MANIFEST: MANIFOLD-INFORMED LEARNING OF FEASIBILITY-ENFORCING NEURAL SOLVERS FOR CONSTRAINED OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient solution of parametric and nonlinear optimization problems is important in several engineering applications. Neural network surrogates have emerged as a promising alternative to traditional optimization solvers in real-time applications where fast solutions are necessary. However, ensuring constraint satisfaction, especially with equality constraints, remains a key challenge in neural solvers. We address this by introducing `MANIFEST`, a Manifold-Informed Feasibility Enforcement Strategy, that directly incorporates the geometry of the equality-constraint manifold into neural surrogate training, thus guiding the learned neural map toward the feasible manifold before any correction is applied. Specifically, we design a training objective that integrates optimality, distance of the predictions from the feasible manifold, and first-order information about the local geometry and parametric variation of the feasible manifold, followed by a differentiable restoration step. In contrast to existing approaches that rely on post-prediction projection or restoration for constraint enforcement, `MANIFEST` trains the learned map itself to remain close to the feasible manifold, thereby avoiding substantial post-prediction corrections and associated numerical issues from projections in challenging optimization landscapes. We evaluate our framework on quadratically constrained quadratic programs (QCQPs), nonlinear programs (NLPs), and nonconvex nonlinear AC optimal power flow (AC-OPF). Across these problems, `MANIFEST` consistently outperforms state-of-the-art approaches, achieving near-optimal solution with negligible equality and inequality violations even without correction, with subsequent restoration further reducing violation residuals to numerical zero.

1 INTRODUCTION

Neural solvers have emerged as a promising approach for learning approximate solutions to parametric constrained optimization problems (Chen et al., 2022a; Amos, 2023), with applications in optimal power flow (Zamzam & Baker, 2020; Fioretto et al., 2020), model predictive control (Drgoňa et al., 2024; Le et al., 2026), and wireless resource allocation (Sun et al., 2018). By exploiting shared problem structure across parameter values, these solvers can reduce the computational cost of repeated online optimization. Within this class, unsupervised learning approaches train neural solvers directly from the objective and constraint functions, avoiding the need for datasets of solver-generated solutions (Donti et al., 2021; Park & Van Hentenryck, 2023; Drgoňa et al., 2024; Li et al., 2025). In this paper, we consider this unsupervised setting, where the goal is to train neural solvers to learn solutions to constrained optimization problems without using optimal solution data.

A key challenge in training neural solvers for constrained optimization is enforcing feasibility, especially with equality constraints, as updates to the network parameters that improve the objective can increase constraint violations. One approach to enforce feasibility is to incorporate penalties or Lagrangian regularization for constraint violations into the training objective (Fioretto et al., 2020; Park & Van Hentenryck, 2023; Drgoňa et al., 2024). While these penalties discourage violations, they do not generally ensure constraint satisfaction. Another approach is to embed differentiable optimization layers for quadratic programs (Amos & Kolter, 2017) or more general convex programs (Agrawal et al., 2019). Recent methods also incorporate explicit constraint enforcement through

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.

equality completion (Donti et al., 2021), projection (Liang et al., 2024; Roy et al., 2026), or iterative feasibility-seeking procedures (Nguyen & Donti, 2025).

A central limitation of existing feasibility-enforcement approaches is that the feasibility of the final output can be dominated by the post-prediction correction or completion procedure rather than by the learned neural solver itself. In principle, even a poor raw prediction can be mapped to a feasible point by a sufficiently powerful projection, completion, or iterative restoration procedure. For nonlinear constraints, such procedures may additionally require repeated Jacobian evaluations and linear solves at inference time, increasing computational cost and numerical sensitivity. Consequently, evaluating only the corrected output does not reveal whether the learned solver has captured the feasible structure of the underlying optimization problem. Motivated by this, we ask the following question: ***can we train neural solvers to produce near-optimal, near-feasible raw predictions without relying on substantial post-prediction correction or completion?***

In this paper, we introduce `MANIFEST`, a Manifold-Informed Feasibility Enforcement Strategy for unsupervised learning of neural solvers. Under standard regularity conditions, the equality constraints define a smooth manifold in a neighborhood of each feasible decision vector (Boumal, 2023). We use the geometry of this equality manifold to train the network toward feasibility while minimizing the optimization objective. Specifically, we construct a geometric feasibility error from a local Gauss-Newton approximation of the squared distance between the raw prediction and the manifold. We also include first-order information about the geometry of the equality manifold to encourage predictions to satisfy the equality constraints under variations in the problem parameters. We incorporate these geometric terms, along with a term capturing the inequality residuals, into a primal-dual learning framework that combines objective minimization with feasibility enforcement. Across experiments spanning quadratically constrained quadratic programs (QCQPs), nonlinear programs (NLPs), and nonconvex nonlinear AC optimal power flow (AC-OPF), `MANIFEST` achieves near-zero equality and inequality violations in raw predictions while maintaining near-optimal solution quality, with a small post-training restoration reducing any remaining violations to numerical zero. Our main contributions are as follows:

- **Learning feasibility through manifold geometry.** We develop a theoretically grounded feasibility error that incorporates geometric information from the equality-feasible manifold, including both proximity to the manifold and its first-order variation with the problem parameters. This goes beyond constraint-residual and correction-displacement penalties by providing a geometry-aware training signal that directly promotes near-feasible raw predictions.
- **Primal-dual unsupervised learning for constrained optimization.** We construct a primal-dual framework that incorporates these geometric terms to simultaneously promote optimality and constraint satisfaction through updates to the network parameters and constraint multipliers. This couples training for solution quality with direct penalties on departures from the equality manifold and inequality violations, without requiring solver-generated solution labels.
- **Feasibility and optimality with minimal correction.** `MANIFEST` significantly outperforms state-of-the-art neural solvers in raw-prediction feasibility with negligible post-prediction correction across QCQPs, NLPs, and nonconvex nonlinear AC-OPF, while maintaining near-optimal solution quality.

2 Related Work

Learning for constrained parametric optimization. Learning-based optimization methods seek to approximate solutions to families of parametric optimization problems, thereby replacing or accelerating repeated calls to a numerical solver (Kotary et al., 2021; Amos, 2023; Chen et al., 2022a). There are two main approaches to training neural solvers, namely supervised and unsupervised learning. Supervised approaches learn from datasets of solutions spanning multiple problem instances, typically generated using classical optimization solvers (Hertneck et al., 2018; Zamzam & Baker, 2020; Fioretto et al., 2020; Chen et al., 2022b; Misra et al., 2022). However, the generation of informative datasets is challenging, especially in nonconvex problems, as changes in operating conditions and active constraints can yield different local optima, and limited coverage can lead to large objective errors and constraint violations in operating conditions that differ from the training distribution (Chen et al., 2022a). To address the dependence on solution labels, unsupervised and self-supervised approaches have been proposed to train directly from the objective and constraint functions of the underlying optimization problem (Donti et al., 2021; Park & Van Hentenryck, 2023;

Drgoňa et al., 2024; Tanneau & Van Hentenryck, 2024; Li et al., 2025). Within this setting, penalty, Lagrangian, and primal-dual formulations provide different mechanisms to objective minimization and constraint satisfaction (Fioretto et al., 2021; Park & Van Hentenryck, 2023). In this work, we adopt the unsupervised setting and design an approach that uses the geometry of the equality-feasible manifold to enforce feasibility.

Constraint enforcement in learning-based optimization. A central challenge in learning-based optimization is ensuring that neural predictions satisfy the constraints of the underlying problem. Existing approaches enforce feasibility through soft constraint penalties or Lagrangian regularization (Fioretto et al., 2020; 2021; Park & Van Hentenryck, 2023), equality completion or structure-preserving parameterizations (Donti et al., 2021; Pan et al., 2021), differentiable projection and hard-constraint layers (Amos & Kolter, 2017; Agrawal et al., 2019; Huang et al., 2021; Liang et al., 2024; Tordesillas et al., 2023; Lastrucci & Schweidtmann, 2025; Min et al., 2024), or iterative feasibility correction (Donti et al., 2021). While these approaches can substantially improve or enforce feasibility at the final output, methods that rely on post-prediction projection or correction may require substantial modifications to an infeasible neural prediction, particularly for nonlinear or poorly conditioned constraint landscapes, where repeated corrections can also introduce additional computational cost and numerical sensitivity. Recent methods therefore also penalize the displacement between the neural prediction and its corrected solution (Nguyen & Donti, 2025; Roy et al., 2026). However, a small correction magnitude does not necessarily imply that the neural predictor itself has learned feasible solutions, and feasibility may still depend on the subsequent correction procedure. In contrast, MANIFEST explicitly trains the neural prediction using the geometry of the equality-feasible manifold, with the goal of directly producing near-feasible solutions with minimal post-prediction restoration.

3 MANIFEST: Manifold-Informed Feasibility Enforcement Strategy for Constrained Optimization

We consider a family of constrained optimization problems indexed by parameters $x \in \mathcal{X} \subseteq \mathbb{R}^p$,

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & c(x, y) \\ \text{subject to} \quad & g(x, y) = 0, \qquad h(x, y) \leq 0. \end{aligned} \tag{1}$$

where $y \in \mathbb{R}^n$ denotes the decision variables, $c : \mathcal{X} \times \mathbb{R}^n \to \mathbb{R}$ is the objective function, and $g : \mathcal{X} \times \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathcal{X} \times \mathbb{R}^n \to \mathbb{R}^q$ collect the equality and inequality constraints. For each x, we define the feasible set

$$\mathcal{F}_x := \{y \in \mathbb{R}^n \,|\, g(x, y) = 0,\ h(x, y) \leq 0\}, \tag{2}$$

and the equality-feasible set

$$\mathcal{E}_x := \{y \in \mathbb{R}^n \,|\, g(x, y) = 0\}. \tag{3}$$

Let $x \sim \mathcal{D}$ denote the distribution of problem parameters. Given samples from $\mathcal{D}$, we seek to train a neural solver $F_\theta : \mathcal{X} \to \mathbb{R}^n$ that produces low-cost feasible decisions without using optimizer solutions as training labels. Specifically, for a problem instance x, the neural solver output is denoted by $y_\theta(x) := F_\theta(x)$. We formulate this unsupervised learning objective as

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{x \sim \mathcal{D}} \left[c(x, F_\theta(x)) \right] \\ \text{subject to} \quad & g(x, F_\theta(x)) = 0, \qquad h(x, F_\theta(x)) \leq 0. \end{aligned} \tag{4}$$

In other words, our goal is to train the neural solver using samples of the problem parameters and evaluations of the objective and constraints, without requiring data on the optimal decisions.

Figure 1 summarizes the MANIFEST framework to address this problem. Given a problem instance x, we use the primal network to compute the raw prediction $y_\theta(x)$. We model the equality-feasible set locally as a smooth manifold and train $y_\theta(x)$ toward this manifold using a geometric distance error and first-order geometry term that captures its variation with the problem parameters. In parallel, we use a dual network to estimate multipliers for the equality and inequality constraints. We train the primal and dual networks through an augmented primal-dual objective that combines the

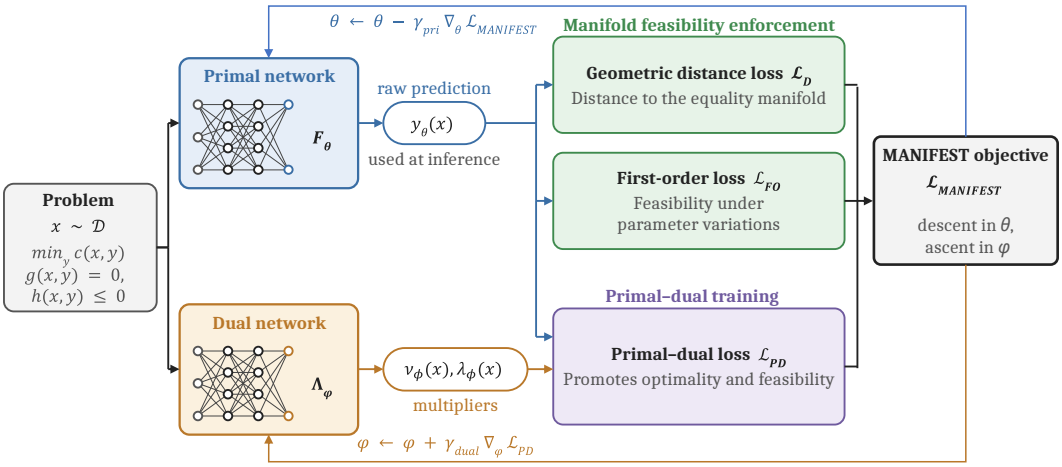

Figure 1: Overview of the MANIFEST framework.

original cost with the remaining equality residuals and inequality violation, alongside a differentiable Gauss-Newton procedure to measure and reduce the remaining equality-feasibility error. By incorporating information about the constraint geometry, we explicitly train the network to produce nearly feasible predictions before correction. We describe each component of the framework in the following subsections.

3.1 Manifold Feasibility Enforcement

We begin by enforcing the equality constraints through the local geometry of the equality-feasible set. We first state some local smoothness and regularity conditions on the equality-feasible set $\mathcal{E}_x$ as defined in (3).

Assumption 1 (Local regularity of the equality constraints)**.** *The mapping g admits a twice continuously differentiable extension to an open neighborhood of the equality-feasible pairs $\{(x,y) : x \in \mathcal{X},\ y \in \mathcal{E}_x\}$. For every $x \in \mathcal{X}$ and $y \in \mathcal{E}_x$, there exists a neighborhood $\mathcal{U}_y \subset \mathbb{R}^n$ of y such that $\operatorname{rank} J_y(x,\widetilde{y}) = \operatorname{rank} J_y(x,y)$ for all $\widetilde{y} \in \mathcal{U}_y$.*

Assumption 1 is mild, allows locally redundant equality constraints, and holds for most smooth nonlinear equalities encountered in practice. Under Assumption 1, the equality-feasible set has a smooth local manifold structure (Boumal, 2023). We now characterize the corresponding tangent and normal spaces which form the basis for the geometric information used in our neural network training. The proofs of all results are provided in Appendix A.

Proposition 1 (Local equality-manifold geometry)**.** *Under Assumption 1 with $y \in \mathcal{E}_x$ and $r = \operatorname{rank} J_y(x,y)$, there exists an open neighborhood $\mathcal{V}_y \subseteq \mathcal{U}_y$ of y such that $\mathcal{E}_x \cap \mathcal{V}_y$ is a smooth embedded submanifold of $\mathbb{R}^n$ of dimension $n-r$. The tangent and normal spaces at y satisfy*

$$T_y\mathcal{E}_x = \ker J_y(x,y), \qquad N_y\mathcal{E}_x = \operatorname{range}\big(J_y(x,y)^\top\big)\,. \tag{5}$$

At a feasible decision, $\ker J_y(x,y)$ contains the decision-space directions that preserve the equality constraints to first order, while $\operatorname{range}(J_y(x,y)^\top)$ identifies the corresponding normal directions. For an infeasible point $\bar{y} \notin \mathcal{E}_x$, the tangent and normal spaces of $\mathcal{E}_x$ are not defined at $\bar{y}$. However, the constraint Jacobian still identifies the directions that leave the equality residual unchanged to first order and their orthogonal complement. We denote these subspaces by

$$\mathcal{T}_x^{\mathrm{lin}}(\bar{y}) := \ker J_y(x,\bar{y}), \qquad \mathcal{N}_x^{\mathrm{lin}}(\bar{y}) := \operatorname{range}\big(J_y(x,\bar{y})^\top\big)\,. \tag{6}$$

We use these constructions to characterize equality feasibility and quantify violations of the equality constraints under perturbations in the problem parameters.

Geometric distance. We measure the distance of the raw prediction $y_\theta(x)$ from the equality manifold using the local linearization of the equality constraints. Define $r_\theta(x) := g(x, y_\theta(x))$ and $J_\theta(x) := J_y(x, y_\theta(x))$. For a perturbation $\Delta y \in \mathbb{R}^n$, we can write $g(x, y_\theta(x) + \Delta y) \approx$

$r_\theta(x) + J_\theta(x)\Delta y$. When $J_\theta(x)$ has full row rank, we define the minimum-norm perturbation that satisfies the linearized equality constraints by

$$\Delta y^\star_{\text{lin}}(x) := \underset{\Delta y \in \mathbb{R}^n}{\arg\min} \left\{ \|\Delta y\|_2^2 \,\middle|\, r_\theta(x) + J_\theta(x)\Delta y = 0 \right\}. \tag{7}$$

Proposition 2 (Local distance and normal direction)**.** *Suppose that $J_\theta(x)$ has full row rank. The minimum perturbation in (7) satisfies*

$$\Delta y^\star_{\text{lin}}(x) = -J_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top \right)^{-1} r_\theta(x) \in \mathcal{N}_x^{\text{lin}}(y_\theta(x)). \tag{8}$$

Moreover,

$$\|\Delta y^\star_{\text{lin}}(x)\|_2^2 = r_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top \right)^{-1} r_\theta(x). \tag{9}$$

Proposition 2 establishes that the minimum perturbation to the raw prediction $y_\theta(x)$ required to satisfy the linearized equality constraints lies in the local normal subspace. Note that (9) equals the squared Euclidean distance from $y_\theta(x)$ to $\mathcal{E}_x$ for affine equality constraints, and gives a local Gauss-Newton approximation of this distance for nonlinear equality constraints. Motivated by this result, we define the regularized geometric distance that serves as a measure of equality violations:

$$D^2_{\delta_d}(x, y_\theta(x)) := r_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top + \delta_d I \right)^{-1} r_\theta(x), \qquad \delta_d \geq 0. \tag{10}$$

We set δ_d to a small positive value during training to accommodate locally redundant constraints and stabilize the inverse when $J_\theta(x)$ is poorly conditioned. When $J_\theta(x)$ has full row rank and is sufficiently well conditioned, setting $\delta_d = 0$ recovers the squared local displacement in (9). The geometric distance therefore reduces equality infeasibility through the local normal directions without directly penalizing tangent directions used to improve the objective.

We are now ready to define a **geometric distance loss** that characterizes equality violations in terms of the distance to the equality manifold as follows:

$$\mathcal{L}_D(\theta) := \mathbb{E}_{X \sim \mathcal{D}} \left[D^2_{\delta_d}(X, y_\theta(X)) \right]. \tag{11}$$

Unlike a standard penalty on the residual $\|g(x,y)\|_2^2$, our geometric distance $D_0^2(x, y_\theta(x))$ is invariant to nonsingular linear reparameterizations of the equality constraints, making the loss less sensitive to arbitrary constraint scaling (Proposition 5; Appendix A), and captures feasibility information beyond the commonly used correction-magnitude penalty (Nguyen & Donti (2025); Roy et al. (2026)) (Proposition 6; Appendix A.2.)

First-order feasibility under parameter variations. The geometric distance loss $\mathcal{L}_D(\theta)$ measures proximity to the equality manifold at a fixed problem instance, but does not capture whether the neural solver follows the manifold as the problem parameters vary. Thus, updates to the network parameters θ that improve the optimization objective for one problem instance can increase equality violations at other problem instances. We account for this effect by incorporating first-order geometric information into training. We begin with a condition that must be satisfied by any differentiable feasible decision rule.

Proposition 3 (First-order manifold feasibility)**.** *Let $F : \mathcal{X} \to \mathbb{R}^n$ be differentiable and satisfy $g(z, F(z)) = 0$ for every z in an open neighborhood of x contained in $\mathcal{X}$. Then,*

$$D_x g(x, F(x)) + D_y g(x, F(x)) D_x F(x) = 0. \tag{12}$$

Proposition 3 characterizes how the predictions $F(x)$ must change with x to preserve equality feasibility as the problem parameters change. We incorporate this first-order geometric condition into training by penalizing directional variations of the raw equality residual. Let $\mathbb{P}_V$ denote the distribution used to sample nonzero directions in the parameter space $\mathbb{R}^p$. We can now define the following **first-order loss** term:

$$\mathcal{L}_{\text{FO}}(\theta) := \mathbb{E}_{\substack{X \sim \mathcal{D} \\ V \sim \mathbb{P}_V}} \left[\frac{\|D_x r_\theta(X)[V]\|_2^2}{\|V\|_2^2} \right]. \tag{13}$$

In implementation, we evaluate the directional derivatives using Jacobian-vector products, avoiding construction of the full residual Jacobian.

The geometric distance loss $\mathcal{L}_D(\theta)$ and the first-order loss $\mathcal{L}_{FO}(\theta)$ provide complementary training signals. The geometric distance error directs the raw predictions toward the equality manifold, while the first-order error trains the neural solver to maintain equality feasibility when the problem parameters are updated. We next combine these errors with a primal-dual objective to jointly promote optimality and feasibility enforcement.

3.2 PRIMAL-DUAL TRAINING FOR OPTIMALITY AND FEASIBILITY

We have so far focused on constructing loss functions that train the raw prediction $y_\theta(x)$ toward equality feasibility. We now introduce an unsupervised primal-dual objective that trains F_θ for both optimality and feasibility of the equality and inequality constraints. Within this formulation, we treat F_θ as the *primal network* as its output $y_\theta(x) = F_\theta(x)$ defines candidate solutions to (1). We introduce a *dual network*

$$\Lambda_\phi : \mathcal{X} \longrightarrow \mathbb{R}^m \times \mathbb{R}^q_+, \qquad \Lambda_\phi(x) := \big(\nu_\phi(x), \lambda_\phi(x)\big), \tag{14}$$

to parameterize the constraint multipliers associated with each problem instance. Here, ϕ denotes the dual-network parameters, $\nu_\phi(x) \in \mathbb{R}^m$ denotes the equality multipliers, and $\lambda_\phi(x) \in \mathbb{R}^q_+$ denotes the inequality multipliers. We leave the equality multipliers unrestricted and enforce nonnegativity of the inequality multipliers through a softplus output parameterization of the dual network (see implementation details in Appendix B).

Primal-dual objective. We construct a loss that embeds the optimization objective and both the equality and inequality constraints through an augmented Lagrangian. The multipliers predicted by the dual network adapt the weight assigned to each constraint, while quadratic penalty terms directly penalize constraint violations. For a problem parameter x and candidate decision y, we define the **augmented loss** function

$$\mathcal{L}_{\text{aug}}(x, y, \nu, \lambda) := w_{\text{obj}} c(x, y) + \nu^\top g(x, y) + \frac{\mu}{2}\|g(x, y)\|_2^2 + \lambda^\top [h(x, y)]_+ + \frac{\eta}{2}\|[h(x, y)]_+\|_2^2, \tag{15}$$

where $w_{\text{obj}} > 0$ scales the objective, ν and λ are the equality and inequality multipliers respectively, $\mu, \eta > 0$ weight the equality and inequality penalties, and $[h(x, y)]_{+,j} := \max\{h_j(x, y), 0\}$ for $j = 1, \ldots, q$, so that only violated inequality constraints contribute to the loss, with the box-type inequality constraints directly enforced by using $tanh$ activations at the last hidden layer of F_θ.

Although the geometric losses in (11) and (13) reduce equality violations in the raw prediction, they do not guarantee exact equality feasibility throughout training. Consequently, a small objective value at an infeasible raw prediction need not indicate a high-quality feasible solution to (1). We incorporate a small number of differentiable Gauss-Newton iterations for the equality constraints within the training loss and evaluate the objective and constraint penalties at the resulting iterate. We backpropagate through these iterations to update the primal parameters θ.

Proposition 4 (Regularized equality error)**.** *Fix $x \in \mathcal{X}$, $y \in \mathbb{R}^n$, and $\delta_r > 0$. The problem*

$$\min_{e \in \mathbb{R}^n} \left\{ \frac{1}{2}\, \|g(x, y) - J_y(x, y)e\|_2^2 + \frac{\delta_r}{2}\|e\|_2^2 \right\} \tag{16}$$

has the unique solution

$$e_{\delta_r}(x, y) = J_y(x, y)^\top \left(J_y(x, y)J_y(x, y)^\top + \delta_r I_m\right)^{-1} g(x, y) \in \text{range}\big(J_y(x, y)^\top\big)\,. \tag{17}$$

If $J_y(x, y)$ has full row rank, then $e_0(x, y) = J_y(x, y)^\top \left(J_y(x, y)J_y(x, y)^\top\right)^{-1} g(x, y)$ is the minimum-norm solution of $J_y(x, y)e = g(x, y)$. In other words, $y - e_0(x, y)$ satisfies the linearized equality constraints.

Starting from the raw prediction $y_\theta^{(0)}(x) := y_\theta(x)$, we apply the differentiable update

$$y_\theta^{(k+1)}(x) := y_\theta^{(k)}(x) - \alpha_r e_{\delta_r}\Big(x, y_\theta^{(k)}(x)\Big)\,, \qquad k = 0, \ldots, K-1, \tag{18}$$

where $\alpha_r \in (0, 1]$ denotes a constant step size. After K iterations, let $\widetilde{y}_\theta(x) := y_\theta^{(K)}(x)$. We use this iterate in the augmented objective and define the **primal-dual loss** as

$$\mathcal{L}_{\text{PD}}(\theta, \phi) := \mathbb{E}_{X \sim \mathcal{D}}\left[\mathcal{L}_{\text{aug}}\left(X, \widetilde{y}_\theta(X), \nu_\phi(X), \lambda_\phi(X)\right)\right]. \tag{19}$$

We combine the primal-dual loss with the geometric distance and first-order losses defined earlier to construct the complete **`MANIFEST` objective function:**

$$\mathcal{L}_{\text{MANIFEST}}(\theta, \phi) := \mathcal{L}_{\text{PD}}(\theta, \phi) + \omega_D \mathcal{L}_D(\theta) + \omega_{\text{FO}} \mathcal{L}_{\text{FO}}(\theta) \tag{20}$$

Algorithm 1 MANIFEST training procedure

Require: training samples $\{x^{(s)}\}_{s=1}^{N}$; primal network F_θ; dual network Λ_ϕ; direction distribution $\mathbb{P}_V$; number of training Gauss-Newton iterations K; parameters $\delta_d, \delta_r, \alpha_r$; weights $\omega_D, \omega_{\mathrm{FO}}$; learning rates $\gamma_{\mathrm{pri}}, \gamma_{\mathrm{dual}}$

for each batch $\mathcal{B} = \{x_i\}_{i=1}^{B}$ **do**
  for $i = 1, \dots, B$ **do**
    $y_{\theta,i} \leftarrow F_\theta(x_i),\ d_i \leftarrow D^2_{\delta_d}(x_i, y_{\theta,i})$
    sample $V_i \sim \mathbb{P}_V$ and compute $\ell_{\mathrm{FO},i} \leftarrow \|D_x r_\theta(x_i)[V_i]\|_2^2 / \|V_i\|_2^2$
    $y_i^{(0)} \leftarrow y_{\theta,i}$
    for $k = 0, \dots, K-1$ **do**
      $e_i^{(k)} \leftarrow e_{\delta_r}(x_i, y_i^{(k)})$
      $y_i^{(k+1)} \leftarrow y_i^{(k)} - \alpha_r e_i^{(k)}$
    end for
    $\widetilde{y}_{\theta,i} \leftarrow y_i^{(K)}$
    $(\nu_i, \lambda_i) \leftarrow \Lambda_\phi(x_i)$, with $\lambda_i \geq 0$
  end for
  $\widehat{\mathcal{L}}_{D,\mathcal{B}} \leftarrow B^{-1}\sum_{i=1}^{B} d_i,\ \widehat{\mathcal{L}}_{\mathrm{FO},\mathcal{B}} \leftarrow B^{-1}\sum_{i=1}^{B} \ell_{\mathrm{FO},i},\ \widehat{\mathcal{L}}_{\mathrm{PD},\mathcal{B}} \leftarrow B^{-1}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{aug}}(x_i, \widetilde{y}_{\theta,i}, \nu_i, \lambda_i)$
  $\widehat{\mathcal{L}}_{\mathrm{MANIFEST},\mathcal{B}} \leftarrow \widehat{\mathcal{L}}_{\mathrm{PD},\mathcal{B}} + \omega_D \widehat{\mathcal{L}}_{D,\mathcal{B}} + \omega_{\mathrm{FO}} \widehat{\mathcal{L}}_{\mathrm{FO},\mathcal{B}}$
  $\theta \leftarrow \theta - \gamma_{\mathrm{pri}} \nabla_\theta \widehat{\mathcal{L}}_{\mathrm{MANIFEST},\mathcal{B}}$
  $\phi \leftarrow \phi + \gamma_{\mathrm{dual}} \nabla_\phi \widehat{\mathcal{L}}_{\mathrm{PD},\mathcal{B}}$
end for
return trained primal network F_θ

where $\omega_D, \omega_{\mathrm{FO}} > 0$ weight the two geometric losses. This yields the optimization problem:

$$\min_{\theta} \max_{\phi}\ \mathcal{L}_{\mathrm{MANIFEST}}(\theta, \phi). \tag{21}$$

The geometric losses act directly on the raw prediction $y_\theta(x)$, while $\mathcal{L}_{\mathrm{PD}}$ trains for optimality and feasibility through the K differentiable Gauss-Newton iterations. During training, we approximate the expectations in $\mathcal{L}_{\mathrm{PD}}$, $\mathcal{L}_D$, and $\mathcal{L}_{\mathrm{FO}}$ using Monte Carlo averages over the batch of data $\mathcal{B} = \{x_i\}_{i=1}^{B}$, where $x_i \overset{\mathrm{iid}}{\sim} \mathcal{D}$, with the corresponding losses denoted by $\widehat{\mathcal{L}}_{\mathrm{PD},\mathcal{B}}$, $\widehat{\mathcal{L}}_{D,\mathcal{B}}$, and $\widehat{\mathcal{L}}_{\mathrm{FO},\mathcal{B}}$. At each training iteration, we approximate $\mathcal{L}_{\mathrm{MANIFEST}}$ using the batch loss $\widehat{\mathcal{L}}_{\mathrm{MANIFEST},\mathcal{B}}$ and solve (21) using alternating stochastic gradient descent in θ and gradient ascent in ϕ. At inference time, the primal network directly computes the raw prediction $y_\theta(x)$ without the dual network. If additional equality-feasibility enforcement is required at inference time, we can use the Gauss-Newton iteration in (18) on the raw prediction $y_\theta(x)$ until the equality residual reaches the prescribed tolerance, yielding the restored prediction $\widehat{y}_\theta(x)$. The complete training procedure for MANIFEST is summarized in Algorithm 1. We also establish some additional theoretical results and intuition to support our algorithm design in Appendix A.2.

4 Numerical Experiments

We evaluate MANIFEST across a broad spectrum of parametric constrained optimization problems, ranging from structured convex QCQPs with affine equalities, to convex NLPs with nonlinear inequality geometry, and finally to nonconvex AC-OPF with nonlinear power-balance equalities. This progression is designed to assess the generality of MANIFEST across increasingly challenging constraint geometries and optimization landscapes.

Baselines. We include three state-of-the-art learning-based approaches for constrained optimization, DC3(Donti et al., 2021), FSNet (Nguyen & Donti, 2025), and NLPOpt-Net (Roy et al., 2026).

Evaluation metrics. For the QCQP and NLP problems, optimality performance is measured by the relative optimality gap $\mathrm{Gap}(x, y) = |c(x, y) - c(x, y^\star)| / \big(|c(x, y^\star)| + \varepsilon\big)$, where $y^\star$ denotes the reference optimal solution. Feasibility is measured by the absolute equality residual $|g(x, y)|$ and the inequality violation $[h(x, y)]_+ = \max\{0, h(x, y)\}$, both applied componentwise. We report the mean and maximum equality and inequality violations over the test set. These metrics are evaluated for both the *raw neural prediction* and the output *after feasibility correction*.

The implementation details for MANIFEST are provided in Appendix B.

4.1 Parametric Quadratically Constrained Quadratic Programs

We first consider convex parametric QCQPs

$$\begin{aligned} \min_{y} \quad & \frac{1}{2} y^\top Q y + c^\top y \\ \text{subject to} \quad & Ay = b + Bx, \\ & y^\top C_i y + d_i^\top y \le \beta_i + E_i x, \qquad i = 1, \dots, n_{\text{ineq}}, \\ & l + Lx \le y \le u + Ux, \end{aligned} \tag{22}$$

where $Q \succeq 0$ and $C_i \succeq 0$. The benchmark contains 100 decision variables, 50 parameters, 50 equality constraints, and 50 inequality constraints.

Results. Table 1 shows that MANIFEST substantially improves the feasibility of the learned map before any correction. Its raw prediction achieves mean and maximum equality violations of 2.7×10^{-5} and 8.4×10^{-5}, respectively, more than one order of magnitude below FSNet and NLPOpt-Net, while maintaining a small optimality gap. Restoration then drives the remaining equality violation to numerical zero with essentially no change in the objective. In contrast, we note that the baseline methods include additional objective optimization during the post-training correction procedure; thus, their improved post-correction objective gaps reflect optimization performed beyond the raw neural prediction.

4.2 Parametric Nonlinear Programs

We next consider convex parametric nonlinear programs

$$\begin{aligned} \min_{y} \quad & \frac{1}{2} y^\top Q y + c^\top y \\ \text{subject to} \quad & Ay = b + Bx, \\ & a_i^\top \exp(y) + y^\top W_i y \le \beta_i + E_i x, \qquad i = 1, \dots, n_{\text{ineq}}, \\ & l + Lx \le y \le u + Ux, \end{aligned} \tag{23}$$

where $Q \succeq 0$, $W_i \succeq 0$, and the exponential is applied componentwise. The benchmark contains 100 decision variables, 50 parameters, 50 equality constraints, and 50 nonlinear inequality constraints.

Results. The nonlinear benchmark provides a more challenging constrained optimization problem than QCQP. From Table 1, MANIFEST achieves a raw optimality gap of 8.0×10^{-8}, **nearly six orders of magnitude smaller** than FSNet and NLPOpt-Net, while reducing the mean equality violation to 4.4×10^{-9} and the maximum equality violation to 1.5×10^{-8}, about **four to five orders of magnitude below** the competing baselines. A small restoration subsequently drives these already negligible residuals to numerical zero without changing the optimality gap, demonstrating that both near-optimality and near-feasibility are already captured by the learned solution map.

4.3 Nonlinear AC Optimal Power Flow

We finally evaluate MANIFEST on the nonconvex AC optimal power flow (AC-OPF) problem (see Appendix C for the detailed formulation), using the IEEE 57-bus standard test system (University of Washington Electrical Engineering). Online solution of the AC-OPF is of great practical importance in power grid applications, where fast solutions to the problem are necessary to balance supply and demand under rapidly varying conditions during dispatch (Wood et al. (1984)). This is a problem class that is NP-hard in general (Bienstock & Verma (2019)), and presents a challenging nonlinear non-convex optimization landscape. The problem parameters are the active and reactive power demands $x = (P_d, Q_d)$, and the neural network predicts the full decision vector $y = (P_g, Q_g, V_r, V_i)$, comprising generator active and reactive powers and bus voltages. The nonlinear equality constraints enforce AC power balance, while the inequality constraints impose generator, voltage, and branch-flow limits. In this setting, all predictions are referenced with respect to solutions obtained using PYPOWER (Lincoln (2015)) to characterize the optimality gap.

Table 1: **Test results on Parametric QCQP, Parametric NLP, and AC-OPF.** Optimality gap and constraint violations for raw prediction (left) and after feasibility correction (right). Opt. gap: relative gap to the reference solution; Mean/Max eq. and ineq. mean and maximum constraint violations. NaN: no valid finite solution; values below 10^{-12} are treated as numerical zero.

Method	Raw neural prediction					After feasibility correction				
	Opt. gap	Mean eq.	Max eq.	Mean ineq.	Max ineq.	Opt. gap	Mean eq.	Max eq.	Mean ineq.	Max ineq.
QCQP										
DC3	3.5e-2	1.3e-2	8.2e-2	0	0	6.5e-3	3.0e-3	1.8e-2	0	0
FSNet	**2.3e-4**	3.0e-4	2.3e-3	0	0	1.8e-7	0	0	0	0
NLPOpt-Net	7.5e-4	1.0e-3	8.2e-3	0	0	**8.3e-9**	0	0	0	0
MANIFEST	5.2e-3	**2.7e-5**	**8.4e-5**	**0**	**0**	5.2e-3	**0**	**0**	**0**	**0**
NLP										
DC3	4.6	1.3e-2	7.7e-2	0	0	8.8e-1	2.9e-3	1.7e-2	0	0
FSNet	2.2e-2	2.1e-4	1.7e-3	0	0	2.5e-6	0	0	0	0
NLPOpt-Net	9.8e-2	1.0e-3	7.5e-3	0	0	1.0e-6	7.7e-12	2.4e-10	0	0
MANIFEST	**8.0e-8**	**4.4e-9**	**1.5e-8**	**0**	**0**	**4.3e-8**	**0**	**0**	**0**	**0**
AC-OPF										
DC3	5.4e-3	5.3e-3	6.4e-2	**2.5e-10**	1.1e-4	5.4e-3	4.3e-3	4.2e-2	**6.2e-10**	1.1e-4
FSNet	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NLPOpt-Net	1.5e-1	1.3e2	1.4e3	2.4	1.6e2	5.4e-2	1.8e1	1.6e2	4.4e-2	1.0e1
MANIFEST	**6.6e-5**	**7.5e-6**	**4.5e-5**	1.7e-8	**3.6e-6**	**6.6e-5**	**0**	**0**	1.7e-8	**3.7e-6**

For AC-OPF, existing methods including DC3 and FSNet typically simplify the learning task by predicting only a subset of the optimization variables and recovering the remaining variables through a problem-specific power-flow or algebraic solver. While effective for AC-OPF, this approach delegates satisfaction of the most challenging nonlinear equality constraints to a conventional solver rather than requiring the learned map itself to represent the full feasible solution. As our goal is to develop neural solvers for general constrained optimization without physics priors or problem-specific completion, we treat AC-OPF as a general nonlinear program and require each method to directly predict the complete decision vector without problem-specific completion. We therefore evaluate all methods under this common formulation.

Results. From Table 1, we observe that `MANIFEST` remains effective in the substantially more difficult nonconvex AC-OPF setting, where the nonlinear power-balance equations induce a highly nonconvex feasible landscape. `MANIFEST` attains a raw relative objective difference of 6.6×10^{-5}, compared with 5.4×10^{-3} for DC3 and 1.5×10^{-1} for NLPOpt-Net, while simultaneously reducing the mean power-balance violation to 7.5×10^{-6}. This corresponds to nearly **three orders of magnitude lower** mean equality violation than DC3 with comparable inequality residuals, and over **seven orders of magnitude lower** than NLPOpt-Net. FSNet fails to produce finite solutions under for the considered AC-OPF problem. After restoration, `MANIFEST` reduces the remaining power-balance residual to numerical zero with essentially no change in objective value, demonstrating that its raw neural prediction already lies close to a high-quality feasible solution even in a strongly nonconvex optimization landscape. These results demonstrate that the advantage of `MANIFEST` extends beyond structured convex benchmarks to complex nonconvex problems with nonlinear equality geometry.

5 Conclusion

We introduced `MANIFEST`, a manifold-informed framework for unsupervised learning of constrained parametric optimization problems. `MANIFEST` leverages the local geometry of the equality-feasible manifold to train neural predictions toward feasibility within a primal-dual learning framework. Across QCQPs, nonlinear programs, and nonconvex AC optimal power flow, `MANIFEST` improves raw-prediction feasibility by orders of magnitude over existing methods while maintaining near-optimal objective values.

AI USAGE DISCLOSURE

Generative AI tools, including large language models, were used to assist with polishing the writing and improving the clarity of the manuscript, as well as for assistance with coding. All AI-assisted text and code were reviewed and verified by the authors. The authors take full responsibility for the content and intellectual contributions of the paper and code.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Brandon Amos. Tutorial on amortized optimization. *Foundations and Trends in Machine Learning*, 16(5):592–732, 2023. doi: 10.1561/2200000102. URL https://arxiv.org/abs/2202.00665.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning (ICML)*, pp. 136–145, 2017.

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

Nicolas Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022a.

Wenbo Chen, Seonho Park, Mathieu Tanneau, and Pascal Van Hentenryck. Learning optimization proxies for large-scale security-constrained economic dispatch. *Electric Power Systems Research*, 213:108566, 2022b. doi: 10.1016/j.epsr.2022.108566.

Steven Diamond and Stephen Boyd. CVXPY: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Priya L. Donti, David Rolnick, and J. Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations (ICLR)*, 2021.

Ján Drgoňa, Aaron Tuor, and Draguna Vrabie. Learning constrained parametric differentiable predictive control policies with guarantees. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(6):3596–3607, 2024. doi: 10.1109/TSMC.2024.3368026.

Ferdinando Fioretto, Terrence W. K. Mak, and Pascal Van Hentenryck. Predicting AC optimal power flows: Combining deep learning and Lagrangian dual methods. In *AAAI Conference on Artificial Intelligence*, pp. 630–637, 2020.

Ferdinando Fioretto, Pascal Van Hentenryck, Terrence W. K. Mak, Cuong Tran, Federico Baldo, and Michele Lombardi. Lagrangian duality for constrained deep learning. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track*, volume 12461 of *Lecture Notes in Computer Science*, pp. 118–135. Springer, 2021. doi: 10.1007/978-3-030-67670-4_8.

John J Grainger, William D Stevenson, and Gary W Chang. Power system analysis. 2016.

Michael Hertneck, Johannes Köhler, Sebastian Trimpe, and Frank Allgöwer. Learning an approximate model predictive controller with guarantees. *IEEE Control Systems Letters*, 2(3):543–548, 2018. doi: 10.1109/LCSYS.2018.2843682.

Dou Huang, Haoran Zhang, Xuan Song, and Ryosuke Shibasaki. Differentiable projection for constrained deep learning. *arXiv preprint arXiv:2111.10785*, 2021.

James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder. End-to-end constrained optimization learning: A survey. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4475–4482, 2021.

Giacomo Lastrucci and Artur M. Schweidtmann. ENFORCE: Exact nonlinear constrained learning with adaptive-depth neural projection. *arXiv preprint arXiv:2502.06774*, 2025. doi: 10.48550/arXiv.2502.06774.

Viet-Anh Le, Mu Xie, and Rahul Mangharam. A hybrid learning-to-optimize framework for mixed-integer quadratic programming. In *Learning for Dynamics and Control Conference*, 2026. URL https://arxiv.org/abs/2511.19383.

Shimiao Li, Aaron Tuor, Draguna Vrabie, Larry Pileggi, and Ján Drgoňa. Homotopy-guided self-supervised learning of parametric solutions for AC optimal power flow, 2025. URL https://arxiv.org/abs/2511.11677.

Enming Liang, Minghua Chen, and Steven H. Low. Homeomorphic projection to ensure neural-network solution feasibility for constrained optimization. *Journal of Machine Learning Research*, 25(329):1–55, 2024. URL https://jmlr.org/papers/v25/23-1577.html.

Richard Lincoln. PYPOWER: A python port of matpower. https://github.com/rwl/PYPOWER, 2015. Power flow and optimal power flow package.

Youngjae Min, Anoopkumar Sonar, and Navid Azizan. HardNet: Hard-constrained neural networks with universal approximation guarantees. *arXiv preprint arXiv:2410.10807*, 2024.

Sidhant Misra, Line Roald, and Yeesian Ng. Learning for constrained optimization: Identifying optimal active constraint sets. *INFORMS Journal on Computing*, 34(1):463–480, 2022. doi: 10.1287/ijoc.2020.1037.

Hoang T. Nguyen and Priya L. Donti. FSNet: Feasibility-seeking neural network for constrained optimization with guarantees. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/3874e2be479a9d4e94d4514046c1f934-Abstract-Conference.html.

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, 2021.

Seonho Park and Pascal Van Hentenryck. Self-supervised primal-dual learning for constrained optimization. In *AAAI Conference on Artificial Intelligence*, pp. 4052–4060, 2023.

Bimol Nath Roy, Rahul Golder, and M. M. Faruque Hasan. NLPOpt-Net: A learning method for nonlinear optimization with feasibility guarantees, 2026. URL https://arxiv.org/abs/2605.00260.

Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D. Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018. doi: 10.1109/TSP.2018.2866382. URL https://doi.org/10.1109/TSP.2018.2866382.

Mathieu Tanneau and Pascal Van Hentenryck. Dual lagrangian learning for conic optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 55538–55561, 2024.

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

University of Washington Electrical Engineering. Power systems test case archive. https://labs.ece.uw.edu/pstca/. Accessed: 2026-09-24.

Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. *Problem Solutions for Power Generation, Operation, and Control*. Wiley, 1984.

Ahmed S. Zamzam and Kyri Baker. Learning optimal solutions for extremely fast AC optimal power flow. In *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020.

A Appendix

A.1 Proofs of Theoretical Results

We provide the proofs of all theoretical results in this section.

Proposition 1 (Local equality-manifold geometry)**.** *Under Assumption 1 with $y \in \mathcal{E}_x$ and $r = \operatorname{rank} J_y(x,y)$, there exists an open neighborhood $\mathcal{V}_y \subseteq \mathcal{U}_y$ of y such that $\mathcal{E}_x \cap \mathcal{V}_y$ is a smooth embedded submanifold of $\mathbb{R}^n$ of dimension $n-r$. The tangent and normal spaces at y satisfy*

$$T_y\mathcal{E}_x = \ker J_y(x,y), \qquad N_y\mathcal{E}_x = \operatorname{range}\big(J_y(x,y)^\top\big)\,. \tag{24}$$

Proof. Fix $x \in \mathcal{X}$ and $y \in \mathcal{E}_x$, and let $r = \operatorname{rank} J_y(x,y)$. Assumption 1 gives

$$g(x,y) = 0, \quad \operatorname{rank} J_y(x,\widetilde{y}) = r, \quad \widetilde{y} \in \mathcal{U}_y.$$

By the constant-rank theorem (Boumal, 2023), there exist an open neighborhood $\mathcal{V}_y \subseteq \mathcal{U}_y$ and a C^2 bijection $\varphi : \mathcal{V}_y \to \varphi(\mathcal{V}_y) \subseteq \mathbb{R}^n$ with open image and C^2 inverse such that

$$\varphi(y) = 0, \quad \varphi(\mathcal{E}_x \cap \mathcal{V}_y) = \varphi(\mathcal{V}_y) \cap \big(\{0\}^r \times \mathbb{R}^{n-r}\big).$$

Thus $\mathcal{E}_x \cap \mathcal{V}_y$ is a C^2 embedded submanifold of $\mathbb{R}^n$, with

$$\dim(\mathcal{E}_x \cap \mathcal{V}_y) = n - r.$$

For any $\xi \in T_y\mathcal{E}_x$, take a C^1 curve $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{E}_x \cap \mathcal{V}_y$ with $\gamma(0) = y$ and $\gamma'(0) = \xi$. Then

$$\begin{aligned} &g(x, \gamma(t)) = 0, \\ &0 = \frac{d}{dt} g(x, \gamma(t))\bigg|_{t=0} = D_y g(x,y)[\gamma'(0)] = J_y(x,y)\xi. \end{aligned}$$

Consequently,

$$\begin{aligned} T_y\mathcal{E}_x &\subseteq \ker J_y(x,y), \\ \dim T_y\mathcal{E}_x &= n - r = n - \operatorname{rank} J_y(x,y) = \dim \ker J_y(x,y). \end{aligned}$$

The inclusion and equality of dimensions give $T_y\mathcal{E}_x = \ker J_y(x,y)$. Taking Euclidean orthogonal complements,

$$N_y\mathcal{E}_x = (T_y\mathcal{E}_x)^\perp = \big(\ker J_y(x,y)\big)^\perp = \operatorname{range}\big(J_y(x,y)^\top\big)\,.$$

□

Proposition 2 (Local distance and normal direction)**.** *Suppose that $J_\theta(x)$ has full row rank. The minimum perturbation in (7) satisfies*

$$\Delta y^\star_{\mathrm{lin}}(x) = -J_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top\right)^{-1} r_\theta(x) \in \mathcal{N}_x^{\mathrm{lin}}(y_\theta(x))\,. \tag{25}$$

Moreover,

$$\|\Delta y^\star_{\mathrm{lin}}(x)\|_2^2 = r_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top\right)^{-1} r_\theta(x). \tag{26}$$

Proof. Fix x. Since $J_\theta(x)$ has full row rank, for every $a \in \mathbb{R}^m \setminus \{0\}$,

$$a^\top J_\theta(x) J_\theta(x)^\top a = \|J_\theta(x)^\top a\|_2^2 > 0.$$

Hence $J_\theta(x) J_\theta(x)^\top$ is invertible. Consider the candidate

$$\Delta y^\star_{\mathrm{lin}}(x) := -J_\theta(x)^\top \big(J_\theta(x) J_\theta(x)^\top\big)^{-1} r_\theta(x).$$

It satisfies

$$\begin{aligned} J_\theta(x)\Delta y^\star_{\mathrm{lin}}(x) &= -J_\theta(x) J_\theta(x)^\top \big(J_\theta(x) J_\theta(x)^\top\big)^{-1} r_\theta(x) = -r_\theta(x), \\ \Delta y^\star_{\mathrm{lin}}(x) &\in \operatorname{range}\big(J_\theta(x)^\top\big) = \mathcal{N}_x^{\mathrm{lin}}(y_\theta(x))\,. \end{aligned}$$

For any other feasible displacement Δy,

$$r_\theta(x) + J_\theta(x)\Delta y = 0,$$
$$J_\theta(x)\big(\Delta y - \Delta y^\star_{\text{lin}}(x)\big) = 0.$$

Therefore,

$$\langle \Delta y^\star_{\text{lin}}(x), \Delta y - \Delta y^\star_{\text{lin}}(x)\rangle = -r_\theta(x)^\top \big(J_\theta(x)J_\theta(x)^\top\big)^{-1} \cdot J_\theta(x)\big(\Delta y - \Delta y^\star_{\text{lin}}(x)\big) = 0.$$

It follows that

$$\|\Delta y\|_2^2 = \|\Delta y^\star_{\text{lin}}(x)\|_2^2 + \|\Delta y - \Delta y^\star_{\text{lin}}(x)\|_2^2 \geq \|\Delta y^\star_{\text{lin}}(x)\|_2^2,$$

with equality only if $\Delta y = \Delta y^\star_{\text{lin}}(x)$. Thus the candidate is the unique minimizer of (7).

Finally, symmetry of $\big(J_\theta(x)J_\theta(x)^\top\big)^{-1}$ gives

$$\begin{aligned}\|\Delta y^\star_{\text{lin}}(x)\|_2^2 &= r_\theta(x)^\top \big(J_\theta(x)J_\theta(x)^\top\big)^{-1} J_\theta(x)J_\theta(x)^\top \cdot \big(J_\theta(x)J_\theta(x)^\top\big)^{-1} r_\theta(x) \\ &= r_\theta(x)^\top \big(J_\theta(x)J_\theta(x)^\top\big)^{-1} r_\theta(x).\end{aligned}$$

□

Proposition 3 (First-order manifold feasibility)**.** *Let $F : \mathcal{X} \to \mathbb{R}^n$ be differentiable and satisfy $g(z, F(z)) = 0$ for every z in an open neighborhood of x contained in $\mathcal{X}$. Then,*

$$D_x g(x, F(x)) + D_y g(x, F(x)) D_x F(x) = 0. \tag{27}$$

Proof. Fix x in the open parameter neighborhood specified in the proposition. For every $v \in \mathbb{R}^p$ and all sufficiently small $|t|$, feasibility gives

$$g(x + tv, F(x + tv)) = 0.$$

Differentiating at $t = 0$ yields

$$\begin{aligned}0 &= \frac{d}{dt} g(x + tv, F(x + tv))\Big|_{t=0} \\ &= D_x g(x, F(x))[v] + D_y g(x, F(x)) D_x F(x)[v] \\ &= \big[D_x g(x, F(x)) + D_y g(x, F(x)) D_x F(x)\big] v.\end{aligned}$$

Since this holds for every $v \in \mathbb{R}^p$, we have

$$D_x g(x, F(x)) + D_y g(x, F(x)) D_x F(x) = 0.$$

□

Proposition 4 (Regularized equality error)**.** *Fix $x \in \mathcal{X}$, $y \in \mathbb{R}^n$, and $\delta_r > 0$. The problem*

$$\min_{e \in \mathbb{R}^n} \left\{ \frac{1}{2} \|g(x, y) - J_y(x, y)e\|_2^2 + \frac{\delta_r}{2} \|e\|_2^2 \right\} \tag{28}$$

has the unique solution

$$e_{\delta_r}(x, y) = J_y(x, y)^\top \left(J_y(x, y)J_y(x, y)^\top + \delta_r I_m\right)^{-1} g(x, y) \in \text{range}\big(J_y(x, y)^\top\big). \tag{29}$$

If $J_y(x, y)$ has full row rank, then $e_0(x, y) = J_y(x, y)^\top \left(J_y(x, y)J_y(x, y)^\top\right)^{-1} g(x, y)$ is the minimum-norm solution of $J_y(x, y)e = g(x, y)$. In other words, $y - e_0(x, y)$ satisfies the linearized equality constraints.

Proof. Fix $x \in \mathcal{X}$, $y \in \mathbb{R}^n$, and $\delta_r > 0$. Differentiating the objective in (16) with respect to e gives

$$\nabla_e \left[\frac{1}{2}\|g(x, y) - J_y(x, y)e\|_2^2 + \frac{\delta_r}{2}\|e\|_2^2\right] = \big(J_y(x, y)^\top J_y(x, y) + \delta_r I_n\big)e - J_y(x, y)^\top g(x, y).$$

For $z \in \mathbb{R}^n \setminus \{0\}$ and $a \in \mathbb{R}^m \setminus \{0\}$,

$$z^\top \big(J_y(x,y)^\top J_y(x,y) + \delta_r I_n\big) z = \|J_y(x,y)z\|_2^2 + \delta_r \|z\|_2^2 > 0,$$
$$a^\top \big(J_y(x,y) J_y(x,y)^\top + \delta_r I_m\big) a = \|J_y(x,y)^\top a\|_2^2 + \delta_r \|a\|_2^2 > 0.$$

Thus the objective is strictly convex, both matrices are invertible, and the unique minimizer satisfies

$$\big(J_y(x,y)^\top J_y(x,y) + \delta_r I_n\big) e_{\delta_r}(x,y) = J_y(x,y)^\top g(x,y).$$

Using

$$\big(J_y(x,y)^\top J_y(x,y) + \delta_r I_n\big) J_y(x,y)^\top = J_y(x,y)^\top \big(J_y(x,y) J_y(x,y)^\top + \delta_r I_m\big),$$

we obtain

$$\begin{aligned} e_{\delta_r}(x,y) &= \big(J_y(x,y)^\top J_y(x,y) + \delta_r I_n\big)^{-1} J_y(x,y)^\top g(x,y) \\ &= J_y(x,y)^\top \big(J_y(x,y) J_y(x,y)^\top + \delta_r I_m\big)^{-1} g(x,y) \in \operatorname{range}\big(J_y(x,y)^\top\big). \end{aligned}$$

If $J_y(x,y)$ has full row rank, then $J_y(x,y)J_y(x,y)^\top$ is invertible. The candidate

$$e_0(x,y) := J_y(x,y)^\top \big(J_y(x,y) J_y(x,y)^\top\big)^{-1} g(x,y) \in \operatorname{range}\big(J_y(x,y)^\top\big)$$

satisfies

$$J_y(x,y) e_0(x,y) = J_y(x,y) J_y(x,y)^\top \big(J_y(x,y) J_y(x,y)^\top\big)^{-1} g(x,y) = g(x,y).$$

For any e satisfying $J_y(x,y)e = g(x,y)$, we have $J_y(x,y)\big(e - e_0(x,y)\big) = 0$, and thus

$$\langle e_0(x,y), e - e_0(x,y)\rangle = g(x,y)^\top \big(J_y(x,y) J_y(x,y)^\top\big)^{-1} \cdot J_y(x,y)\big(e - e_0(x,y)\big) = 0.$$

Hence

$$\|e\|_2^2 = \|e_0(x,y)\|_2^2 + \|e - e_0(x,y)\|_2^2 \geq \|e_0(x,y)\|_2^2,$$

with equality holding only if $e = e_0(x,y)$. Therefore $e_0(x,y)$ is the unique minimum-norm solution of $J_y(x,y)e = g(x,y)$. Finally,

$$g(x,y) + J_y(x,y)\big[(y - e_0(x,y)) - y\big] = g(x,y) - J_y(x,y) e_0(x,y) = 0,$$

so $y - e_0(x,y)$ satisfies the linearized equality constraints. □

A.2 Additional Theoretical Results

Proposition 2 shows that the unregularized geometric distance is the squared minimum-norm displacement required to satisfy the linearized equality constraints. We next show that this quantity, and the corresponding displacement, are invariant to nonsingular linear reparameterizations of the equality constraints.

Proposition 5 (Constraint-reparameterization invariance)**.** *Suppose that $J_\theta(x)$ has full row rank, and let $R \in \mathbb{R}^{m\times m}$ be nonsingular. Consider the equivalent reparameterization of the equality constraints*

$$\widetilde{g}(x,y) := Rg(x,y). \tag{30}$$

At the neural prediction $y_\theta(x)$, define

$$\widetilde{r}_\theta(x) := Rr_\theta(x), \qquad \widetilde{J}_\theta(x) := RJ_\theta(x). \tag{31}$$

Then the minimum-norm linearized displacement is unchanged:

$$-\widetilde{J}_\theta(x)^\top \left(\widetilde{J}_\theta(x)\widetilde{J}_\theta(x)^\top\right)^{-1} \widetilde{r}_\theta(x) = -J_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top\right)^{-1} r_\theta(x). \tag{32}$$

Consequently, the unregularized geometric distance is invariant:

$$\widetilde{r}_\theta(x)^\top \left(\widetilde{J}_\theta(x)\widetilde{J}_\theta(x)^\top\right)^{-1} \widetilde{r}_\theta(x) = r_\theta(x)^\top \left(J_\theta(x) J_\theta(x)^\top\right)^{-1} r_\theta(x). \tag{33}$$

Proof. Fix x and abbreviate $J := J_\theta(x)$ and $r := r_\theta(x)$. Under the transformed constraints, $\widetilde{J} = RJ$ and $\widetilde{r} = Rr$. Since R is nonsingular and J has full row rank,

$$\left(\widetilde{J}\widetilde{J}^\top\right)^{-1} = \left(RJJ^\top R^\top\right)^{-1} = R^{-\top}\left(JJ^\top\right)^{-1}R^{-1}.$$

Therefore,

$$-\widetilde{J}^\top\left(\widetilde{J}\widetilde{J}^\top\right)^{-1}\widetilde{r} = -J^\top R^\top R^{-\top}\left(JJ^\top\right)^{-1}R^{-1}Rr = -J^\top\left(JJ^\top\right)^{-1}r,$$

which proves (32). Similarly,

$$\widetilde{r}^\top\left(\widetilde{J}\widetilde{J}^\top\right)^{-1}\widetilde{r} = r^\top R^\top R^{-\top}\left(JJ^\top\right)^{-1}R^{-1}Rr = r^\top\left(JJ^\top\right)^{-1}r,$$

which proves (33). □

Unlike the ordinary residual penalty $\|g(x,y)\|_2^2$, whose magnitude and relative weighting of constraints can change under rescaling or linear mixing of the equalities, the unregularized geometric distance is invariant to such equivalent representations. This distinction is important for learning, as a residual penalty can produce different training signals for the same feasible set depending on the numerical scaling or parameterization of the constraints. In contrast, the `MANIFEST` distance measures the minimum-norm displacement required to satisfy the locally linearized constraints and therefore reflects the local geometry of the feasible set rather than the particular representation of its defining equations. This provides a geometrically normalized feasibility signal and distinguishes the proposed loss from directly penalizing equality residuals. Note that the exact invariance in Proposition 5 applies to the unregularized case $\delta_d = 0$; in practice, a small positive regularization can be used for stability if the constraint Jacobian is poorly conditioned or rank deficient.

The invariance result concerns the unregularized distance. We next characterize how regularization changes this quantity and its relationship to the Gauss-Newton correction.

Proposition 6 (Properties of the regularized geometric distance)**.** *Fix x, y, write $r := g(x,y)$ and $J := J_y(x,y)$, and let $\delta > 0$. Define $D_\delta^2(x,y) := r^\top(JJ^\top + \delta I_m)^{-1}r$ and $e_\delta(x,y) := J^\top(JJ^\top + \delta I_m)^{-1}r$. Let $J = U_s\Sigma_s V_s^\top$ be a compact singular-value decomposition, where $s = \operatorname{rank} J$, and set $r_\perp := (I_m - U_s U_s^\top)r$. Then*

$$D_\delta^2(x,y) = \sum_{i=1}^{s}\frac{(u_i^\top r)^2}{\sigma_i^2+\delta} + \frac{\|r_\perp\|_2^2}{\delta},$$

$$\|e_\delta(x,y)\|_2^2 = \sum_{i=1}^{s}\frac{\sigma_i^2(u_i^\top r)^2}{(\sigma_i^2+\delta)^2} \le D_\delta^2(x,y).$$

If J has full row rank, with $D_0^2(x,y) := r^\top(JJ^\top)^{-1}r$, we have

$$\frac{\sigma_{\min}(J)^2}{\sigma_{\min}(J)^2+\delta}D_0^2(x,y) \le D_\delta^2(x,y) \le D_0^2(x,y).$$

Proof. The singular-value decomposition gives

$$(JJ^\top + \delta I_m)^{-1}r = \sum_{i=1}^{s}\frac{u_i^\top r}{\sigma_i^2+\delta}u_i + \frac{r_\perp}{\delta},$$

$$e_\delta(x,y) = \sum_{i=1}^{s}\frac{\sigma_i(u_i^\top r)}{\sigma_i^2+\delta}v_i.$$

Orthogonality yields the stated representations, and $\sigma_i^2/(\sigma_i^2+\delta) \le 1$ proves $\|e_\delta(x,y)\|_2^2 \le D_\delta^2(x,y)$. For full row rank, $r_\perp = 0$ and $D_0^2(x,y) = \sum_{i=1}^{m}(u_i^\top r)^2/\sigma_i^2$. Comparing the corresponding summands using

$$\frac{\sigma_{\min}(J)^2}{\sigma_{\min}(J)^2+\delta} \le \frac{\sigma_i^2}{\sigma_i^2+\delta} \le 1$$

proves the remaining bounds. □

Proposition 6 shows that the correction magnitude and the geometric distance capture different information about equality infeasibility. Specifically, the correction magnitude is determined by the displacement produced by the correction map, whereas the geometric distance directly weights the equality residual through the local constraint Jacobian and additionally penalizes $r_\perp$, which cannot be removed by the linearized correction. Thus, methods that penalize the correction magnitude may not fully capture the remaining equality infeasibility and can still rely on the subsequent correction procedure to enforce feasibility. This result provides further motivation for using the geometric distance directly as a training signal in `MANIFEST`.

The geometric distance loss measures equality infeasibility at a given problem instance; the following result shows how the first-order loss complements it by controlling residual variation with the problem parameters.

Proposition 7 (First-order loss and parameter sensitivity)**.** *Let $r_\theta(x) := g(x, F_\theta(x))$ be differentiable, and write $A_\theta(x) := D_x r_\theta(x)$. Suppose $V \neq 0$ almost surely, V is independent of X, and the expectations below are finite. With $M_V := \mathbb{E}_V[VV^\top/\|V\|_2^2]$, we have*

$$\mathcal{L}_{\mathrm{FO}}(\theta) = \mathbb{E}_X\left[\mathrm{tr}\big(A_\theta(X)^\top A_\theta(X) M_V\big)\right].$$

For $M_V = I_p/p$, this becomes $\mathcal{L}_{\mathrm{FO}}(\theta) = p^{-1}\mathbb{E}_X\|A_\theta(X)\|_F^2$. If A_θ is Lipschitz with constant L_r in operator norm on a convex neighborhood containing x and $x + \Delta x$, then

$$\|r_\theta(x+\Delta x)\|_2 \le \|r_\theta(x)\|_2 + \|A_\theta(x)\|_F\|\Delta x\|_2 + \frac{L_r}{2}\|\Delta x\|_2^2.$$

Proof. Fix x and abbreviate $A := A_\theta(x)$. By the trace identity,

$$\mathbb{E}_V\left[\frac{\|AV\|_2^2}{\|V\|_2^2}\right] = \mathrm{tr}\left(A^\top A\,\mathbb{E}_V\left[\frac{VV^\top}{\|V\|_2^2}\right]\right) = \mathrm{tr}(A^\top A M_V).$$

Taking expectation over X proves the first identity. When $M_V = I_p/p$, $\mathrm{tr}(A^\top A M_V) = p^{-1}\|A\|_F^2$, which proves the second identity. For the perturbation bound, write $r_\theta(x+\Delta x) = r_\theta(x) + A\Delta x + R$. The fundamental theorem of calculus and the Lipschitz assumption give

$$R = \int_0^1 \big(A_\theta(x + t\Delta x) - A\big)\Delta x\, dt,$$

$$\|R\|_2 \le \int_0^1 L_r t\|\Delta x\|_2^2\, dt = \frac{L_r}{2}\|\Delta x\|_2^2.$$

The conclusion follows from the triangle inequality and $\|A\|_{\mathrm{op}} \le \|A\|_F$. □

When $M_V = I_p/p$, the first-order loss satisfies $\mathcal{L}_{\mathrm{FO}}(\theta) = p^{-1}\mathbb{E}_X\|D_x r_\theta(X)\|_F^2$. Thus, $\mathcal{L}_D$ penalizes the geometric distance associated with the raw prediction, while $\mathcal{L}_{\mathrm{FO}}$ penalizes how rapidly its equality residual $r_\theta(x)$ changes with x. The perturbation bound shows that both $\|r_\theta(x)\|_2$ and $\|D_x r_\theta(x)\|_F$ determine the residual at nearby inputs.

We next quantify the equality-feasibility improvement obtained from a Gauss-Newton step.

Proposition 8 (Local equality-feasibility improvement)**.** *Fix x, and suppose $J_y(x,\cdot)$ is Lipschitz with constant L_J in operator norm on a convex neighborhood $\mathcal{U}$. Let $y \in \mathcal{U}$ satisfy $\mathrm{rank}\, J_y(x,y) = m$ and $\sigma_{\min}(J_y(x,y)) \ge s_0 > 0$. For $\delta_r \ge 0$ and $\alpha_r \in (0,1]$, suppose $y^+ := y - \alpha_r e_{\delta_r}(x,y) \in \mathcal{U}$. Then, we have*

$$\|g(x,y^+)\|_2 \le \left(1 - \frac{\alpha_r s_0^2}{s_0^2 + \delta_r}\right)\|g(x,y)\|_2 + \frac{L_J\alpha_r^2}{2s_0^2}\|g(x,y)\|_2^2.$$

Proof. Write $r := g(x,y)$, $J := J_y(x,y)$, and $e := e_{\delta_r}(x,y)$. Since $\sigma_i/(\sigma_i^2 + \delta_r) \le 1/\sigma_i \le 1/s_0$, the singular-value decomposition gives $\|e\|_2 \le \|r\|_2/s_0$. Taylor expansion along the correction segment therefore yields

$$g(x,y^+) = r - \alpha_r J e + R,$$

$$\|R\|_2 \le \frac{L_J\alpha_r^2}{2}\|e\|_2^2 \le \frac{L_J\alpha_r^2}{2s_0^2}\|r\|_2^2.$$

The eigenvalues of $I_m - \alpha_r JJ^\top (JJ^\top + \delta_r I_m)^{-1}$ are $1 - \alpha_r \sigma_i^2/(\sigma_i^2 + \delta_r)$. Full row rank and $\alpha_r \in (0, 1]$ thus imply

$$\begin{aligned}\|r - \alpha_r Je\|_2 &= \left\| \left(I_m - \alpha_r JJ^\top (JJ^\top + \delta_r I_m)^{-1}\right) r \right\|_2 \\ &\leq \left(1 - \frac{\alpha_r s_0^2}{s_0^2 + \delta_r}\right) \|r\|_2.\end{aligned}$$

Combining these bounds proves the result. □

Corollary 1 (Repeated residual reduction)**.** *Suppose the assumptions of Proposition 8 hold uniformly along the correction iterates, with the same $s_0, L_J, \alpha_r, \delta_r$, and all iterates remain in $\mathcal{U}$. Define*

$$a_r := 1 - \frac{\alpha_r s_0^2}{s_0^2 + \delta_r}, \qquad C := \frac{L_J \alpha_r^2}{2 s_0^2}.$$

If $\|g(x, y^{(0)})\|_2 \leq \varepsilon$ for some $\varepsilon > 0$ and $\kappa := a_r + C\varepsilon < 1$, then

$$\|g(x, y^{(K)})\|_2 \leq \kappa^K \|g(x, y^{(0)})\|_2.$$

For $\delta_r = 0$ and $\alpha_r = 1$, the one-step bound reduces to

$$\|g(x, y^{(k+1)})\|_2 \leq C \|g(x, y^{(k)})\|_2^2.$$

Proof. Whenever $\|g(x, y^{(k)})\|_2 \leq \varepsilon$, the proposition gives

$$\begin{aligned}\|g(x, y^{(k+1)})\|_2 &\leq \left(a_r + C\|g(x, y^{(k)})\|_2\right) \|g(x, y^{(k)})\|_2 \\ &\leq \kappa \|g(x, y^{(k)})\|_2 \leq \varepsilon.\end{aligned}$$

Induction proves the first claim. The second follows by setting $a_r = 0$. □

These bounds explain why reducing the raw residual can improve the effectiveness of a small number of correction steps. Equality correction need not improve the cost or inequality feasibility. If $c(x, \cdot)$ and $h_j(x, \cdot)$ are Lipschitz on the relevant neighborhood, then

$$\begin{aligned}|c(x, \widetilde{y}) - c(x, y)| &\leq L_c \|\widetilde{y} - y\|_2, \\ [h_j(x, \widetilde{y})]_+ &\leq [h_j(x, y)]_+ + L_{h_j} \|\widetilde{y} - y\|_2.\end{aligned}$$

Small corrections therefore limit these changes, while evaluating the augmented loss at $\widetilde{y}$ accounts for their actual effect.

Finally, we examine the local sensitivity of the correction map used during training.

Proposition 9 (Local sensitivity of the correction map)**.** *Fix x, assume $g(x, \cdot)$ is C^2 near an equality-feasible point $y^\star$, and let $\delta_r > 0$ and $\alpha_r \in (0, 1]$. Define $T_x(y) := y - \alpha_r e_{\delta_r}(x, y)$ and $J := J_y(x, y^\star)$. Then*

$$D_y T_x(y^\star) = I_n - \alpha_r J^\top (JJ^\top + \delta_r I_m)^{-1} J. \tag{34}$$

This derivative acts as the identity on $\ker J$. For each right singular vector v_i associated with a nonzero singular value σ_i of J,

$$D_y T_x(y^\star) v_i = \left(1 - \frac{\alpha_r \sigma_i^2}{\sigma_i^2 + \delta_r}\right) v_i.$$

If J has full row rank, the result also holds for $\delta_r = 0$. With $\delta_r = 0$ and $\alpha_r = 1$, $D_y T_x(y^\star)$ is the orthogonal projector onto $\ker J$.

Proof. Set $B(y) := J_y(x, y)^\top (J_y(x, y) J_y(x, y)^\top + \delta_r I_m)^{-1}$. This map is differentiable near $y^\star$. Since $g(x, y^\star) = 0$, the product rule gives, for every $v \in \mathbb{R}^n$,

$$\begin{aligned}D_y e_{\delta_r}(x, y^\star)[v] &= D_y B(y^\star)[v]\, g(x, y^\star) + B(y^\star) J v \\ &= J^\top (JJ^\top + \delta_r I_m)^{-1} J v.\end{aligned}$$

Since $T_x(y) = y - \alpha_r e_{\delta_r}(x, y)$, this establishes (34). If $v \in \ker J$, then $D_y T_x(y^\star)v = v$. For a compact singular-value decomposition $J = U_s \Sigma_s V_s^\top$,

$$J^\top (JJ^\top + \delta_r I_m)^{-1} J = \sum_{i=1}^{s} \frac{\sigma_i^2}{\sigma_i^2 + \delta_r} v_i v_i^\top,$$

$$D_y T_x(y^\star) v_i = \left(1 - \frac{\alpha_r \sigma_i^2}{\sigma_i^2 + \delta_r}\right) v_i.$$

If J has full row rank, continuity of $J_y(x, \cdot)$ ensures that $J_y(x, y) J_y(x, y)^\top$ remains invertible near $y^\star$. The same argument therefore applies when $\delta_r = 0$. With $\delta_r = 0$ and $\alpha_r = 1$,

$$D_y T_x(y^\star) = I_n - J^\top (JJ^\top)^{-1} J = P_{\ker J},$$

where $P_{\ker J}$ denotes the orthogonal projector onto $\ker J$. □

At feasible points covered by Proposition 1, the correction preserves tangent perturbations and attenuates normal perturbations to first order. This describes its local geometric action during backpropagation; away from feasibility, derivatives of the constraint Jacobian also contribute to encourage optimality and feasibility.

B Implementation Details

We summarize the experimental details for the implementation of the MANIFEST algorithm.

B.1 Dataset Generation and Data Augmentation

QCQP and NLP. The QCQP and NLP reference datasets each contain $2{,}000$ instances with 100 decision variables, 50 parameters, 50 affine equalities, 50 nonlinear inequalities, and 200 scalar box inequalities. Parameters are sampled independently from $\mathcal{U}([-1,1]^{50})$, and problem coefficients are fixed after construction. Reference solutions to quantify the optimality gap are obtained using CVXPY (Diamond & Boyd (2016)). For both QCQP and NLP, we split the dataset into $1{,}600$ training and 400 test instances; the stored 200 validation indices belong to the training set rather than an independent held-out split.

IEEE 57-bus AC-OPF. At buses with positive base active or reactive demand, we independently sample $P_{d,b} = P^0_{d,b}\xi^P_b$ and $Q_{d,b} = Q^0_{d,b}\xi^Q_b$, with $\xi^P_b, \xi^Q_b \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0.8, 1.2)$; zero base-demand components remain zero. Each instance is solved using PYPOWER's `runopf` to confirm a feasible solution exists, and unsuccessful solves are discarded. We collect $10{,}000$ feasible data points, which are randomly divided into $8{,}000$ training and $2{,}000$ held-out test instances. The reference solutions to quantify the optimality gap are numerical AC-OPF solutions from PYPOWER (Lincoln (2015)).

Unlabeled data augmentation. As part of the MANIFEST training procedure, we augment the training inputs with additional parameter realizations drawn from the same distributions as the original training data. Specifically, we generate auxiliary datasets of $100,000$ parameter vectors for each of QCQP and NLP and $20,000$ demand vectors for AC-OPF. During training, the original and auxiliary inputs are shuffled together and sampled using the same minibatch size as in the original training setup. Since MANIFEST is trained fully unsupervised, the additional samples contain only problem parameters x drawn from the same distribution and require no solver-generated solutions or feasibility labels. The geometric distance and first-order losses are evaluated without reference solution labels, and test instances are excluded from training.

For a fair comparison, all methods use the same training-sample budget with parameter samples drawn from the same underlying problem distribution.

B.2 Experimental Details

Dual-network parameterization and regularization. The dual network produces unconstrained outputs $\widetilde{\nu}_\phi(x) \in \mathbb{R}^m$ and $\widetilde{\lambda}_\phi(x) \in \mathbb{R}^q$. We set $\nu_\phi(x) = \widetilde{\nu}_\phi(x)$ and $\lambda_\phi(x) = \mathrm{softplus}(\widetilde{\lambda}_\phi(x))$, where $\mathrm{softplus}(z) = \log(1 + e^z)$ is applied componentwise with no additional multiplier floor. Equality multipliers therefore remain unrestricted, while inequality multipliers are nonnegative by construction, without projection after optimizer updates. To prevent the dual variables from growing without bound during maximization, we include an ℓ_2 regularization term $-\rho_{\mathrm{dual}}\mathcal{R}_{\mathrm{dual}}(\phi)$ in the dual objective, where $\mathcal{R}_{\mathrm{dual}}(\phi)$ is the mean squared magnitude of the equality and inequality multipliers predicted by the dual network, and ρ_{dual} is set to 10^{-4}.

Tail-sensitive aggregation. For cases where a small number of outlier data points produce substantially larger feasibility errors than the rest of the batch, the mean loss alone can underemphasize these high-violation samples. In these cases, we can supplement the ordinary mean distance loss with a weighted mean of the largest per-instance manifold-distance scores.

Training configuration. The training configurations used for the three benchmark problems are summarized in Table 2. For the comparison methods DC3, FSNet, and NLPOpt-Net, we train each method until convergence on all three benchmark problems. No AC-OPF implementation is provided for FSNet. We therefore evaluate published FSNet code on the same generic nonlinear-program formulation used by all methods. Under this common formulation, the method did not produce finite solutions. All experiments are run on compute nodes with Intel Xeon processors and an NVIDIA A100 GPU with $80\,\mathrm{GB}$ of memory.

Table 2: Training configurations for MANIFEST.

Hyperparameter	**Value**
NLP:	
Hidden layers	2
Hidden width	512
Activation	SiLU
Maximum epochs	15,000
Source minibatch size	64
Learning rate (initial; final)	5×10^{-4}; 10^{-7}
Objective weight	100
Equality-penalty weight	100
Inequality-penalty weight	10^4
Tail weight	10^3
Gauss-Newton iterations, K	2
Restoration damping, δ_r	10^{-4}
Distance damping, δ_d	10^{-4}
Manifold-distance weight, ω_D (initial; final)	10^6; 10^{10}
First-order weight, ω_{FO}	3×10^{-2}
Gauss-Newton step size, α_r	1.0
QCQP:	
Hidden layers	2
Hidden width	512
Maximum epochs	10,000
Source minibatch size	64
Primal learning rate (initial; final)	10^{-4}; 10^{-6}
Dual learning rate (initial; final)	5×10^{-5}; 10^{-6}
Gauss-Newton iterations, K	2
Restoration damping, δ_r	10^{-4}
Distance damping, δ_d	10^{-4}
Manifold-distance weight, ω_D (initial; final)	10^6; 10^{10}
First-order weight, ω_{FO}	5×10^{-3}
Gauss–Newton step size, α_r	1.0
IEEE 57-bus AC-OPF:	
Primal architecture	4 independent MLP heads for P_g, Q_g, V_r, V_i
Dual architecture	Separate MLP
Hidden layers	4
Hidden width	1536
Activation	SiLU
Maximum epochs	30,000
Source minibatch size	128
Primal learning rate (initial; final)	2×10^{-4}; 10^{-7}
Dual learning rate (initial; final)	5×10^{-4}; 10^{-7}
Gauss-Newton iterations, K	10
Restoration damping, δ_r	10^{-3}
Distance damping, δ_d (initial; final)	10^{-3}; 10^{-6}
Manifold-distance weight, ω_D (initial; final)	10^3; 10^8
First-order weight, ω_{FO} (initial; final)	0; 0.5
Gauss-Newton step size, α_r	1.0

C AC-OPF FORMULATION

We consider a power network represented by a graph $\mathcal{G} = (\mathcal{B}, \mathcal{L})$, where $\mathcal{B} = \{1, \dots, n_b\}$ is the set of buses and $\mathcal{L}$ is the set of transmission lines. For each bus $i \in \mathcal{B}$, let $\mathcal{N}(i)$ denote the set of neighboring buses connected to i, and let $\mathcal{G}(i)$ denote the set of generators located at bus i. AC optimal power flow determines generator outputs and bus voltages that minimize generation cost while satisfying the nonlinear AC power-flow equations and operational limits. The bus voltages may be represented in rectangular coordinates as $V_i = V_{r,i} + \mathrm{j}V_{i,i}$, with $V_{r,i} = |V_i| \cos\theta_i$ and $V_{i,i} = |V_i| \sin\theta_i$ or equivalently, in polar coordinates as $V_i = |V_i| e^{\mathrm{j}\theta_i}$ (Grainger et al. (2016)). In this paper, we use the rectangular representation. The input and decision variables are

$$x = \begin{bmatrix} P_d \\ Q_d \end{bmatrix}, \qquad y = \begin{bmatrix} P_g \\ Q_g \\ V_r \\ V_i \end{bmatrix}, \tag{35}$$

where $P_d, Q_d \in \mathbb{R}^{n_b}$ denote the active and reactive demands, $P_g, Q_g \in \mathbb{R}^{n_g}$ denote the generator active and reactive powers, and $V = V_r + \mathrm{j}V_i \in \mathbb{C}^{n_b}$ is the complex bus-voltage vector.

Let $Y_{\text{bus}} = G + \mathrm{j}B$ denote the bus-admittance matrix, which encodes the network topology and line admittances. The active and reactive power injections at bus b are then

$$\begin{aligned} P_b^{\text{inj}} &= V_{r,b} \sum_{j \in \mathcal{N}(b) \cup \{b\}} (G_{bj} V_{r,j} - B_{bj} V_{i,j}) + V_{i,b} \sum_{j \in \mathcal{N}(b) \cup \{b\}} (B_{bj} V_{r,j} + G_{bj} V_{i,j}), \\ Q_b^{\text{inj}} &= V_{i,b} \sum_{j \in \mathcal{N}(b) \cup \{b\}} (G_{bj} V_{r,j} - B_{bj} V_{i,j}) - V_{r,b} \sum_{j \in \mathcal{N}(b) \cup \{b\}} (B_{bj} V_{r,j} + G_{bj} V_{i,j}). \end{aligned} \tag{36}$$

These expressions define the nonlinear AC power-balance constraints used in our formulation.

The rectangular AC-OPF problem can then be written as

$$\begin{aligned} \min_{P_g, Q_g, V_r, V_i} \quad & \sum_{g=1}^{n_g} C_g(S_{\text{base}} P_g) \\ \text{s.t.} \quad & \sum_{g \in \mathcal{G}(b)} P_g - P_{d,b} - P_b^{\text{inj}}(V_r, V_i) = 0, && b \in \mathcal{B}, \\ & \sum_{g \in \mathcal{G}(b)} Q_g - Q_{d,b} - Q_b^{\text{inj}}(V_r, V_i) = 0, && b \in \mathcal{B}, \\ & \underline{P}_g \le P_g \le \overline{P}_g, && g = 1, \dots, n_g, \\ & \underline{Q}_g \le Q_g \le \overline{Q}_g, && g = 1, \dots, n_g, \\ & \underline{V}_b^2 \le V_{r,b}^2 + V_{i,b}^2 \le \overline{V}_b^2, && b \in \mathcal{B}, \\ & |S_\ell^f(V_r, V_i)| \le \overline{S}_\ell, && \ell \in \mathcal{L}, \\ & |S_\ell^t(V_r, V_i)| \le \overline{S}_\ell, && \ell \in \mathcal{L}, \\ & V_{i,b_{\text{ref}}} = 0. \end{aligned} \tag{37}$$

Here, $\mathcal{G}(b)$ denotes the set of generators connected to bus b, $C_g(\cdot)$ is the production cost of generator g, and S_{base} is the system base apparent power used to convert per-unit generator powers to physical units for evaluating the cost. The quantities S_ℓ^f and S_ℓ^t denote the complex power flows at the two ends of line ℓ, and $\overline{S}_\ell$ is its apparent-power limit. The first two constraints enforce active and reactive power balance at each bus, while the remaining constraints impose generator, voltage-magnitude, branch-flow, and reference-angle limits.