WSNET: LEARNING TO WARM-START INTERIOR-POINT METHODS FOR NONLINEAR OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interior-point methods (IPMs) are widely used to solve constrained nonlinear programs (NLPs), but effective warm-starting remains challenging. Proximity to an optimal solution alone is insufficient: the initial primal-dual point must also be strictly interior and sufficiently central. We consider a family of NLPs with shared structure and varying parameters, and use the IPM trajectory of a previously solved reference instance to warm-start a new instance. We propose the Warm-Start Network (WSNet), which combines attention-based trajectory selection with neural adjustment. The selection stage identifies a promising iterate from the reference trajectory, and the adjustment stage refines its primal-dual variables for the new instance. This refinement combines a learned dual correction with finite-step residual minimization and complementarity-based losses that promote centrality. Under explicit optimization and regularity assumptions, we show that mini-batch training yields strictly interior primal-dual points for the training instances with high probability. Together with suitable finite-step recovery, this analysis provides sufficient conditions for membership in a prescribed central-path neighborhood. Experiments on a range of convex and nonconvex NLPs demonstrate that WSNet effectively accelerates IPOPT, reducing its iteration count by up to 59.9% and total solution time by up to 60.3%.

1 INTRODUCTION

Constrained decision problems in power systems (Zhu, 2015), robotics (Schaal & Atkeson, 2010), wireless communications (Gao et al., 2026), and machine learning (Cortes & Vapnik, 1995) frequently involve nonlinear objectives and constraints, making *nonlinear programs* (NLPs) a central modeling framework across science and engineering. The need to solve these problems reliably and at scale has established *interior-point methods* (IPMs) as a leading numerical approach. Their scope includes twice continuously differentiable NLPs, *quadratic programs* (QPs), *quadratically constrained quadratic programs* (QCQPs), and *second-order cone programs* (SOCPs).

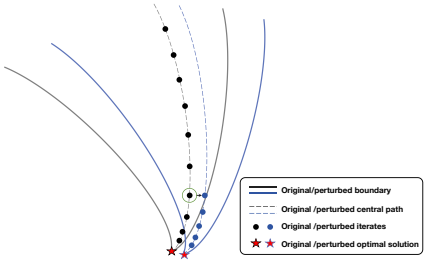


Figure 1: Central path and warm-starting in an interior-point method.

IPMs constitute a preeminent primal-dual framework for solving NLPs. Among their many variants, infeasible primal-dual path-following methods are among the most widely used and underpin mature solvers such as Gurobi (Gurobi Optimization, 2023), IPOPT (Wächter & Biegler, 2006), MOSEK (ApS, 2019), and CVXGEN (Mattingley & Boyd, 2012). As illustrated in Fig. 1, these methods generate a sequence of primal-dual iterates that tracks the central path, namely, the set of primal-dual points satisfying perturbed *Karush–Kuhn–Tucker* (KKT) conditions parameterized by a barrier parameter. By progressively reducing this parameter, they drive the iterates toward feasibility and optimality while maintaining sufficient centrality. This mechanism, however, also makes IPMs sensitive to initialization. An initial point that lies too close to the boundary or is poorly cen-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

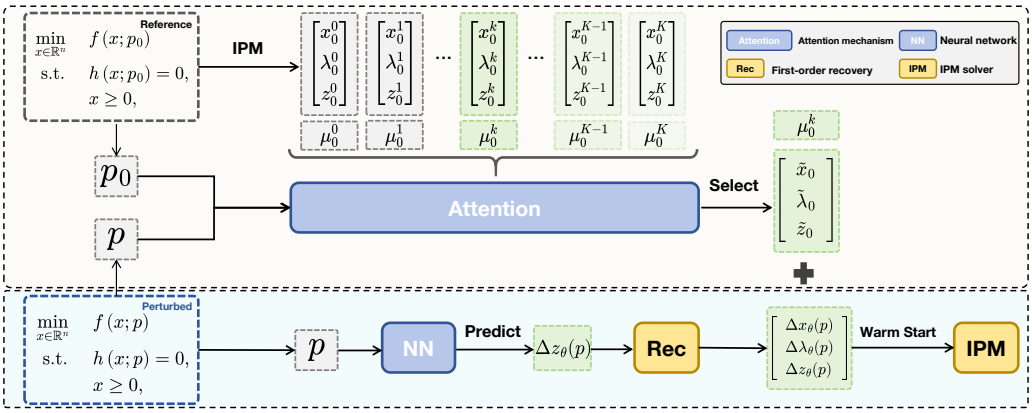


Figure 2: Overview of WSNet. The framework first selects an informative iterate from the reference IPM trajectory and then adjusts the selected primal-dual point before using it to warm-start the solver for the perturbed problem.

tered for the target instance can impede convergence and cause numerical instability. Warm-starting an IPM is therefore considerably more delicate than initializing many first-order or unconstrained optimization methods.

Traditional IPM warm-starting methods are often tailored to specific problem classes, and their effectiveness can deteriorate as perturbations grow (Yildirim & Wright, 2002; John & Yildirim, 2008; Gondzio & Grothey, 2008). Meanwhile, *Learning to Optimize* (L2O) has attracted increasing attention, with applications ranging from solution prediction (Bengio et al., 2021; Donti et al., 2021; Nguyen & Donti, 2026) and optimization modeling (Ramamonjison et al., 2023; Ahmaditeshnizi et al., 2024) to solver warm-starting (Sambharya et al., 2024; Gao et al., 2024; Suri et al., 2026). By exploiting problem parameters and previously computed solutions or trajectories, neural networks can learn to construct informative initializations for related instances. However, primal-only warm-starting approaches focus on approximating optimal primal solutions and do not, by themselves, ensure the primal-dual interiority and centrality needed for effective IPM initialization (Baker, 2019; Donti et al., 2021). For IPMs, *Learning to Warm Start* (L2WS) therefore requires balancing proximity to a solution with these solver-specific requirements. The cost of constructing the warm start is equally important: any reduction in solver time must outweigh this additional overhead to yield a net computational benefit.

In this study, we assume access to the primal-dual trajectory generated by an IPM for a reference problem and investigate how to exploit this information when solving a perturbed problem. We address this challenge with the Warm-Start Network (WSNet), a neural warm-starting framework for IPMs with two components, *selection* and *adjustment*, as illustrated in Fig. 2. The selection module identifies a promising iterate from the reference trajectory by modeling the relation between the reference and perturbed contexts. The adjustment module then refines the selected primal-dual point to better satisfy the central-neighborhood requirements of the perturbed problem. Specifically, WSNet predicts a dual-variable correction, recovers the remaining primal-dual updates through a residual-minimization procedure, and enforces centrality through a neighborhood-aware loss.

The distinct contributions of our work can be summarized as follows:

- **A novel L2WS framework.** Given the IPM trajectory of a reference problem, WSNet selects and adjusts an iterate to warm-start a fixed-dimensional perturbed problem. It combines attention-based trajectory selection with a centrality-aware adjustment mechanism, integrating neural prediction, finite-step residual recovery, and neighborhood-specific loss design.

- **Theoretical guarantees.** Under a restricted Jacobian condition on the empirical complementarity loss, we show that mini-batch training drives each training instance to strict primal-dual interiority with high probability. We further prove that finite-step recovery

yields a certificate that the resulting warm-start point lies in a prescribed central-path neighborhood.

- **Empirical results.** Across convex and nonconvex QP, QCQP, and SOCP benchmarks, WSNet reduces IPOPT iterations by up to 59.9% and total solution time by up to 60.3%. Additional experiments demonstrate generalization to larger contextual perturbations and show that the learned selector identifies effective intermediate trajectory iterates.

2 Related Work

Traditional warm-starting for IPMs. Early work on IPM warm-starting focused on *linear programs* (LPs) with perturbed data and fixed problem dimensions. Least-squares and Newton corrections were developed to transform iterates from the reference LP into warm-start points for the perturbed instance, together with iteration bounds in terms of the perturbation size and problem conditioning (Yildirim & Wright, 2002). Several such strategies were later implemented and compared, with iteration reductions found to be most reliable for small perturbations, whereas improvements in total runtime were less consistent (John & Yildirim, 2008). Subsequent work introduced sensitivity-based unblocking procedures that identify components responsible for short step sizes and modify them to improve the next IPM direction (Gondzio & Grothey, 2008). These ideas were extended to QPs, including sequential and model predictive control, where the solution of a preceding subproblem provides information for constructing a strictly feasible starting point (Shahzad et al., 2011; Shahzad & Goulart, 2011). For general NLPs, regularization and primal-dual penalty terms have been used to improve robustness and facilitate warm-starting (Benson & Shanno, 2007). Related strategies have also been developed for homogeneous self-dual formulations, SOCPs, and general conic programs (Skajaa et al., 2011; Cay et al., 2017; Chen et al., 2025). Despite these advances, traditional approaches typically rely on problem-specific correction rules, sensitivity calculations, or additional feasibility-restoration procedures, and their effectiveness can deteriorate as the perturbation grows.

L2WS for IPMs. Learning-based optimization methods have also been used to construct warm starts. A primal-only approach predicts an approximate solution from the problem data and supplies it to an IPM solver, as demonstrated for AC optimal power flow (Baker, 2019); similar end-to-end L2O methods can likewise provide useful initial points, but do not explicitly model the dual variables or the barrier parameter. In contrast, IPM-LSTM (Gao et al., 2024) learns approximate primal-dual solutions by replacing the conventional linear-system solve with a neural recovery procedure and then uses the resulting point to warm-start an IPM solver. More recent work has emphasized the importance of evaluating the complete primal-dual-barrier state rather than the primal variables alone (Suri et al., 2026). These studies demonstrate the potential of learning-based warm-starting, while also highlighting the centrality and dual-variable requirements that distinguish IPMs from first-order methods.

3 Preliminary

3.1 The Classic IPM

We consider the following continuous constrained optimization problem:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & f(x;p) \\ \text{s.t.} & h(x;p)=0, \\ & x\geq 0. \end{array} \tag{1}$$

Here, $x \in \mathbb{R}^n$ is the decision variable, while $p \in \mathbb{R}^d$ denotes the input contextual parameter. For each fixed p, the objective and equality constraints are represented by the mappings $f(\cdot; p) : \mathbb{R}^n \to \mathbb{R}$ and $h(\cdot; p) : \mathbb{R}^n \to \mathbb{R}^m$, respectively. Problems with general nonlinear inequality constraints can also be handled in this form by introducing slack variables. In this study, we assume access to a reference instance with contextual parameter p_0 and represent a perturbed instance by $p = p_0 + \Delta p$. Consequently, the reference and perturbed instances have the same numbers of variables and constraints.

For a fixed context p, one of the most widely used IPM variants for problem (1) is the *infeasible path-following primal-dual* method. It defines the central path $\mathcal{C} := \{(x(\mu), \lambda(\mu), z(\mu)) \mid \mu > 0\}$ as the set of primal-dual points satisfying the following perturbed KKT conditions:

$$\nabla f(x;p) + \nabla h(x;p)^\top \lambda - z = 0, \tag{2a}$$

$$h(x;p) = 0, \tag{2b}$$

$$\operatorname{diag}(z)\operatorname{diag}(x)e = \sigma\mu e, \tag{2c}$$

$$x, z \geq 0, \tag{2d}$$

where $\lambda \in \mathbb{R}^m$ and $z \in \mathbb{R}^n_+$ are the dual variables associated with the equality and nonnegativity constraints, respectively. Here, $\operatorname{diag}(\cdot)$ denotes a diagonal matrix, e is the all-ones vector, and $\sigma > 0$ is a centering constant. Equations (2a) and (2b) are the dual and primal feasibility equations, and their stacked associated residual is given by

$$\mathcal{R}(x, \lambda, z; p) := \begin{bmatrix} h(x;p) \\ \nabla f(x;p) + \nabla h(x;p)^\top \lambda - z \end{bmatrix}. \tag{3}$$

The infeasible path-following primal-dual IPM generates a sequence of interior iterates (x^k, λ^k, z^k) that satisfy (2d) but are not necessarily feasible. At iteration k, it applies a Newton correction to the perturbed KKT system and uses a line search to preserve $x^{k+1}, z^{k+1} > 0$. Meanwhile, the barrier parameter $\mu^k := [(x^k)^\top z^k]/n$ is progressively reduced, driving the iterates toward feasibility, complementarity, and optimality (Wächter & Biegler, 2006).

3.2 Central Neighborhoods in IPMs

Path-following IPMs are designed to follow the central path $\mathcal{C}$ toward the solution, but their iterates do not need to lie exactly on $\mathcal{C}$. Instead, they are maintained in a loose but well-defined neighborhood of $\mathcal{C}$ while the barrier parameter is steadily reduced to zero. In an infeasible path-following IPM, the generated iterates (x^k, λ^k, z^k) may have nonzero primal-dual residuals; what is required is that this residual remains controlled and vanishes as $\mu^k \to 0$. Different primal-dual IPMs impose different centrality requirements on the pairwise complementarity products $x_i z_i$. Two common choices are the following neighborhoods. Let $c^0 := \|\mathcal{R}(x^0, \lambda^0, z^0; p)\|/\mu^0$, where μ^0 is evaluated at the initial point (x^0, λ^0, z^0):

$$\mathcal{N}_2(\alpha, \beta) = \big\{(x, \lambda, z) \mid x, z > 0,\ \|\mathcal{R}(x, \lambda, z; p)\| \leq \beta c^0 \mu,\ \|\operatorname{diag}(x)\operatorname{diag}(z)e - \mu e\|_2 \leq \alpha\mu\big\}, \tag{4a}$$

$$\mathcal{N}_{-\infty}(\gamma, \beta) = \big\{(x, \lambda, z) \mid x, z > 0,\ \|\mathcal{R}(x, \lambda, z; p)\| \leq \beta c^0 \mu,\ x_i z_i \geq \gamma\mu,\ i = 1, \ldots, n\big\}. \tag{4b}$$

where $\alpha, \gamma \in (0, 1)$ and $\beta > 0$ are prescribed constants. The neighborhood $\mathcal{N}_2$ keeps all complementarity products as uniform as possible and close to their average value in a Euclidean sense, whereas $\mathcal{N}_{-\infty}$ prevents x and z from approaching the boundary of the region $(x, z) \geq 0$ prematurely. Membership in either neighborhood does not imply exact feasibility. Rather, it indicates that the point is strictly interior, sufficiently central, and has primal-dual residuals that decrease at the same order as the barrier parameter. This is precisely the kind of numerical state desired for warm-starting: an initial point that is close to a solution but poorly centered may still be difficult for an IPM to refine, whereas a well-centered point can be advanced reliably as μ decreases.

4 WSNet: A Neural Network for Warm-Starting IPMs

We now consider how to obtain a warm-start point for a perturbed instance p using a neural network. Suppose that the reference instance with parameter p_0 has already been processed by an IPM for K iterations, yielding the primal-dual trajectory $\{(x_0^k, \lambda_0^k, z_0^k, \mu_0^k)\}_{k=1}^K$. Since the reference and perturbed instances share the same numbers of variables and constraints, the reference trajectory provides a sequence of candidate primal-dual points for warm-starting the IPM applied to the perturbed instance. These candidates, however, are not equally useful: late-stage iterates may be close to the reference solution but too close to the boundary, while earlier iterates may be more central but less accurate for the perturbed problem. In this work, we therefore seek to select an appropriate candidate from the reference trajectory and adjust its primal-dual variables so that the resulting point lies in the perturbed central-path neighborhood.

4.1 Attention-Based Selection

We start with a simple experiment on the convex QP dataset. Each stored iterate from the reference instance p_0 is used independently to warm-start a perturbed instance p, and the resulting iteration-reduction ratios are shown in Fig. 3. Two closely related optimization problems should, in general, share similar characteristics. Thus, more advanced iterates often provide better warm starts because they are closer to the optimum. However, the optimal primal-dual solution of the reference problem is not necessarily the best warm-start point, since it may also lie closer to the boundary. A key task is therefore to select a suitable candidate according to the similarity between the reference and perturbed problems.

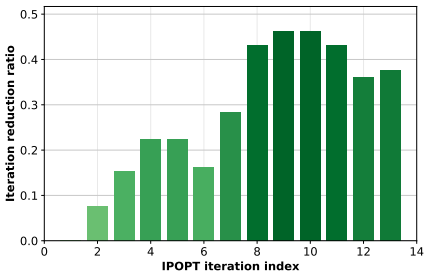


Figure 3: Warm-starting perturbed instances from the reference IPM trajectory on the convex QP dataset.

To implement this selection, we use an attention-based neural module that produces a score vector $\omega \in \mathbb{R}^K$ from the two contextual parameters:

$$\omega = \text{Gumbel-Softmax}\left(W_\theta \begin{bmatrix} p \\ p_0 \end{bmatrix}\right) \tag{5}$$

where $W_\theta \in \mathbb{R}^{K\times 2d}$ is a parameter matrix, included in the trainable neural parameters θ, that maps the concatenated contextual parameters to scores. The Gumbel–Softmax operation provides a differentiable approximation to categorical sampling, allowing the selection module to be trained by gradient-based optimization while approaching a hard, one-hot candidate selection during inference (Jang et al., 2017; Maddison et al., 2017). The explicit sampling equations are provided in Appendix C.3. The weight vector is then used to select, or softly combine during training, a candidate point from the reference IPM trajectory:

$$\left(\tilde{x}_0, \tilde{\lambda}_0, \tilde{z}_0, \tilde{\mu}_0\right) = \sum_{k=1}^{K} \omega_k \left(x_0^k, \lambda_0^k, z_0^k, \mu_0^k\right). \tag{6}$$

When ω is one-hot, this operation selects a single iterate; during training, a softened ω gives a differentiable relaxation of this discrete selection.

4.2 Neural Adjustment

The selection module identifies a promising iterate from the reference trajectory, but the selected point $(\tilde{x}_0, \tilde{\lambda}_0, \tilde{z}_0, \tilde{\mu}_0)$ is not necessarily acceptable for the perturbed instance p. It may lie outside a central-path neighborhood of p and therefore fail to provide a well-centered warm-start point. The role of the adjustment module is to apply a small correction $(\Delta x, \Delta\lambda, \Delta z)$ to the selected candidate $(\tilde{x}_0, \tilde{\lambda}_0, \tilde{z}_0)$, while retaining the selected barrier parameter $\tilde{\mu}_0$. According to the neighborhood definitions in (4a) and (4b), an acceptable warm-start point should satisfy three types of requirements:

- **Interiority:** x and z are strictly positive.
- **Boundedness:** The primal-dual residual norm $\|\mathcal{R}(x, \lambda, z; p)\|$ is bounded.
- **Centrality:** $x_i z_i$ are uniform or bounded away from the neighborhood boundary.

These requirements guide the design of the adjustment module. We begin with interiority and boundedness, which are enforced through projection and governed by the primal-dual residual equations, respectively. To exploit this structure, the adjustment stage first predicts the dual-slack correction:

$$\Delta z_\theta(p) = \text{NN}_\theta(p), \tag{7}$$

where NN_θ denotes a neural network with trainable parameters θ. The predicted $\Delta z_\theta(p)$ is then scaled so that its entries do not exceed the smallest entry of $\tilde{z}_0$ in magnitude, thereby preserving interiority of $\tilde{z}_0 + \Delta z_\theta(p)$. The specific scaling rule is given in Appendix C.2.

Given $\Delta z_\theta(p)$, the remaining updates Δx and $\Delta\lambda$ can be recovered from the primal and dual equations in (2b) and (2a). These equations form a possibly nonlinear system, which we handle through

the following unconstrained residual-minimization problem:

$$\min_{\Delta x\in\mathbb{R}^n,\,\Delta\lambda\in\mathbb{R}^m} \quad \phi(\Delta x,\Delta\lambda;p,\theta) := \left\|\mathcal{R}(\tilde{x}_0+\Delta x,\tilde{\lambda}_0+\Delta\lambda,\tilde{z}_0+\Delta z_\theta(p);p)\right\|_2^2. \tag{8}$$

In practice, the primal residual is decomposed into equality and inequality parts. Inequality violations are penalized through the squared positive-part function $[\cdot]_+ = \max(\cdot,0)$, rather than by introducing additional slack variables. This implementation is consistent with the stored primal-dual trajectory, which does not contain explicit slack-variable iterates.

Residual boundedness does not require solving (8) exactly. We therefore use a finite number of first-order steps to obtain an approximate recovery at low computational cost. The resulting finite-step recovery map is denoted by

$$(\Delta x_\theta(p),\Delta\lambda_\theta(p)) = \mathrm{Rec}_s(\Delta z_\theta(p);p). \tag{9}$$

where Rec_s denotes an s-step gradient-based recovery procedure. In our experiments, it is implemented using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm with the parameters reported in Appendix C. After $\Delta x_\theta(p)$ and $\Delta\lambda_\theta(p)$ are recovered, we enforce interiority by projecting $\tilde{x}_0+\Delta x_\theta(p)$ onto the nonnegative orthant. It remains to control centrality, which is determined by the pairwise complementarity products in (4a) and (4b). We incorporate this condition into the training objective. For a mini-batch $\mathcal{B}$, the two neighborhood definitions lead to the following two possible losses:

$$\mathcal{L}_2(\theta) = \frac{1}{2|\mathcal{B}|}\sum_{b\in\mathcal{B}}\Big(\|\mathrm{ReLU}(\tilde{x}_0+\Delta x_\theta(p))\odot\mathrm{ReLU}(\tilde{z}_0+\Delta z_\theta(p))-\tilde{\mu}_0 e\|_2^2\Big)_b, \tag{10}$$

$$\mathcal{L}_{-\infty}(\theta) = \frac{1}{2|\mathcal{B}|}\sum_{b\in\mathcal{B}}\Big(\big\|[\tilde{\mu}_0 e-\mathrm{ReLU}(\tilde{x}_0+\Delta x_\theta(p))\odot\mathrm{ReLU}(\tilde{z}_0+\Delta z_\theta(p))]_+\big\|_2^2\Big)_b. \tag{11}$$

Here, $\odot$ denotes componentwise multiplication. These two losses correspond to the two central-neighborhood properties used in IPMs: after training, the adjusted primal-dual variables are encouraged either to maintain uniform complementarity products or to stay sufficiently away from the boundary. The complete training procedure is summarized in Algorithm 1. Consequently, the adjustment module converts the selected iterate into a well-centered warm-start point for the perturbed instance by addressing interiority through projection, boundedness through finite-step residual recovery, and centrality through the complementarity loss.

Algorithm 1 WSNet Training Algorithm

1: **init.** NN weights θ
2: **repeat**
3: **input** perturbed context p
4: **select** $(\tilde{x}_0,\tilde{\lambda}_0,\tilde{z}_0,\tilde{\mu}_0)$ via (6)
5: **predict** $\Delta z_\theta(p)$ via (7)
6: **recover** $\Delta x_\theta(p),\Delta\lambda_\theta(p)$ via (9)
7: **update** θ via loss function of (10) or (11)
8: **until** convergence

5 Theoretical Analysis

We establish sufficient conditions for WSNet to produce warm-start points in a prescribed central neighborhood. Let $\boldsymbol{l}(\theta)\in\mathbb{R}^{n|\mathcal{B}|}$ collect the sample-wise complementarity residuals associated with either loss in (10) or (11). The corresponding empirical objective is

$$\mathcal{L}(\theta) := \frac{1}{2|\mathcal{B}|}\|\boldsymbol{l}(\theta)\|_2^2. \tag{12}$$

The training and recovery guarantees rely on the following assumptions.

Assumption 5.1.

(A1) On the training region, $\mathcal{L}$ has an L_{loss}-Lipschitz gradient.

(A2) $\inf_\theta \mathcal{L}(\theta)=0$, and the Jacobian $J_l(\theta) := \nabla_\theta \boldsymbol{l}(\theta)$ satisfies, throughout the training region and for some $\lambda_J>0$,

$$\|J_l(\theta)^\top \boldsymbol{l}(\theta)\|_2^2 \geq \lambda_J\|\boldsymbol{l}(\theta)\|_2^2. \tag{13}$$

(A3) At each training epoch $t = 1, \ldots, T$, the mini-batch gradient g_t satisfies $\mathbb{E}_t[g_t] = \nabla\mathcal{L}(\theta_t)$ and $\mathbb{E}_t\|g_t - \nabla\mathcal{L}(\theta_t)\|_2^2 \le \sigma^2$.

(A4) For every training instance, $\inf_{\Delta x,\, \Delta\lambda} \phi(\Delta x, \Delta\lambda; p, \theta) = 0$.

The restricted Jacobian condition in (A2) yields a PL inequality for the empirical loss.

Theorem 5.1 (Training-loss bounds)**.** Suppose Assumption 5.1 holds. For any $\eta, \delta \in (0, 1)$, let $\tilde{\mu}_{\min} := \min_{b\in\mathcal{B}} \tilde{\mu}_{0,b} > 0$ and $\tau := \eta^2\tilde{\mu}_{\min}^2/(2|\mathcal{B}|)$. There exist a positive step size and a finite number of mini-batch SGD epochs T such that $\mathcal{L}(\theta_T) < \tau$ with probability at least $1 - \delta$. For either loss in (10) or (11), the adjusted primal-dual variables of every training instance are then strictly positive and satisfy the corresponding pairwise centrality condition.

The proof is given in Appendix D.1. Theorem 5.1 controls the empirical complementarity error and establishes strict interiority with high probability. The following corollary combines this training guarantee with the recovery convergence bound. It gives sufficient conditions under which a finite-step WSNet output lies in the desired central neighborhood.

Corollary 5.2 (Finite-step recovery certificate)**.** Suppose that Assumption 5.1 holds and that ϕ has an L_ϕ-Lipschitz gradient and satisfies the PL condition with constant $\mu_\phi > 0$. Then the number of recovery steps can be chosen finite and sufficiently large so that, with probability at least $1 - \delta$, the following conclusions hold:

$$\begin{bmatrix} \tilde{x}_0 + \Delta x_\theta(p) \\ \tilde{\lambda}_0 + \Delta\lambda_\theta(p) \\ \tilde{z}_0 + \Delta z_\theta(p) \end{bmatrix} \in \begin{cases} \mathcal{N}_2(\alpha_2, \beta), & \text{for (10) if } \eta \le \dfrac{\alpha_2}{1 + \alpha_2/\sqrt{n}}, \\ \mathcal{N}_{-\infty}((1-\eta)/\kappa, \beta), & \text{for (11) if } \tilde{\mu} \le \kappa\tilde{\mu}_0, \quad \kappa \ge 1. \end{cases} \tag{14}$$

6 Experiments

6.1 Experimental Settings

We evaluate `WSNet` on a diverse collection of convex and nonconvex NLPs. For each class, the primal-dual trajectory of the reference problem is generated using `IPOPT` (Wächter & Biegler, 2006), whose flexibility allows it to handle all problem classes considered in our experiments within a common solver framework. Our primary focus is warm-start performance, measured by the reductions in solver iterations and total solution time relative to cold-start `IPOPT`. We also report the resulting objective values and primal constraint violations to assess the quality of the final solutions.

Datasets. We consider convex QP, QCQP, and SOCP benchmarks, together with nonconvex variants obtained by applying an elementwise sine transformation to the objective while retaining the corresponding convex constraints. The precise problem formulations are provided in Appendix A. For each problem class, we generate 10,000 samples and split them into training, validation, and test sets in a $10 : 1 : 1$ ratio, respectively. All numerical results are reported on the test set.

Baseline Algorithms. We compare `WSNet` with `IPOPT`, two L2O baselines, and a selection-only ablation. `IPOPT` serves as both the numerical reference and the downstream warm-start solver. `DC3` (Donti et al., 2021) predicts a partial primal solution, completes it through the equality constraints, and iteratively corrects inequality violations. `FSNet` (Nguyen & Donti, 2026) embeds a differentiable feasibility-seeking procedure into neural-network training to produce nearly feasible primal solutions. `AttNet` retains only the selection stage of `WSNet` and directly initializes `IPOPT` with the selected primal-dual iterate.

Evaluation Configuration. All our experiments were conducted on an NVIDIA RTX A6000 GPU, an Intel Xeon 2.10GHz CPU, using Python 3.10.0 and PyTorch 1.13.1.

6.2 Computational Results

Table 1 summarizes our comparison of `WSNet` with the baseline methods in terms of end-to-end solution quality and warm-start performance across a range of convex and nonconvex problems. Unless otherwise specified, the numbers of variables, inequality constraints, and equality constraints are fixed at 1000, 500, and 500, respectively. We use $\texttt{WSNet}_2$ and $\texttt{WSNet}_{-\infty}$ to denote the models trained with the losses in (10) and (11), respectively. For a fair comparison with conventional solvers,

Table 1: Computational results on NLP benchmarks.

Method	End-to-End						IPOPT (warm start)		Total Time (s)↓	Gain (Ite./Time)↑
	Obj. ↓	Max ineq. ↓	Mean ineq. ↓	Max eq. ↓	Mean eq. ↓	Time (s) ↓	Ite. ↓	Time (s) ↓		
Convex QPs: $n = 1000, n_{\text{ineq}} = 500, n_{\text{eq}} = 500$										
IPOPT	-162.793	0.000	0.000	0.000	0.000	32.053	13.08	-	-	-
DC3	-147.701	1.322	0.008	0.000	0.000	0.027	11.00	30.009	30.036	15.9%/6.3%
FSNet	-162.176	0.000	0.000	0.000	0.000	0.407	8.96	24.831	25.238	31.5%/21.3%
AttNet	-158.214	0.000	0.000	1.934	0.672	0.009	10.90	31.756	31.765	16.7%/0.9%
$\text{WSNet}_{-\infty}$	-161.898	0.006	0.000	0.008	0.002	1.053	6.57	14.318	15.371	**49.8%**/52.0%
WSNet_2	-161.870	0.012	0.000	0.004	0.001	1.003	7.13	13.825	14.828	45.5%/**53.7%**
Convex QCQPs: $n = 1000, n_{\text{ineq}} = 500, n_{\text{eq}} = 500$										
IPOPT	-186.986	0.000	0.000	0.000	0.000	90.632	13.18	-	-	-
DC3	-70.098	2.468	0.018	0.000	0.000	0.032	13.14	106.594	106.626	0.3%/-17.6%
FSNet	-185.995	0.000	0.000	0.000	0.000	0.462	9.42	68.153	68.615	28.5%/24.3%
AttNet	-99.598	0.352	0.001	1.921	0.661	0.009	10.86	71.227	71.236	17.6%/21.4%
$\text{WSNet}_{-\infty}$	-182.087	0.012	0.000	0.007	0.002	1.723	8.41	60.671	62.394	36.2%/31.2%
WSNet_2	-185.823	0.006	0.000	0.002	0.001	1.643	6.92	48.441	50.084	**47.5%/44.7%**
Convex SOCPs: $n = 1000, n_{\text{ineq}} = 500, n_{\text{eq}} = 500$										
IPOPT	-147.963	0.000	0.000	0.000	0.000	63.190	16.26	-	-	-
DC3	-122.961	2.250	0.096	0.000	0.000	0.047	14.36	57.266	57.313	11.7%/9.3%
FSNet	-145.874	0.000	0.000	0.000	0.000	0.629	11.78	51.060	51.689	27.6%/18.2%
AttNet	-111.494	0.000	0.000	1.916	0.670	0.009	13.04	45.506	45.515	19.8%/28.0%
$\text{WSNet}_{-\infty}$	-145.361	0.003	0.000	0.003	0.001	2.338	9.04	33.081	35.419	44.4%/44.0%
WSNet_2	-147.524	0.000	0.000	0.001	0.000	1.980	6.52	23.126	25.106	**59.9%/60.3%**
Nonconvex QPs: $n = 1000, n_{\text{ineq}} = 500, n_{\text{eq}} = 500$										
IPOPT	-124.601	0.000	0.000	0.000	0.000	54.535	19.77	-	-	-
DC3	-112.778	1.164	0.007	0.000	0.000	0.026	15.86	46.477	46.503	19.8%/14.7%
FSNet	-123.834	0.000	0.000	0.000	0.000	0.405	12.90	40.144	40.549	34.7%/25.6%
AttNet	-23.922	0.000	0.000	1.933	0.671	0.010	14.95	40.196	40.206	24.4%/26.3%
$\text{WSNet}_{-\infty}$	-123.871	0.012	0.000	0.005	0.001	1.009	11.76	27.854	28.863	40.5%/47.1%
WSNet_2	-124.043	0.013	0.000	0.005	0.001	1.006	11.35	24.758	25.764	**42.6%/52.8%**
Nonconvex QCQPs: $n = 1000, n_{\text{ineq}} = 500, n_{\text{eq}} = 500$										
IPOPT	-158.229	0.000	0.000	0.000	0.000	149.670	20.18	-	-	-
DC3	-143.089	0.786	0.016	0.000	0.000	0.031	18.12	152.465	152.496	10.2%/-1.9%
FSNet	-157.442	0.000	0.000	0.000	0.000	0.365	15.98	126.389	126.754	20.8%/15.3%
AttNet	-156.635	0.011	0.000	0.956	0.333	0.008	16.22	114.660	114.668	19.6%/23.4%
$\text{WSNet}_{-\infty}$	-153.461	0.022	0.000	0.008	0.002	1.460	17.81	117.255	118.715	11.7%/20.7%
WSNet_2	-157.591	0.005	0.000	0.002	0.001	1.533	13.29	93.586	95.119	**34.1%/36.4%**
Nonconvex SOCPs: $n = 1000, n_{\text{ineq}} = 500, n_{\text{eq}} = 500$										
IPOPT	-123.537	0.000	0.000	0.000	0.000	74.051	18.55	-	-	-
DC3	-113.572	0.628	0.035	0.000	0.000	0.047	16.44	70.005	70.052	11.4%/5.4%
FSNet	-121.855	0.000	0.000	0.000	0.000	0.655	14.58	37.703	38.358	21.4%/48.2%
AttNet	-31.914	0.000	0.000	1.932	0.670	0.008	16.64	53.896	53.904	10.3%/27.2%
$\text{WSNet}_{-\infty}$	-115.549	0.032	0.000	0.011	0.003	2.258	11.37	40.436	42.694	38.7%/42.3%
WSNet_2	-123.070	0.001	0.000	0.002	0.000	2.253	10.73	38.466	40.719	**42.2%/45.0%**

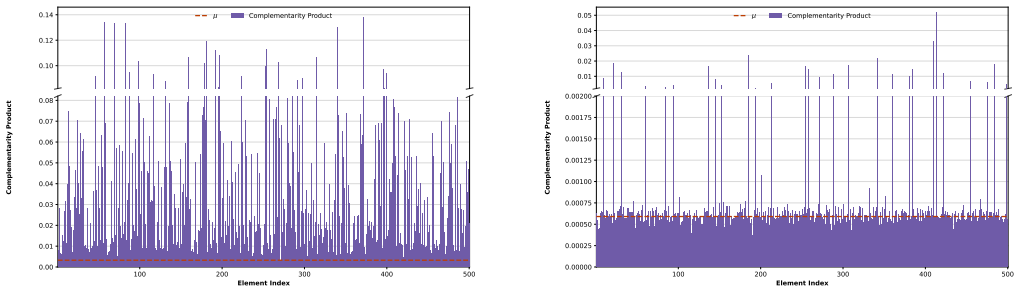
all runtimes reported here are measured under sequential execution. Under the "End-to-End" heading, we report the objective value, the maximum and mean inequality violations, the maximum and mean equality violations, and the model inference time. For warm-start performance, we report the number of IPOPT iterations, the IPOPT solution time after warm-starting, the cumulative time for constructing and using the warm start, and the gains relative to a cold start.

End-to-End Performance. From an end-to-end perspective, IPOPT provides the numerical reference, attaining optima on the convex benchmarks and stationary points on the nonconvex ones, while all L2O methods produce predictions much faster. Through its "completion" step, DC3 satisfies the equality constraints by construction; however, its first-order "correction" leaves noticeable inequality violations and objective gaps. The feasibility correction in FSNet, implemented using L-BFGS, yields nearly feasible primal solutions with smaller objective gaps, at a higher inference cost than DC3. By merely selecting a stored primal-dual point, AttNet is the fastest L2O method, but that point is interior only for the reference instance and thus incurs relatively large equality violations on the perturbed one. WSNet, especially WSNet_2, approaches FSNet in primal objective quality. Its finite-step first-order recovery must adjust the complete primal-dual state, resulting in longer inference and mild feasibility violations. The one-sided loss in (11) trains $\text{WSNet}_{-\infty}$ to avoid the boundary of the central neighborhood, whereas (10) drives WSNet_2 toward the central path; accordingly, the former generally has weaker primal solution quality.

Warm-Start Performance. The warm-start results reveal a different picture. Since DC3 supplies only a relatively low-quality primal point, it provides limited acceleration for IPOPT and even increases the total solution time on the convex and nonconvex QCQP benchmarks. The higher-quality

primal predictions of FSNet yield stronger gains: for example, they reduce IPOPT iterations by 31.5% on the convex QP benchmark and total solution time by 28.0% on the convex QCQP benchmark. Despite its poorer end-to-end feasibility, AttNet often performs comparably to FSNet because it supplies IPOPT with complete primal-dual information from a stored trajectory point. WSNet further adjusts this selected state to better match the perturbed problem. Although this dual-variable recovery incurs additional inference cost and may yield slightly weaker end-to-end solution quality than FSNet, it leads to stronger warm-start gains and lower cumulative solution times in most benchmark classes. In particular, WSNet reduces the iteration count by up to 59.9% and the total solution time by up to 60.3% on the convex SOCP benchmark. Although $\texttt{WSNet}_{-\infty}$ and $\texttt{WSNet}_2$ differ in end-to-end primal quality, their warm-start gains are generally comparable, and neither loss uniformly dominates: the variants alternate in attaining the largest iteration and total-time reductions.

6.3 Centrality



(a) $\texttt{WSNet}_{-\infty}$ (b) $\texttt{WSNet}_2$

Figure 4: Distributions of the complementarity products $x_i z_i$ on the convex SOCP benchmark.

The uniformity of the pairwise complementarity products $x_i z_i$ is critical to warm-start quality, since a poorly centered initialization can impede an IPM even when it is close to a solution. We therefore examine whether the two neighborhood-informed losses induce their intended complementarity structures. Figure 4 plots the products generated by $\texttt{WSNet}_{-\infty}$ and $\texttt{WSNet}_2$ on the convex SOCP benchmark. A clear distinction emerges: $\texttt{WSNet}_2$ produces a substantially more uniform distribution, with a variance of 1.464×10^{-5} versus 7.152×10^{-4} for $\texttt{WSNet}_{-\infty}$. This contrast is consistent with the underlying neighborhood definitions. The $\mathcal{N}_2$ neighborhood in (4a) controls the deviations of all $x_i z_i$ from μ, thereby encouraging the products to cluster around their mean, whereas $\mathcal{N}_{-\infty}$ in (4b) imposes only the lower bound $x_i z_i \geq \gamma\mu$ and permits substantially greater variation above it. Their locations relative to μ provide further evidence: most products in Fig. 4(a) lie above μ, while those in Fig. 4(b) are tightly concentrated around μ, with most lying slightly below it. Taken together, these observations show that the losses in (10) and (11) effectively impart the centrality characteristics of their respective neighborhoods to the corresponding outputs. The greater uniformity achieved by $\texttt{WSNet}_2$ also provides a plausible explanation for its stronger warm-start performance on most benchmarks in Table 1.

7 Conclusion

This paper introduced WSNet, a learning-based framework that reuses the primal-dual trajectory of a reference problem to warm-start IPMs for perturbed NLPs. By combining trajectory selection with structure-aware primal-dual adjustment, WSNet constructs interior and well-centered initializations while retaining the reliability of a numerical solver. Our analysis provides conditions under which the adjusted outputs lie in a prescribed central neighborhood, and the experiments demonstrate consistent acceleration across a range of convex and nonconvex problems. The current guarantees rely on strong optimization and regularity assumptions and do not characterize generalization; relaxing these assumptions and extending the evaluation to real-world applications are important directions for future work.

AI USE STATEMENT

Generative AI tools were used to assist with translation and language editing. All AI-assisted content was reviewed by the authors, who take responsibility for the final manuscript.

REFERENCES

Ali Ahmaditeshnizi, Wenzhi Gao, and Madeleine Udell. OptiMUS: Scalable optimization modeling with (MI)LP solvers and large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 577–596. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ahmaditeshnizi24a.html.

Mosek ApS. Mosek optimization toolbox for matlab. *User's Guide and Reference Manual, Version*, 4(1), 2019.

Kyri Baker. Learning warm-start points for ac optimal power flow. *arXiv preprint arXiv:1905.08860*, 2019.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421, 2021.

Hande Y. Benson and David F. Shanno. Interior-point methods for nonconvex nonlinear programming: Regularization and warmstarts. *Computational Optimization and Applications*, 38:371–399, 2007.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Sertalp B. Cay, Imre Polik, and Tamas Terlaky. Warm-start of interior point methods for second order cone optimization via rounding over optimal jordan frames. Technical Report 17T-006, Lehigh University, 2017.

Yuwen Chen, Paul Goulart, and Colin Jones. A warmstarting technique for general conic optimization in interior point methods. *arXiv preprint arXiv:2512.00693*, 2025.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

Xi Gao, Jinxin Xiong, Akang Wang, Qihong Duan, Jiang Xue, and Qingjiang Shi. Ipm-lstm: A learning-based interior point method for solving nonlinear programs. *Advances in Neural Information Processing Systems*, 37:122891–122916, 2024.

Xi Gao, Akang Wang, Junkai Zhang, Qihong Duan, and Jiang Xue. An accelerated mixed weighted-unweighted mmse approach for mu-mimo beamforming. *IEEE Transactions on Signal Processing*, 2026.

Jacek Gondzio and Andreas Grothey. A new unblocking technique to warmstart interior point methods based on sensitivity analysis. *SIAM Journal on Optimization*, 19(3):1184–1210, 2008. doi: 10.1137/060678129.

LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual*, 2023. URL https://www.gurobi.com/documentation/.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel–softmax. In *International Conference on Learning Representations*, 2017.

Elizabeth John and E. Alper Yildirim. Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, 41:151–183, 2008. doi: 10.1007/s10589-007-9096-y.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak–łojasiewicz condition. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Jacob Mattingley and Stephen Boyd. Cvxgen: A code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

Hoang Nguyen and Priya Donti. Fsnet: Feasibility-seeking neural network for constrained optimization with guarantees. *Advances in Neural Information Processing Systems*, 38:39670–39708, 2026.

Rindranirina Ramamonjison, Timothy Yu, Raymond Li, Haley Li, Giuseppe Carenini, Bissan Ghaddar, Shiqi He, Mahdi Mostajabdaveh, Amin Banitalebi-Dehkordi, Zirui Zhou, et al. Nl4opt competition: Formulating optimization problems based on their natural language descriptions. In *NeurIPS 2022 competition track*, pp. 189–203. PMLR, 2023.

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

Stefan Schaal and Christopher G. Atkeson. Learning control in robotics. *IEEE Robotics & Automation Magazine*, 17(2):20–29, 2010. doi: 10.1109/MRA.2010.936957.

Amir Shahzad and Paul J. Goulart. A new hot-start interior-point method for model predictive control. In *Proceedings of the 18th IFAC World Congress*, 2011.

Amir Shahzad, Eric C. Kerrigan, and George A. Constantinides. A warm-start interior-point method for predictive control. In *Proceedings of the 18th IFAC World Congress*, 2011.

Anders Skajaa, Erling D. Andersen, and Yinyu Ye. Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems. *Mathematical Programming Computation*, 2011.

Dhruv Suri, Helgi Hilmarsson, and Shourya Bose. Warp: A benchmark for primal-dual warm-starting of interior-point solvers. *arXiv preprint arXiv:2605.05728*, 2026.

Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1): 25–57, 2006.

E. Alper Yildirim and Stephen J. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3):782–810, 2002. doi: 10.1137/S1052623400369235.

Jizhong Zhu. *Optimization of Power System Operation*. John Wiley & Sons, 2 edition, 2015.

A Benchmark Problem Formulations

We consider six synthetic benchmark classes: convex and nonconvex QPs, QCQPs, and SOCPs. Let $x \in \mathbb{R}^n$ denote the decision variable, let $p \in \mathbb{R}^m$ denote the contextual parameter, and let $q \in \mathbb{R}^n$ denote the linear objective coefficient vector. For each generated dataset, the structural coefficients are sampled once and shared across all instances, whereas p is resampled to produce different instances. For compactness, $\sin(x)$ acts elementwise. The six benchmark formulations are given below.

Convex QP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \tfrac{1}{2}x^\top Qx + q^\top x \\ \text{s.t.} \quad & Ax = p, \\ & Gx \le c. \end{aligned} \tag{15}$$

Convex QCQP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \tfrac{1}{2}x^\top Qx + q^\top x \\ \text{s.t.} \quad & Ax = p, \\ & \tfrac{1}{2}x^\top H_i x + G_i^\top x \le c_i, \quad i = 1,\dots,r. \end{aligned} \tag{16}$$

Convex SOCP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & q^\top x \\ \text{s.t.} \quad & Ax = p, \\ & \|G_i x + h_i\|_2 \le C_i^\top x + d_i, \quad i = 1,\dots,r. \end{aligned} \tag{17}$$

The nonconvex variants modify only the objective through an elementwise sine transformation; their equality and inequality constraints remain identical to those of the corresponding convex templates.

Nonconvex QP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \tfrac{1}{2}x^\top Qx + q^\top \sin(x) \\ \text{s.t.} \quad & Ax = p, \\ & Gx \le c. \end{aligned} \tag{18}$$

Nonconvex QCQP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \tfrac{1}{2}x^\top Qx + q^\top \sin(x) \\ \text{s.t.} \quad & Ax = p, \\ & \tfrac{1}{2}x^\top H_i x + G_i^\top x \le c_i, \quad i = 1,\dots,r. \end{aligned} \tag{19}$$

Nonconvex SOCP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \tfrac{1}{2}x^\top Qx + q^\top \sin(x) \\ \text{s.t.} \quad & Ax = p, \\ & \|G_i x + h_i\|_2 \le C_i^\top x + d_i, \quad i = 1,\dots,r. \end{aligned} \tag{20}$$

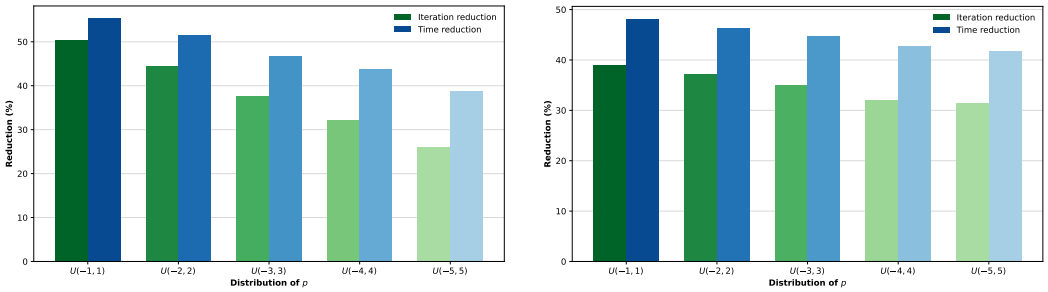
The first three formulations are convex because Q is positive semidefinite and, for QCQPs, each H_i is positive semidefinite. Their nonconvex counterparts retain the same constraints and coefficient distributions but use the sine-transformed objective described above.

B More experiment results

B.1 Generalization and Sensitivity

We assess the generalization and sensitivity of a fixed `WSNet` by applying it, without retraining, to instances that match the dimensions of the reference problem but exhibit progressively larger perturbations in p. For each test distribution, we average the warm-start performance over 100 instances; the resulting reductions in `IPOPT` iteration count and total solution time are shown in Fig. 5. We consider convex QP and nonconvex SOCP benchmarks, each with 1,000 variables, 500 inequality constraints, and 500 equality constraints. Specifically, Fig. 5(a) evaluates $\texttt{WSNet}_{-\infty}$ on the convex QP benchmark, whereas Fig. 5(b) evaluates $\texttt{WSNet}_2$ on the nonconvex SOCP benchmark. During training, the entries of both the reference context p_0 and the perturbed context p are sampled from $\mathcal{U}(-1,1)$. At test time, we progressively expand the distribution of p to $\mathcal{U}(-5,5)$, corresponding to a fivefold increase in the scale of the contextual perturbation relative to the training setting.

As the distribution shift increases, the warm-start gains of both variants gradually decline, with a more pronounced degradation for $\texttt{WSNet}_{-\infty}$. Nevertheless, on the convex QP benchmark, $\texttt{WSNet}_{-\infty}$ still achieves at least a 25% reduction in iteration count and nearly a 40% reduction in total solution time under the largest perturbation range. The degradation of $\texttt{WSNet}_2$ on the nonconvex SOCP benchmark is milder: even when the perturbation range is expanded fivefold, it retains approximately a 32% reduction in iteration count and a 42% reduction in total solution time. This contrast suggests that the $\mathcal{N}_{-\infty}$-based loss in (11) is less robust to contextual perturbations than the $\mathcal{N}_2$-based loss in (10). The former penalizes only complementarity products that fall below a prescribed threshold, allowing greater variation above it, whereas the latter directly penalizes deviations from the central-path target and thereby promotes more uniform complementarity. The resulting tighter centrality provides a plausible explanation for the greater stability of $\texttt{WSNet}_2$ under distribution shift. Overall, the results demonstrate the generalization and robustness of WSNet to contextual perturbations.



(a) $\texttt{WSNet}_{-\infty}$ on convex QP (b) $\texttt{WSNet}_2$ on nonconvex SOCP

Figure 5: Performance of WSNet under different distributions of p.

B.2 Selection Quality

We further evaluate the quality of the iterates identified by the selection module. For each benchmark class, we apply $\texttt{WSNet}_2$, trained with the loss in (10), to 100 perturbed instances and compare its average selected iterate with an oracle choice. The oracle is obtained by using every stored iterate from the trajectory of p_0 to warm-start each perturbed instance and selecting the one that yields the best warm-start performance. Figure 6 reports the average iterate indices. The line with circular markers denotes the length of the reference trajectory, while the left and right bars represent the selections made by $\texttt{WSNet}_2$ and the oracle, respectively. Although the learned selections do not always coincide exactly with the oracle choices, they generally identify similar regions of the trajectory. Moreover, $\texttt{WSNet}_2$ does not select the final iterate in any benchmark class, but instead consistently favors relatively advanced, nonterminal iterates. This behavior indicates that the selection module learns to balance proximity to the solution with the greater interiority typically retained by intermediate iterates.

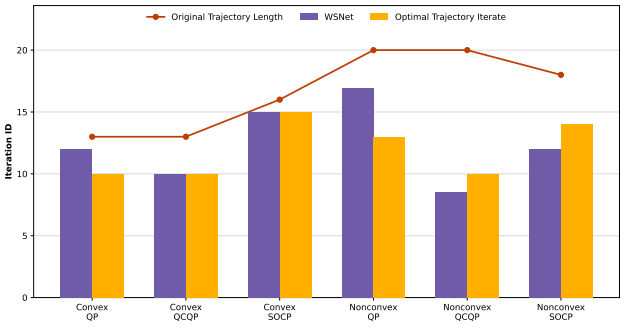


Figure 6: Selected and oracle trajectory iterates across benchmark classes.

C Implementation Details

C.1 Hyperparameters

Table 2 summarizes the hyperparameters used for WSNet. The selection module computes learned similarity scores and uses hard Gumbel–Softmax sampling. The complete model is trained with Adam, and checkpoints are selected according to the warm-start iteration count on the validation set. The recovery map Rec_s in (9) is implemented by finite-step L-BFGS with zero initial corrections and a backtracking line search.

Table 2: Hyperparameters of WSNet.

Hyperparameter	Value
Number of hidden layers	2
Hidden layer width	100
Hidden block	ReLU, batch normalization, and dropout (0.2)
Gumbel–Softmax	Temperature 0.4 with hard sampling
Dual-correction projection margin	0.99
Optimization	Adam with a learning rate of 10^{-4}
Training schedule	At most 1000 epochs with patience 10
Training batch size	128 (QP/QCQP), 32 (SOCP)
Random seed	17
Recovery method	L-BFGS
Recovery budget	50 steps with a memory size of 30
Backtracking line search	Initial step 1, 10 trials, contraction 0.5
Inequality-penalty margin	10^{-6}

C.2 Neural Network Details

Let $\widehat{\Delta z}$ denote the unconstrained dual-slack correction produced by the neural network in (7). Because this raw correction may move the adjusted dual-slack variable to the boundary, we rescale it as follows:

$$\begin{aligned} z_{\min} &:= \min_{1\le j\le n} (\tilde{z}_0)_j, \\ \alpha_z &:= \min\left\{1, \frac{\rho_z z_{\min}}{\|\widehat{\Delta z}\|_\infty + \varepsilon}\right\}, \\ \Delta z_\theta(p) &:= \alpha_z \widehat{\Delta z}. \end{aligned} \tag{21}$$

Here, $z_{\min}$ is the smallest component of the selected dual-slack vector $\tilde{z}_0$, $\rho_z \in (0,1)$ controls the maximum correction magnitude, and $\varepsilon > 0$ is a small numerical constant that prevents division by zero. The scaling ensures $\|\Delta z_\theta(p)\|_\infty \le \rho_z z_{\min}$. Consequently, for every j, the adjusted dual-slack variable satisfies

$$\tilde{z}_j = \tilde{z}_{0,j} + (\Delta z_\theta(p))_j \ge \tilde{z}_{0,j} - |(\Delta z_\theta(p))_j| \ge (1-\rho_z) z_{\min} > 0, \tag{22}$$

provided that the selected reference iterate satisfies $\tilde{z}_0 > 0$. Thus, the rescaling preserves strict positivity of the adjusted dual-slack variable.

C.3 Gumbel–Softmax Details

Let e_i denote the score associated with the ith trajectory point. The Gumbel–Softmax weights used in (5) are

$$\omega_i = \frac{\exp\left((e_i + g_i)/\tau\right)}{\sum_{j=1}^{K} \exp\left((e_j + g_j)/\tau\right)}, \qquad g_i = -\log\left(-\log(u_i)\right), \quad u_i \sim \mathrm{Uniform}(0,1), \tag{23}$$

where $\tau > 0$ is the temperature and the g_i are independent Gumbel variables. Lower temperatures make the distribution closer to one-hot, whereas higher temperatures produce smoother weights.

D Proofs of the Main Theoretical Results

D.1 Proof of Theorem 5.1

Proof. Fix an arbitrary training instance $b \in \mathcal{B}$ and define its adjusted complementarity product by

$$w(\theta) := \mathrm{ReLU}(\tilde{x}_0 + \Delta x_\theta(p)) \odot \mathrm{ReLU}(\tilde{z}_0 + \Delta z_\theta(p)). \tag{24}$$

For the loss in (10), set $l(\theta) = w(\theta) - \tilde{\mu}_0 e$; for the loss in (11), set $l(\theta) = [\tilde{\mu}_0 e - w(\theta)]_+$. Thus, throughout the proof, $l(\theta)$ denotes the sample-wise residual associated with the selected loss, and $\boldsymbol{l}(\theta)$ stacks these residuals over $\mathcal{B}$.

Let $M := |\mathcal{B}|$. By (12),

$$\mathcal{L}(\theta) = \frac{1}{2M}\|\boldsymbol{l}(\theta)\|_2^2, \qquad \nabla\mathcal{L}(\theta) = \frac{1}{M} J_l(\theta)^\top \boldsymbol{l}(\theta). \tag{25}$$

We first derive the PL inequality used in the SGD analysis. Define

$$\mu_{\mathrm{WS}} := \min\left\{\frac{\lambda_J}{M}, L_{\mathrm{loss}}\right\}, \tag{26}$$

Assumption 5.1(A2) therefore implies

$$\begin{aligned}\frac{1}{2}\|\nabla\mathcal{L}(\theta)\|_2^2 &= \frac{1}{2M^2}\boldsymbol{l}(\theta)^\top J_l(\theta) J_l(\theta)^\top \boldsymbol{l}(\theta) \\ &\geq \frac{\lambda_J}{2M^2}\|\boldsymbol{l}(\theta)\|_2^2 = \frac{\lambda_J}{M}\mathcal{L}(\theta).\end{aligned} \tag{27}$$

Thus, either empirical loss satisfies the PL inequality with constant μ_{WS} in (26) (Karimi et al., 2016).

We now specify the SGD parameters. Choose

$$0 < \alpha \leq \min\left\{\frac{1}{L_{\mathrm{loss}}}, \frac{\mu_{\mathrm{WS}}\delta\tau}{L_{\mathrm{loss}}\sigma^2}\right\}, \tag{28}$$

where the second bound is omitted when $\sigma = 0$. If $2\mathcal{L}(\theta_0) > \delta\tau$, choose

$$T \geq \frac{\log(2\mathcal{L}(\theta_0)/(\delta\tau))}{\log(1/(1-\alpha\mu_{\mathrm{WS}}))}. \tag{29}$$

When $1 - \alpha\mu_{\mathrm{WS}} = 0$, take $T \geq 1$. The definition of μ_{WS} and (28) ensure that $0 \leq 1 - \alpha\mu_{\mathrm{WS}} < 1$.

By Assumption 5.1(A1), the empirical loss has an L_{loss}-Lipschitz gradient on the parameter region visited during training. Using this smoothness property and $\theta_{t+1} = \theta_t - \alpha g_t$,

$$\begin{aligned}\mathbb{E}_t[\mathcal{L}(\theta_{t+1})] &\leq \mathcal{L}(\theta_t) - \alpha\|\nabla\mathcal{L}(\theta_t)\|_2^2 + \frac{L_{\mathrm{loss}}\alpha^2}{2}\mathbb{E}_t\|g_t\|_2^2 \\ &\leq \mathcal{L}(\theta_t) - \frac{\alpha}{2}\|\nabla\mathcal{L}(\theta_t)\|_2^2 + \frac{L_{\mathrm{loss}}\alpha^2\sigma^2}{2} \\ &\leq (1 - \alpha\mu_{\mathrm{WS}})\mathcal{L}(\theta_t) + \frac{L_{\mathrm{loss}}\alpha^2\sigma^2}{2}.\end{aligned} \tag{30}$$

The second line uses $\alpha \leq 1/L_{\mathrm{loss}}$ and $\mathbb{E}_t\|g_t\|_2^2 \leq \|\nabla\mathcal{L}(\theta_t)\|_2^2 + \sigma^2$; the third uses (27). This is the standard smooth-SGD recursion (Bottou et al., 2018).

Set $r := 1 - \alpha\mu_{\mathrm{WS}} \in [0, 1)$. Taking total expectations and unrolling (30) gives

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\theta_T)] &\leq r^T\mathcal{L}(\theta_0) + \frac{L_{\mathrm{loss}}\alpha^2\sigma^2}{2}\sum_{s=0}^{T-1} r^s \\ &\leq r^T\mathcal{L}(\theta_0) + \frac{L_{\mathrm{loss}}\alpha\sigma^2}{2\mu_{\mathrm{WS}}}.\end{aligned} \tag{31}$$

The step-size restriction (28) bounds the second term by $\delta\tau/2$. If $r \in (0, 1)$, (29) implies

$$T\log(1/r) \geq \log\left(\frac{2\mathcal{L}(\theta_0)}{\delta\tau}\right), \tag{32}$$

and hence

$$r^T\mathcal{L}(\theta_0) = \exp\big(-T\log(1/r)\big)\mathcal{L}(\theta_0) \le \frac{\delta\tau}{2}. \tag{33}$$

If $r = 0$, this bound holds for every $T \ge 1$. If $2\mathcal{L}(\theta_0) \le \delta\tau$, it already holds at $T = 0$. Consequently,

$$\mathbb{E}[\mathcal{L}(\theta_T)] \le \delta\tau. \tag{34}$$

Since the loss is nonnegative, Markov's inequality yields

$$\Pr\{\mathcal{L}(\theta_T) \ge \tau\} \le \frac{\mathbb{E}[\mathcal{L}(\theta_T)]}{\tau} \le \delta. \tag{35}$$

$$\Pr\{\mathcal{L}(\theta_T) < \tau\} = 1 - \Pr\{\mathcal{L}(\theta_T) \ge \tau\} \ge 1 - \delta. \tag{36}$$

On the event $\mathcal{E}_T := \{\mathcal{L}(\theta_T) < \tau\}$, fix any $b \in \mathcal{B}$. Then

$$\begin{aligned}\|l(\theta_T)\|_2^2 \le \|\boldsymbol{l}(\theta_T)\|_2^2 &= 2M\mathcal{L}(\theta_T)\\ &\le 2M\tau = \eta^2\tilde{\mu}_{\min}^2 \le \eta^2\tilde{\mu}_0^2.\end{aligned} \tag{37}$$

Because the same event controls the stacked residual, the bound holds simultaneously for all training instances and requires no union bound. We next translate it into positivity and pairwise centrality. Under either loss,

$$w(\theta_T)_j \ge (1-\eta)\tilde{\mu}_0 > 0, \qquad \forall j. \tag{38}$$

For the loss in (10), this follows from

$$|w(\theta_T)_j - \tilde{\mu}_0| = |l(\theta_T)_j| \le \|l(\theta_T)\|_2 \le \eta\tilde{\mu}_0,$$

whereas for the loss in (11), it follows from

$$[\tilde{\mu}_0 - w(\theta_T)_j]_+ \le \|l(\theta_T)\|_2 \le \eta\tilde{\mu}_0.$$

Because $0 < \eta < 1$ and $\tilde{\mu}_0 > 0$, the right-hand side of (38) is strictly positive. By the definition of $w(\theta_T)$, this means

$$w(\theta_T)_j > 0.$$

A product of two nonnegative numbers can be strictly positive only if both factors are strictly positive. Since $\mathrm{ReLU}(u) > 0$ if and only if $u > 0$, we conclude that

$$\big(\tilde{x}_0 + \Delta x_{\theta_T}(p)\big)_j > 0, \qquad \big(\tilde{z}_0 + \Delta z_{\theta_T}(p)\big)_j > 0, \qquad \forall j.$$

Consequently, ReLU acts as the identity at the trained output, and hence

$$w(\theta_T)_j = \big(\tilde{x}_0 + \Delta x_{\theta_T}(p)\big)_j\big(\tilde{z}_0 + \Delta z_{\theta_T}(p)\big)_j.$$

Let $\tilde{\mu} := e^\top w(\theta_T)/n$ denote the average complementarity of the trained output. It remains to express the resulting centrality bounds for the two losses.

For the 2-norm neighborhood loss (10), we have $l(\theta_T) = \big(\tilde{x}_0 + \Delta x_{\theta_T}(p)\big) \odot \big(\tilde{z}_0 + \Delta z_{\theta_T}(p)\big) - \tilde{\mu}_0 e$. Cauchy–Schwarz gives

$$|\tilde{\mu} - \tilde{\mu}_0| = \left|\frac{e^\top l(\theta_T)}{n}\right| \le \frac{\|e\|_2\|l(\theta_T)\|_2}{n} \le \frac{\eta}{\sqrt{n}}\tilde{\mu}_0. \tag{39}$$

With $P := I - ee^\top/n$, which is an orthogonal projector,

$$\begin{aligned}\big\|\big(\tilde{x}_0 + \Delta x_{\theta_T}(p)\big) \odot \big(\tilde{z}_0 + \Delta z_{\theta_T}(p)\big) - \tilde{\mu}e\big\|_2 &= \|Pl(\theta_T)\|_2 \le \|l(\theta_T)\|_2\\ &\le \eta\tilde{\mu}_0 \le \frac{\eta}{1-\eta/\sqrt{n}}\tilde{\mu}.\end{aligned} \tag{40}$$

For the ℓ_∞-type neighborhood loss (11), the common bound (38) gives the one-sided pairwise centrality condition. If, in addition, $\tilde{\mu} \le \kappa\tilde{\mu}_0$, then

$$\big(\tilde{x}_0 + \Delta x_{\theta_T}(p)\big)_j\big(\tilde{z}_0 + \Delta z_{\theta_T}(p)\big)_j \ge (1-\eta)\tilde{\mu}_0 \ge \frac{1-\eta}{\kappa}\tilde{\mu}. \tag{41}$$

Together with (36), these conclusions hold simultaneously for all $b \in \mathcal{B}$ with probability at least $1 - \delta$. ■

D.2 Proof of Corollary 5.2

Proof. For a fixed training instance, let $v^s := (\Delta x_\theta^s(p), \Delta\lambda_\theta^s(p))$ denote the iterate after the sth recovery step, where the subscript T in the trained parameter θ_T is suppressed for readability. Choose gradient descent with step size $1/L_\phi$ as the recovery procedure. Assumption 5.1(A4) gives $\inf_v \phi(v) = 0$. The standard PL convergence bound therefore yields

$$\phi(v^s) \le \left(1 - \frac{\mu_\phi}{L_\phi}\right)^s \phi(v^0) =: E_s \tag{42}$$

(Karimi et al., 2016). Because $0 \le 1 - \mu_\phi/L_\phi < 1$, we have $E_s \to 0$ as $s \to \infty$. Moreover, by the definition of ϕ,

$$\|\mathcal{R}(\tilde{x}_0 + \Delta x_\theta^s(p), \tilde{\lambda}_0 + \Delta\lambda_\theta^s(p), \tilde{z}_0 + \Delta z_\theta(p); p)\|_2 = \sqrt{\phi(v^s)} \le \sqrt{E_s}. \tag{43}$$

We condition on the event in (36), which holds with probability at least $1 - \delta$, and combine the recovery bound with the two training-loss guarantees.

For the 2-norm loss (10), Theorem 5.1 gives strict positivity and

$$\|l\|_2 \le \eta\tilde{\mu}_0, \qquad l := \big(\tilde{x}_0 + \Delta x_\theta^s(p)\big) \odot \big(\tilde{z}_0 + \Delta z_\theta(p)\big) - \tilde{\mu}_0 e. \tag{44}$$

The average complementarity of the recovered point is

$$\tilde{\mu} = \frac{1}{n} e^\top \left[\big(\tilde{x}_0 + \Delta x_\theta^s(p)\big) \odot \big(\tilde{z}_0 + \Delta z_\theta(p)\big)\right]. \tag{45}$$

Hence, by Cauchy–Schwarz,

$$|\tilde{\mu} - \tilde{\mu}_0| = \frac{|e^\top l|}{n} \le \frac{\|e\|_2\|l\|_2}{n} \le \frac{\eta}{\sqrt{n}}\tilde{\mu}_0, \tag{46}$$

and therefore

$$\tilde{\mu} \ge \left(1 - \frac{\eta}{\sqrt{n}}\right)\tilde{\mu}_0. \tag{47}$$

Since $E_s \to 0$, a finite s can be chosen such that

$$E_s \le \left[\beta c^0 \left(1 - \frac{\eta}{\sqrt{n}}\right)\tilde{\mu}_0\right]^2. \tag{48}$$

Equations (43), (47), and (48) then imply

$$\|\mathcal{R}(\tilde{x}_0 + \Delta x_\theta^s(p), \tilde{\lambda}_0 + \Delta\lambda_\theta^s(p), \tilde{z}_0 + \Delta z_\theta(p); p)\|_2 \le \beta c^0 \tilde{\mu}. \tag{49}$$

With $P := I - ee^\top/n$, the centrality error satisfies

$$\begin{aligned}\left\|\big(\tilde{x}_0 + \Delta x_\theta^s(p)\big) \odot \big(\tilde{z}_0 + \Delta z_\theta(p)\big) - \tilde{\mu}e\right\|_2 &= \|Pl\|_2 \le \|l\|_2 \\ &\le \frac{\eta}{1 - \eta/\sqrt{n}}\tilde{\mu} \le \alpha_2\tilde{\mu},\end{aligned} \tag{50}$$

where the last inequality follows from $\eta \le \alpha_2/(1+\alpha_2/\sqrt{n})$. Thus the output belongs to $\mathcal{N}_2(\alpha_2, \beta)$.

For the one-sided loss (11), Theorem 5.1 gives, for every j,

$$\left[\tilde{\mu}_0 - \big(\tilde{x}_0 + \Delta x_\theta^s(p)\big)_j \big(\tilde{z}_0 + \Delta z_\theta(p)\big)_j\right]_+ \le \eta\tilde{\mu}_0. \tag{51}$$

Consequently,

$$\big(\tilde{x}_0 + \Delta x_\theta^s(p)\big)_j \big(\tilde{z}_0 + \Delta z_\theta(p)\big)_j \ge (1 - \eta)\tilde{\mu}_0, \qquad j = 1, \ldots, n. \tag{52}$$

Summing these inequalities and dividing by n gives

$$\begin{aligned}\tilde{\mu} &= \frac{1}{n}\sum_{j=1}^n \big(\tilde{x}_0 + \Delta x_\theta^s(p)\big)_j \big(\tilde{z}_0 + \Delta z_\theta(p)\big)_j \\ &\ge (1 - \eta)\tilde{\mu}_0.\end{aligned} \tag{53}$$

Because $E_s \to 0$, a finite s can also be chosen such that

$$E_s \le \left[\beta c^0 (1-\eta)\tilde{\mu}_0\right]^2. \tag{54}$$

Together with (43) and (53), this gives the required residual bound $\|\mathcal{R}\|_2 \le \beta c^0 \tilde{\mu}$. Finally, if $\tilde{\mu} \le \kappa \tilde{\mu}_0$, then

$$\left(\tilde{x}_0 + \Delta x_\theta^s(p)\right)_j \left(\tilde{z}_0 + \Delta z_\theta(p)\right)_j \ge \frac{1-\eta}{\kappa}\tilde{\mu}, \qquad j = 1, \dots, n. \tag{55}$$

The upper-complementarity condition is needed because the one-sided loss does not penalize products above $\tilde{\mu}_0$ and therefore does not control $\tilde{\mu}$ from above. The output consequently belongs to $\mathcal{N}_{-\infty}((1-\eta)/\kappa, \beta)$. ■

E Supporting Results

E.1 Context Transfer

Assumption E.1 (Contextual residual regularity)**.** On a set containing the stored trajectory and WSNet outputs, there is an $L_R \ge 0$ such that

$$\|\mathcal{R}(x, \lambda, z; p) - \mathcal{R}(x, \lambda, z; p_0)\|_2 \le L_R \|p - p_0\|_2. \tag{56}$$

Assumption E.2 (Reference trajectory quality)**.** Each stored reference iterate satisfies, for some $\alpha_i \ge 0$ and $\gamma_i \in (0, 1)$,

$$\|\mathcal{R}(x_0^i, \lambda_0^i, z_0^i; p_0)\|_2 \le \alpha_i \mu_0^i, \tag{57}$$

$$x_0^i > 0, \quad z_0^i > 0, \qquad x_{0,j}^i z_{0,j}^i \ge \gamma_i \mu_0^i \quad \forall j. \tag{58}$$

Proposition E.1 (Residual transfer bound)**.** Under Assumptions E.1 and E.2, candidate i satisfies

$$\|\mathcal{R}(x_0^i, \lambda_0^i, z_0^i; p)\|_2 \le \alpha_i \mu_0^i + L_R \|p - p_0\|_2. \tag{59}$$

Its perturbed residual coefficient, normalized by μ_0^i, is therefore at most $\alpha_i + L_R\|p - p_0\|_2/\mu_0^i$.

Proof. Add and subtract the residual at p_0 and apply the triangle inequality:

$$\begin{aligned} &\|\mathcal{R}(x_0^i, \lambda_0^i, z_0^i; p)\|_2 \\ &\quad \le \|\mathcal{R}(x_0^i, \lambda_0^i, z_0^i; p_0)\|_2 + \|\mathcal{R}(x_0^i, \lambda_0^i, z_0^i; p) - \mathcal{R}(x_0^i, \lambda_0^i, z_0^i; p_0)\|_2 \\ &\quad \le \alpha_i \mu_0^i + L_R \|p - p_0\|_2. \end{aligned} \tag{60}$$

Dividing by $\mu_0^i > 0$ proves the normalized statement. ■

Corollary E.2 (Candidate-specific context radius)**.** Fix $\bar{\alpha} > \alpha_i$. Candidate i satisfies $\|\mathcal{R}\|_2 \le \bar{\alpha}\mu_0^i$ for the perturbed problem whenever

$$\|p - p_0\|_2 \le \frac{(\bar{\alpha} - \alpha_i)\mu_0^i}{L_R}. \tag{61}$$

Proof. Substitute the stated context radius into Proposition E.1. ■

The perturbation term in (59) is divided by μ_0^i after normalization. Late trajectory points can therefore be more sensitive to contextual changes, whereas earlier points are generally more central but less advanced. This trade-off motivates trajectory-point selection.

E.2 Gumbel–Softmax Selection

Proposition E.3 (Concentration of the relaxed selector)**.** Condition on the perturbed scores $(e_1 + g_1, \dots, e_K + g_K)$ and suppose q is their unique maximizer. If

$$\Delta_e := (e_q + g_q) - \max_{j \ne q}(e_j + g_j) > 0, \tag{62}$$

then the selector in Section 4.1 satisfies

$$1 - \omega_q \le (K-1)\exp(-\Delta_e/\tau). \tag{63}$$

Proof. For every $j \neq q$, $(e_j + g_j) - (e_q + g_q) \leq -\Delta_e$. Hence

$$1 - \omega_q = \frac{\sum_{j \neq q} \exp(((e_j + g_j) - (e_q + g_q))/\tau)}{1 + \sum_{j \neq q} \exp(((e_j + g_j) - (e_q + g_q))/\tau)} \leq \sum_{j \neq q} \exp(((e_j + g_j) - (e_q + g_q))/\tau) \leq (K-1)\exp(-\Delta_e/\tau). \tag{64}$$

■

This result connects the differentiable training relaxation (Jang et al., 2017; Maddison et al., 2017) to the hard trajectory decision used at inference.

E.3 Conditional Path-Following Budget

Assumption E.3 (Barrier contraction)**.** Once initialized in an admissible neighborhood, the path-following method remains in that neighborhood and satisfies

$$\mu_{k+1} \leq \rho \mu_k \qquad \text{for a fixed } \rho \in (0, 1). \tag{65}$$

The method terminates when $\mu_k \leq \varepsilon_\mu$.

Proposition E.4 (Conditional path-following step budget)**.** Under Assumption E.3, an admissible initialization with barrier parameter μ_{init} reaches $\mu_k \leq \varepsilon_\mu$ within

$$N(\mu_{\text{init}}) = \max\left\{0, \left\lceil \frac{\log(\mu_{\text{init}}/\varepsilon_\mu)}{\log(1/\rho)} \right\rceil\right\} \tag{66}$$

path-following steps.

Proof. Repeated contraction gives $\mu_k \leq \rho^k \mu_{\text{init}}$. Solving $\rho^k \mu_{\text{init}} \leq \varepsilon_\mu$ for k and taking the non-negative ceiling proves (66). ■