5. Additional experiment

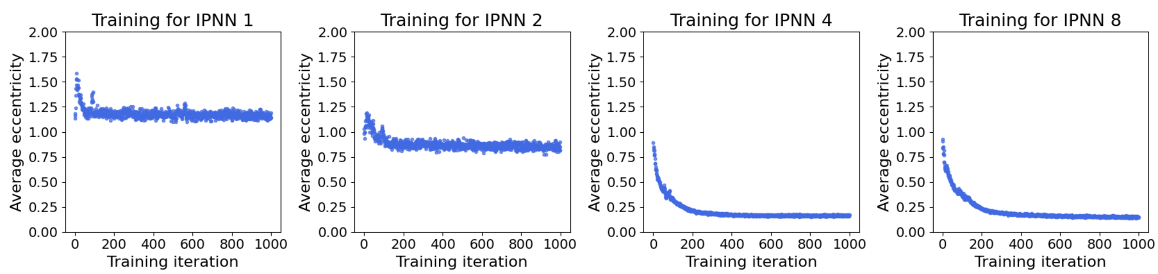
5.1. Benchmark Irregular Constraint Set

We construct the following non-trivial constraint set with explicit MEIPs as the benchmark to test different algorithms:

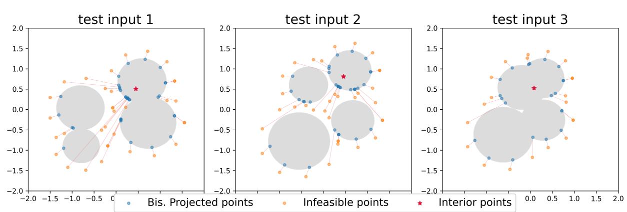
$$C_\theta = \cup_{i=1}^{4} \mathcal{B}(x_i, r_i) \tag{1}$$

where $x_i \in \mathbb{R}^2$, $r_i \in \mathbb{R}^+$, and the input parameter is defined as $\theta = \{x_i, r_i\}_{i=1}^{4}$. We remark that the geometry of this constraint set is highly irregular, non-convex, and may be disconnected, making it suitable for testing the BP framework. The minimum eccentricity of 4 IPs is zeros due to the symmetry of the balls.

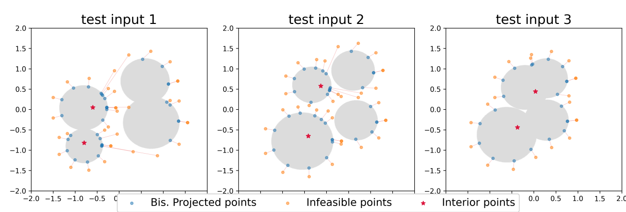
1. BP performance: We then visualize the IPNN output under test input parameters and the bisection protection trajectory given infeasible points.



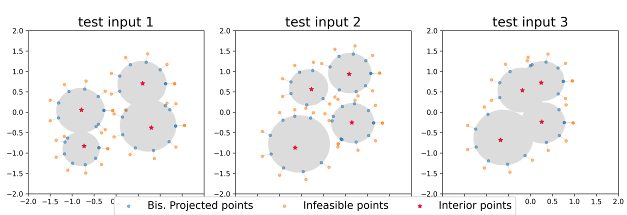
(a) Unsupervised IPNN training with different IP predictors.



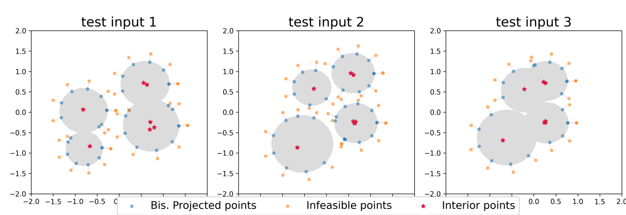
(b) IPNN with 1 IP predictors and average eccentricity of 0.89



(c) IPNN with 2 IP predictors and average eccentricity of 0.21.



(d) IPNN with 4 IP predictors and average eccentricity of 0.16.



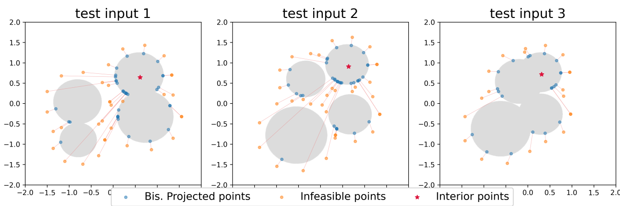
(e) IPNN with 8 IP predictors and average eccentricity of 0.14.

Figure 1: Training and testing IPNN for approximating MEIPs of non-convex sets.

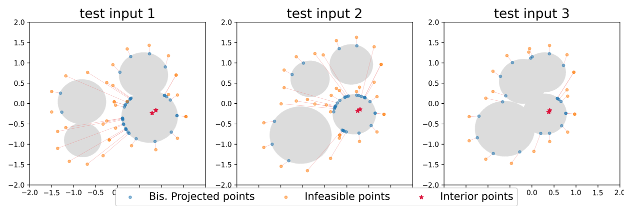
2. IPNN training without eccentricity loss term To validate the necessity of IPNN training with minimizing eccentricity measure, we train IPNN with penalty term only and visualize its performance as follows:

5.2. Sensitivity analysis for BP framework

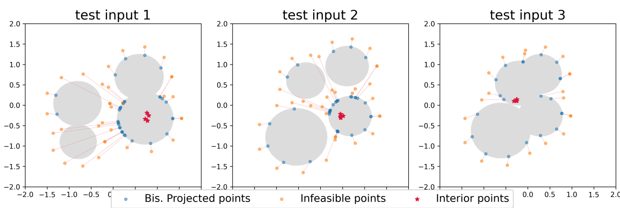
We present the sensitivity analysis for the BP framework under different numbers of IPs, random IPs, and NN structures, as well as the boundary sampling time statistics. Specifically, we select the non-convex, NP-hard JCCIM problem as the test bed to compare the sensitivity of different parameters. Further, we compare their performance in an adversarial setting with a poorly trained NN predictor to provide infeasible predictions. The performance is shown in the following Table.



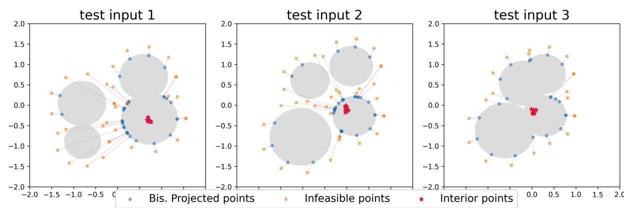
(a) IPNN with 1 IP predictors without minimizing eccentricity



(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 2: Training and testing IPNN without minimizing eccentricity of non-convex sets.

Table 1: Sensitivity analysis for BP framework.

	Feasibility		**Optimality**		**Speedup**		**Training**
	feas. rate (%)	ineq. vio.	sol. err. (%)	obj. gap (%)	total (×)	post. (×)	per iteration (s)
	JCC-IM: : $n = 400,\ d = 100,\ n_{\text{eq}} = 0,\ n_{\text{ineq}} = 10100$						
NN							
Random-IPNN 1							
Random-IPNN 2							
Random-IPNN 4							
Random-IPNN 8							
ME-IPNN 1							
ME-IPNN 2							
ME-IPNN 4							
ME-IPNN 8							

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

5.3. Verification for IPNN

First, recall the feasibility verification problem as:

$$\textbf{P1:} \quad \min_{\theta, x} t \tag{2}$$
$$\text{s.t. } g(x,\theta) \le t \tag{3}$$
$$x = \text{NN}(\theta) \tag{4}$$
$$\theta \in \Theta \tag{5}$$

The optimal objective t^* can be viewed as the worst-case constraint violation for the output of NN given arbitrary input $\theta \in \Theta$. In general, this problem is non-convex either due to the NN structure or non-convex constraint function.

We then apply the relaxation techniques to reformulate the problem to derive the upper bound of constraint violation, denoted t^*, to verify the feasibility of NN:

$$\textbf{P2:} \quad \min_{\theta, x} t \tag{6}$$
$$\text{s.t. } \hat{g}(x,\theta) \le t \tag{7}$$
$$x \in \text{Relax}(\text{NN}(\theta)) \tag{8}$$
$$\theta \in \Theta \tag{9}$$

Here we apply the relaxation (e.g., linear relaxation or SDP relaxation) for the ReLU-based NN and the convex restriction for the constraint function $g(x,\theta) \leq \hat{g}(x,\theta)$. Therefore, the relaxed problem is a convex programming in general and can be solved with polynomial complexity. We denote the optimal solution as $\hat{t}^*$, which is an upper bound of the worst-case constraint violation. The tightness of the relaxation depends on both NN reformulation and convex restriction, which might be problem-dependent for empirical experiments.

We then solve the verification problem for the following problem, and the $\hat{t}^*$ is given in the following table:

Table 2: The setting for verification

Problem	Dimension	IPNN Structure	Input constraint	Output constraint	NN relaxation	Constraint violation
QP	$\theta \in \mathbb{R}^{50}$ $x \in \mathbb{R}^{100}$	$\theta \in [\theta_l, \theta_u]$	$Ax = \theta$ $Gx \leq h$	Linear relaxation		
Convex QCQP	$\theta \in \mathbb{R}^{50}$ $x \in \mathbb{R}^{100}$	$\theta \in [\theta_l, \theta_u]$	$Ax = \theta$ $x^{\mathrm{T}} H_i x + g_i x \leq h_i$	Linear relaxation		
Non-convex JCCIM	$\theta \in \mathbb{R}^{50}$ $x \in \mathbb{R}^{100}$	$\theta \in [\theta_l, \theta_u]$	$\frac{1}{100}\sum_{k=1}^{100} \mathbf{I}(Ax \geq \theta + \omega_k) \geq 1 - \epsilon$ $Gx \leq h$	SDP		

References