 2^{-K}(\epsilon_{\rm pre} + D)$.
- when $m > 1$ and $2^{-K}(\epsilon_{\rm pre} + D) > \delta_g$, the index s_1 and s_2 may not be the same. The distance can be bounded as $\|\hat{x}_{\theta,s_1} - \hat{x}_{\theta,s_2}^K\| \le \|\hat{x}_{\theta,s_1} - \hat{x}_{\theta,s_2}\| + \|\hat{x}_{\theta,s_2} - \hat{x}_{\theta,s_2}^K\| \le \epsilon_g + 2^{-K}(\epsilon_{\rm pre} + D)$, where we denote $\epsilon_g = \|\hat{x}_{\theta,s_1} - \hat{x}_{\theta,s_2}\|$ as the distance between different converged solutions.

Combining the three terms together, we have:

$$\|\hat{x}_\theta^K - x_\theta^*\| \le 2\epsilon_{\rm pre} + \mathcal{E}(X_{\theta,m}^\circ, \Gamma_\theta) + 2^{-K}(\epsilon_{\rm pre} + D) + \epsilon_g \tag{78}$$

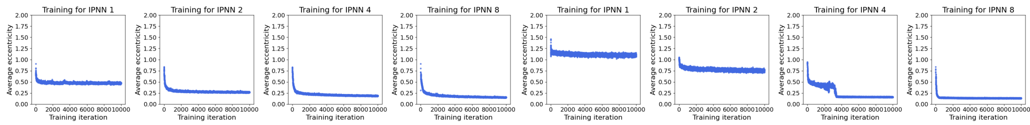
When $m = 1$ or $K \ge \log_2(C_3)$ with $C_3 = \frac{\epsilon_{\rm pre}+D}{\delta_g}$, we have $\epsilon_g = 0$.

The complexity of executing the bisection algorithm involves the iteration steps, the number of IPs, and the complexity of verifying the inequality constraints at each iteration as G. For example, if the inequality constraint $g_i(x,\theta)$ is a linear function for all $i = 1, \cdots, n_{\rm ineq}$, then $G = n \cdot n_{\text{ineq}}$. In contrast, iterative algorithms such as interior point methods have a complexity of $\mathcal{O}((n + n_{\rm ineq})^3)$ at each iteration due to the matrix inversion operation.

$\square$

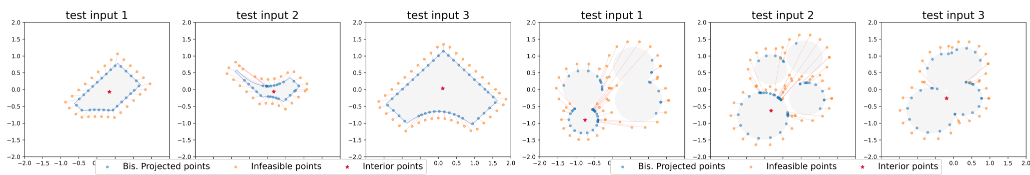
F Supplementary Experiment Results

F.1 Visualization of BP for 2-dim non-convex constraint set

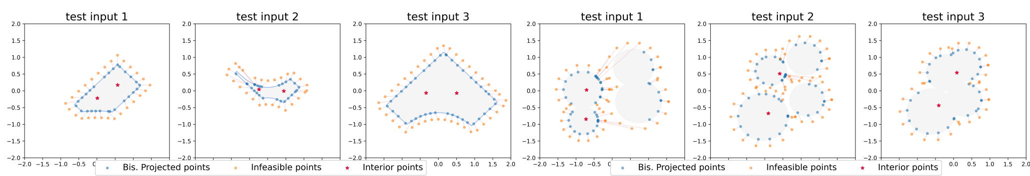


(a) Unsupervised IPNN training over case 1. (b) Unsupervised IPNN training over case 2.

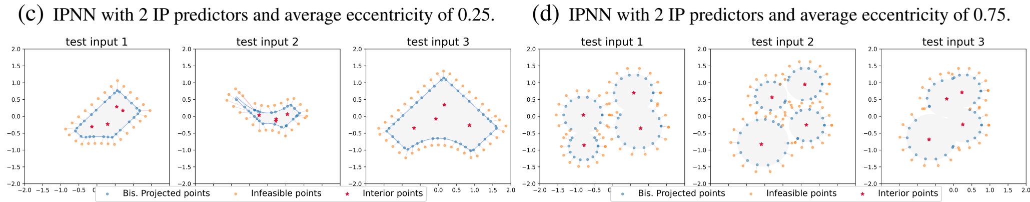
Figure 5: IPNN training and convergence of eccentricity.



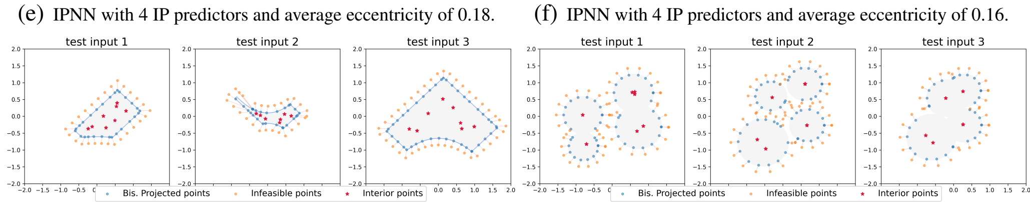
(a) IPNN with 1 IP predictors and average eccentricity of 0.49. (b) IPNN with 1 IP predictors and average eccentricity of 1.12



(c) IPNN with 2 IP predictors and average eccentricity of 0.25. (d) IPNN with 2 IP predictors and average eccentricity of 0.75.



(e) IPNN with 4 IP predictors and average eccentricity of 0.18. (f) IPNN with 4 IP predictors and average eccentricity of 0.16.



(g) IPNN with 8 IP predictors and average eccentricity of 0.12. (h) IPNN with 8 IP predictors and average eccentricity of 0.13.

Figure 6: Training IPNN for approximating MEIPs and testing Bisection Projection with trained IPNN over case 1 (**Left** column) and case 2 (**Right** column).

We investigate the training of IPNN ψ to learn the MEIPs for two non-convex examples, including a ball-homeomorphic set and a more complex one, defined as:

$$\text{Case 1: } \mathcal{K}_\omega = \{x \mid x^\top Qx + q^\top x + b \leq 0\}, \qquad \text{Case 2: } \mathcal{C}_\theta = \cup_{i=1}^{4} \mathcal{B}(x_i, r_i), \tag{79}$$

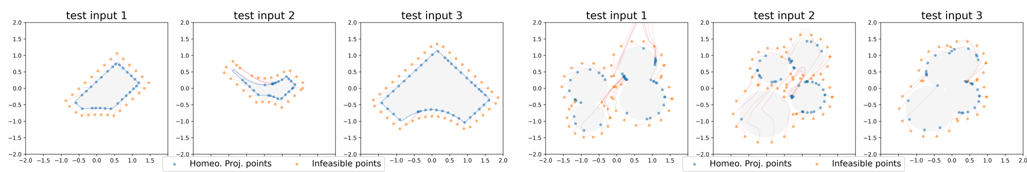
where the input parameter is defined as $\omega = \{Q, q, b\}$ and $\theta = \{x_i, r_i\}_{i=1}^{4}$. We remark that both sets are non-convex, and their geometry can be highly irregular, such as disconnectivity, making them suitable for testing the BP framework. We then train IPNN to approximate the MEIPs and test them on unseen parameters to validate the effectiveness of BP.

We provide the visualizations of BP for different constraint sets with different parameter settings, including:

- In Fig. 5, we train IPNN for ball-homeomorphic and non-ball-homeomorphic sets with 1,2,4,8 IP predictors, and compare their eccentricity convergence with iterations.

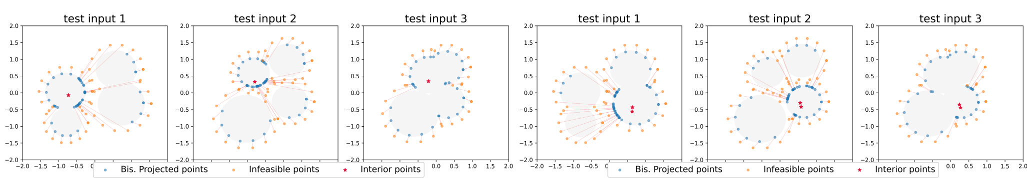
- In Fig. 6, we test BP with trained IPNN over those sets to compare the projection distance for different infeasible points.
- In Fig. 7, we compare Homeomorphic Projection (LCL23; LCL24) over those sets to compare it with our BP methods.
- In Fig. 8, we train IPNN without minimizing eccentricity and test BP with them to justify the necessity of eccentricity design.

we have the following observations: (i) the unsupervised training approach derives valid IPNN over unseen input parameters; (ii) the bisection projection recovers the feasibility of all sampled infeasible predictions with minor projection distance; (iii) increasing the number of IPs helps to reduce eccentricity and further mitigate the optimality loss over complex constraint sets regardless of the ball-homeomorphism property; (iv) homeomorphic projection performs similarly to bisection over case 1 but fails over case 2. In particular, it is not difficult to observe that bisection projection outperforms homeomorphic projection with significantly smaller optimality losses over these non-ball-homeomorphic sets.

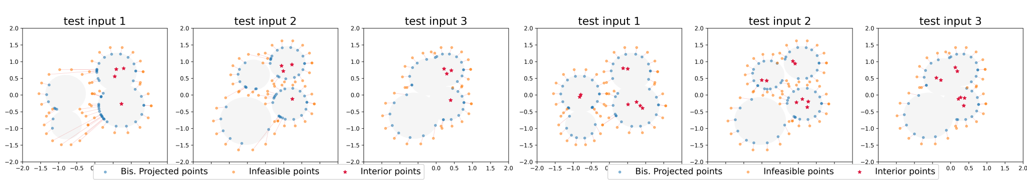


(a) Homeomorphic projection over case 1. (b) Homeomorphic projection over case 2.

Figure 7: Bisection *vs* Homeomorphic projection over case 1 and case 2.



(a) IPNN with 1 IP predictors without minimizing eccentricity (b) IPNN with 2 IP predictors without minimizing eccentricity.



(c) IPNN with 4 IP predictors without minimizing eccentricity. (d) IPNN with 8 IP predictors without minimizing eccentricity.

Figure 8: Training IPNN without minimizing eccentricity and test Bisection Projection with it over non-ball-homeomorphic sets.

F.2 Feasibility verification of IPNN over convex/non-convex problem

In Theorem 1, we present a verification-based condition to validate the validity of IPNN over the entire input domain after it is trained to be feasible over finite input samples. To demonstrate the effectiveness of this condition, we consider solving the constraint violation upper bound for the following problem with different constraint structures. The results are shown in Table 5, which demonstrates the feasibility guarantee for IPNN trained by minimizing eccentricity.

Table 5: Verification for IPNN over different constraint sets.

IPNN	Constraint violation upper bound
QP: : $n = 100,\ d = 50,\ n_{\text{eq}} = 50,\ n_{\text{ineq}} = 50$ **Input constraint:** $\theta \in [\theta_l, \theta_u]$**, output constraint:** $Ax = \theta, Gx \leq h$ **NN relaxation: linear, constraint reformulation: N/A**	
Untrained	315.285
Trained	-0.622
Convex QCQP: : $n = 100,\ d = 50,\ n_{\text{eq}} = 50,\ n_{\text{ineq}} = 50$ **Input constraint:** $\theta \in [\theta_l, \theta_u]$**, output constraint:** $Ax = \theta, x^{\mathrm{T}} H_i x + g_i x \leq h_i$ **NN relaxation: linear, constraint reformulation: SDP (QOB$^+$19)**	
Untrained	1267.653
Trained	-2.67
Non-Convex JCCIM: : $n = 400,\ d = 100,\ n_{\text{eq}} = 100,\ n_{\text{ineq}} = 10100$ **Input constraint:** $\theta \in [\theta_l, \theta_u]$**, output constraint:** $\frac{1}{100} \sum_{k=1}^{100} \mathbf{I}(Ax \geq \theta + \omega_k) \geq 1 - \epsilon, Gx \leq h$ **NN relaxation: linear, constraint reformulation: robust (PAS09)**	
Untrained	95.905
Trained	-0.053

[1] We apply a 3-layer IPNN with ReLU activation, hidden dimension of n, and residual connection in hidden layers.
[2] We apply CVXPY to formulate all relaxed verification problems and use GUROBI to solve them optimally.

G Detailed Experiment Setting

All neural network-based methods were conducted on an Ubuntu server equipped with an NVIDIA A800 GPU. Iterative algorithms were executed in parallel on a CPU featuring 255 processors. To solve two convex optimization problems, we utilized the MOSEK optimizer under an academic license. Alternating Current Optimal Power Flow (ACOPF) problems were addressed using the open-source PyPower tool (ZMS11). Detailed specifications for additional experimental configurations are provided in the respective sections and footnotes below the Table.

G.1 Descriptions of compared approaches in experiments

- **Optimizer**: for convex optimization, we use MOSEK to solve the optimal solution. For AC-OPF problems, we adopt PYPOWER as the specialized solver (ZMS11);
- **NN**: it directly maps input parameter to the solution;
- **Proj**: the infeasible predicted solution by NN is processed by orthogonal projection and solved with the iterative solver;
- **WS**: The infeasible prediction of NN is regarded as the warm-start initialization for the iterative solver;
- **D-Proj**: this is proposed in DC3 (DRK20), which applies gradient descent with implicit function theorem to conduct projection in a differentiable manner;
- **H-Proj**: the homeomorphic projection are applied to the infeasible predictions (LCL23);
- **B-Proj**: we apply bisection in Alg. 1 with predicted IPs to recover the feasibility.

The criteria include (i) **Feasibility**: the feasibility rate of 1,024 testing instances and average constraint violation; (ii) **Optimality**: the solution error and objective gap evaluate the optimality of predicted solutions; (iii) **Speedup**: the total inference time (NN inference time + post-processing time) of all testing instances and calculate the speedup compared with the solvers under parallel execution.

G.2 Formulation for constrained optimization problems

We test the Bisection Projection framework for four constrained optimization problems, including two convex optimization problems and two real-world non-convex problems.

G.2.1 CONVEX PROBLEM FORMULATION

The Convex QCQP extends the QP by including quadratic constraints. The Convex QCQP problem is given by:

$$\textbf{Convex QCQP}: \quad \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\mathbf{x}^\intercal Q\mathbf{x} + \mathbf{p}^\intercal \mathbf{x} \tag{80}$$

$$\text{subject to} \quad \mathbf{x}^\intercal H_i \mathbf{x} + \mathbf{g}_i^\intercal \mathbf{x} \leq h_i,\ i = 1, \ldots, n_{\text{ineq}}, \tag{81}$$

$$A\mathbf{x} = \boldsymbol{\theta}, \tag{82}$$

where $Q \in \mathbb{S}^n_{++}$ is a positive definite matrix, ensuring the convexity of the objective function, $\mathbf{p} \in \mathbb{R}^n$ is a vector of linear coefficients, $A \in \mathbb{R}^{n_{\text{eq}}\times n}$ is a matrix defining equality constraints, each $H_i \in \mathbb{S}^n_{++}$ is a positive definite matrix corresponding to the i-th quadratic constraint, $\mathbf{g}_i \in \mathbb{R}^n$ is a vector of linear coefficients for the quadratic constraints, and $h_i \in \mathbb{R}$ represents the upper bound for the i-th quadratic constraint.

The SOCP is a convex optimization problem that generalizes linear and quadratic programs by allowing conic constraints. A SOCP problem is formulated as follows:

$$\textbf{SOCP}: \quad \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\mathbf{x}^\intercal Q\mathbf{x} + \mathbf{p}^\intercal \mathbf{x} \tag{83}$$

$$\text{subject to} \quad \|G_i\mathbf{x} + \mathbf{h}_i\|_2 \leq \mathbf{c}_i^\intercal \mathbf{x} + d_i,\ i = 1, \ldots, n_{\text{ineq}}, \tag{84}$$

$$A\mathbf{x} = \boldsymbol{\theta}, \tag{85}$$

where $G_i \in \mathbb{R}^{m\times n}$ and $\mathbf{h}_i \in \mathbb{R}^m$ define the second-order cone, $\mathbf{c}_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$ are the coefficients and scalar terms of the conic constraints, respectively.

G.2.2 JOINT CHANCE CONSTRAINED INVENTORY MANAGEMENT (JCC-IM)

We consider the Joint Chance-Constrained Inventory Management (JCC-IM) problem, which seeks to optimize inventory levels across multiple warehouses under conditions of demand uncertainty, ensuring a high probability of meeting that demand. The JCC-IM problem is formally defined as:

$$\textbf{JCC-IM}: \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \mathbf{c}^\intercal \mathbf{x} \tag{86}$$

$$\text{subject to} \quad \mathbf{Prob}(A\mathbf{x} \geq \theta + \omega) \geq 1 - \delta \tag{87}$$

$$G\mathbf{x} \leq \mathbf{h},\ \mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max}, \tag{88}$$

where n denotes the number of warehouses located in distinct regions, the decision variable $\mathbf{x}$ represents the inventory order quantity to be determined in advance for n warehouse, in order to satisfy future demand. The vector θ encapsulates the historical average demand, and the term $\omega \sim p(\cdot)$ models the stochastic deviations from this average, capturing the inherent uncertainty of demand. The matrix A characterizes the interdependencies among different warehouses, which may arise from shared types of inventory or geographical proximity. The parameter δ specifies the acceptable risk level, thus ensuring that the probability of meeting demand across all warehouses is at least $1 - \delta$. The additional constraints, $G\mathbf{x} \leq \mathbf{h}$ and $\mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max}$, represent warehouse-specific capacity limitations and inventory bounds, respectively.

Given the absence of an analytical representation for the JCC, we employ a Sample-Average (SA) approach to conservely approximate the chance-constrained problem. This technique involves generating a finite set of scenarios $\{\tilde{\omega}_j\}_{j=1}^N$ from the underlying distribution of θ. The SA variant of the JCC-IM is formulated as:

$$\textbf{SA-JCC-IM}: \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \mathbf{c}^\intercal \mathbf{x} \tag{89}$$

$$\text{subject to} \quad P_N = \frac{1}{N}\sum_{j=1}^N \mathrm{I}(A\mathbf{x} \geq \theta + \tilde{\omega}_j) \geq 1 - \delta \tag{90}$$

$$G\mathbf{x} \leq \mathbf{h},\ \mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max} \tag{91}$$

where $\mathrm{I}(\cdot)$ is the indicator function. A solution is deemed to have a probabilistic JCC feasibility guarantee if $P_N \geq 1 - \delta$. This empirical evaluation provides a practical measure of the reliability of the SA-based solution in adhering to the demand satisfaction requirements stipulated by the JCC-IM problem.

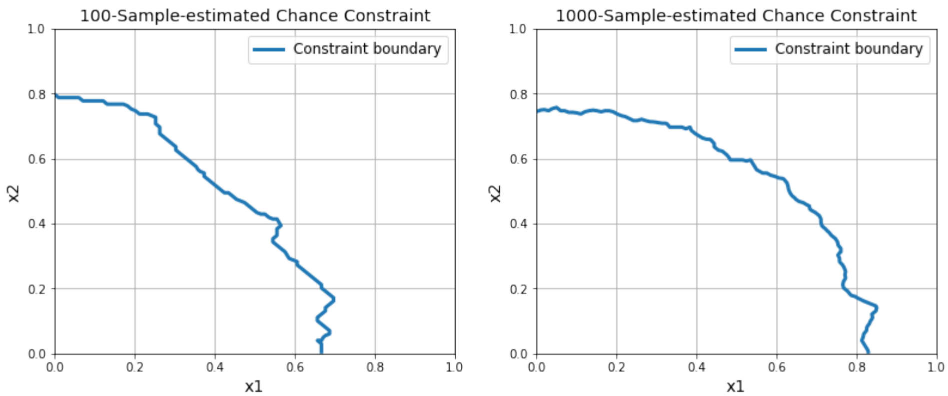


Figure 9: This figure demonstrates a two-dimensional sample-based chance constraint defined as

$$\mathrm{Prob}(\omega_1 x_1 + \omega_2 x_2 \leq \omega_3) \geq 90\%,$$

where $\omega_1, \omega_2, \omega_3$ are independent Gaussian variables. The probability of satisfying this constraint is estimated using N samples and the indicator function $I(\cdot)$ as: $\frac{1}{N}\sum_{i=1}^{N} I(\omega_1^i x_1 + \omega_2^i x_2 \leq \omega_3^i) \geq 90\%$. The visuals underscore the **non-smooth** geometry and optimization difficulty (PAS09). We **remark** that BP was tested in a high-dimensional scenario with **400** decision variables in our experiments.

G.2.3 Alternating Current Optimal Power FLow (AC-OPF)

The Alternating Current Optimal Power Flow (AC-OPF) problem is pivotal in ensuring the efficient and safe operation of power grids. It requires real-time decision-making and adherence to operational constraints to maintain system integrity. The AC-OPF is inherently a non-convex Quadratically Constrained Quadratic Program (QCQP) and is recognized as NP-hard, posing significant computational challenges. The formal mathematical formulation of the AC-OPF problem is as follows:

$$\begin{aligned}
\textbf{AC-OPF}: \min_{p_g, q_g, v} \quad & p_g^\top Q p_g + b^\top p_g && (92)\\
\text{subject to} \quad & |v_i(\bar{v}_i - \bar{v}_j)\bar{w}_{ij}| \leq S_{ij}^{\max}, \quad \forall (i,j) \in \mathcal{E}, && (93)\\
& (p_g - p_d) + (q_g - q_d)\, i = \mathrm{diag}(v)\bar{W}\bar{v}, \quad \forall i \in \mathcal{N}, && (94)\\
& p_g^{\min} \leq p_g \leq p_g^{\max},\ q_g^{\min} \leq q_g \leq q_g^{\max},\ v^{\min} \leq |v| \leq v^{\max}. && (95)
\end{aligned}$$

where the power network comprises n nodes, indexed by the set $\mathcal{N}$. The vectors $p_d, q_d \in \mathbb{R}^n$ represent the real and reactive power demand at each node, respectively. The vectors $p_g, q_g \in \mathbb{R}^n$ denote the real and reactive power generation, which are the decision variables of the optimization problem. The vector $v \in \mathbb{C}^n$ signifies the nodal voltage phasors. 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 ($v^{\min}, v^{\max}$), thermal line limits ($S_{ij}^{\max}$), and power flow balance equations. The set $\mathcal{E}$ denotes the set of edges (transmission lines) connecting the nodes in the power network. Constraint (5) represents the complex power flow balance at each node, ensuring that the generation and demand are matched while accounting for power losses.

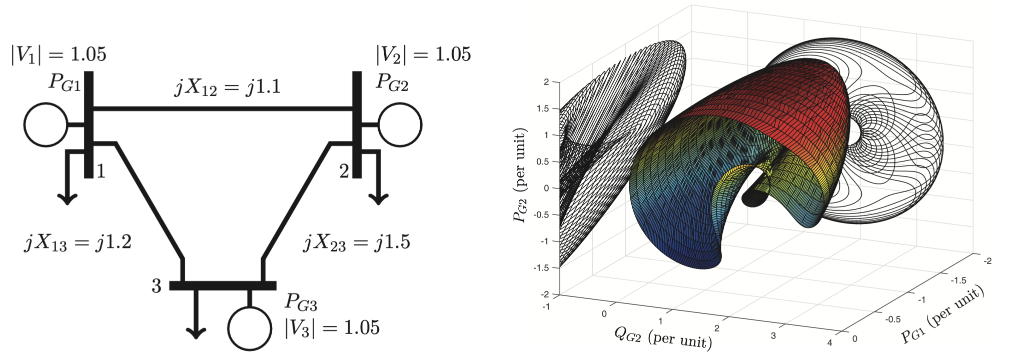


Figure 10: The **left** figure displays a simple 3-node power network, while the **right** figure illustrates a part of its constraint set (MH$^+$19). These visuals highlight the complex geometry and inherent challenges of the ACOPF problem. It is **noteworthy** that in our experiments, BP was tested on a **200-node** power network involving over 400 decision variables and 1000 constraints.

G.3 Neural Network structure and hyper-parameters for training

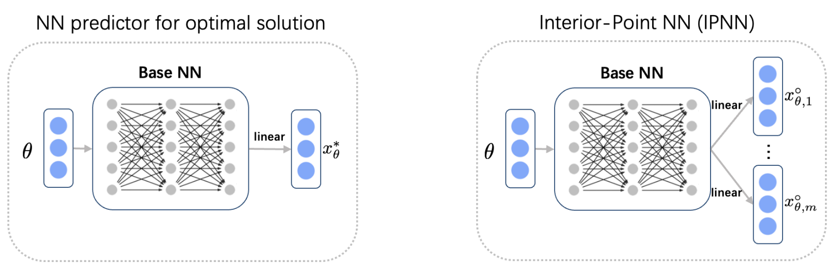


Figure 11: The **Left** one is the NN predictor for learning the input-to-solution mapping, where we use feed-forward NN structure with ReLU activation. The **Right** one is the IPNN used to predict input-dependent IPs, where all IP predictors share the same base network to embed the input parameters as $e = \mathrm{NN}(\theta)$, then embedding e is then passed by different linear layers to predictions as $x^\circ_{\theta,k} = W_k^\top e$ for $k = 1, \cdots, m$. In practice, two NNs can share the same based network for the parameter-efficient implementation.

G.3.1 NN predictor for input-to-solution mapping

We adopt the fully connected NN with residual connection and equality reconstruction, denoted as F, to predict the optimal solution for constrained optimization problems given the input parameters. We follow a supervised learning scheme to train the NN predictor. Firstly, to prepare the training and testing data, we collect the optimal solutions under various input parameters using an iterative solver (DRK20; LCL23). For convex optimization, we use MOSEK to solve the optimal solution. For AC-OPF problems, we adopt PYPOWER as the specialized solver (ZMSG97). For JCCIM problems, we solve its scenario approximated version with sampled uncertainty. Then, we apply the reconstruction technique to complete the decision variables and ensure equality satisfaction for the predicted solution with partial variables. 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 and the objective function. This can be represented as:

$$\min_F \ \mathbb{E}_\theta[\mathrm{MSE}(F(\theta), x^*_\theta) + \lambda_1 \|\mathrm{ReLU}(\mathbf{g}(F(\theta), \theta))\|_1 + \lambda_2 f(x, \theta)] \tag{96}$$

The second term stands for the penalty for inequality constraint violation by the NN-predictable solutions. For the equality constraint present in the optimization problem, we utilize the variable selection and completion techniques elaborated in Appendix A.1 to ensure its feasibility. By adjusting the penalty coefficient λ, we can train the NN predictor to achieve different qualities in terms of solution feasibility and optimality. The detailed parameters are in Table 6.

Table 6: Structure of NN predictor in experiments

Parameter	Value
NN structure	
dimension of input layer	d
dimension of output layer	n
dimension of hidden layer	$\lfloor (d+n)/2 \rfloor$
activation function	$\mathrm{ReLU}(\cdot)$
number of layer	3
last-layer activation	$\mathrm{Sigmoid}(\cdot)$ or None
NN training parameters	
number of training samples	10,000
number of testing samples	1,024
number of iteration	10,000
optimizer	Adam
learning rate	0.0001
batch size	64
the coefficient for objective function	0.001
the coefficient for inequality penalty	0.1

G.3.2 IPNN FOR INPUT-TO-IPS MAPPING

For the IPNN design and training, we apply a similar structure to the NN predictor. It has a shared base network for different IP predictor heads and applies different linear layers to map the embedding to the IPs. For each IP, we apply the reconstruction techniques to solve the full variables. The detailed parameters are in Table 7.

Table 7: Structure of IPNN predictor in experiments

Parameter	Value
IPNN training parameters	
number of training samples	10,000
number of iteration	10,000
number of interior points	{1,2,4,8}
number of boundary samples	10
bisection steps for boundary sampling	10
LSE parameter β	$\mathrm{iter}/5$
optimizer	Adam
learning rate	0.0001
batch size	64
the coefficient for eccentricity loss	0.1