# EG-Opt: Efficient Learning for Nonlinear Programming via Explicit Gradients from Sample-Calibrated Linear Surrogates

**Anonymous authors**
Paper under double-blind review

## Abstract

Large-scale nonlinear programs remain challenging, especially in repeated applications. Learning-based methods enable fast inference. However, feasibility-preserving training often requires repeated nonlinear solutions or iterative projections, while implicit differentiation or unrolling adds substantial computational overhead. We propose EG-Opt[1], an efficient framework based on explicit gradients from sample-calibrated linear surrogates. During training, EG-Opt replaces repeated nonlinear completion with a reusable linear surrogate enabling explicit gradient propagation and efficient batching, caching, and factorization reuse. In addition, we derive local error bounds for state and gradient, and provide sufficient conditions for surrogate gradients to be descent directions for the exact completion loss. At inference, the linear surrogate warm-starts the original nonlinear completion solver, reducing iterations and helping recover the reference solution branch associated with the labels when nonlinear completion admits multiple roots, while ensuring equality feasibility. Across ten benchmarks spanning alternate-current optimal power flow, water distribution, gas flow optimization, and a controlled nonlinear problem, EG-Opt achieves full equality and inequality feasibility with high solution quality. It speeds up end-to-end training by up to 128× versus implicit differentiation and 2,160× versus two-point zeroth-order training while accelerating inference by approximately 1.5× over nonlinear-completion baselines.

## 1 Introduction

A variety of scientific problems and engineering applications require repeated solutions of parametric nonlinear programs (NLPs). Conventional solvers are reliable but computationally costly at scale. Learning methods offer fast inference but struggle to ensure feasibility. Strategies that preserve feasibility typically embed nonlinear completion, projection, or iterative correction in training and differentiate through them implicitly or by unrolling. As predictions change, nonlinear states and their Jacobians must be recomputed. This computational cost grows with problem size, training updates, and numerical accuracy, often becoming the main bottleneck. Faster training allows us to tackle larger problems and evaluate more model configurations within a fixed computational budget. It also enables quicker adaptation to changing operating conditions and data.

We propose an efficient learning framework based on sample calibration and explicit gradients (EG-Opt) for optimization with hard nonlinear equality constraints. The framework uses two forms of a linear surrogate with shared frozen structure and parameters: a sample-calibrated linear surrogate for training and a label-independent surrogate for inference. They differ only in calibration, which ensures that linear state recovery reproduces the reference state at each labeled point. During training, the sample-calibrated linear surrogate recovers state variables from predicted control variables. State computation and explicit gradient propagation follow the same computational graph, avoiding repeated nonlinear completion, implicit differentiation, and unrolling. At inference, the label-independent surrogate initializes nonlinear completion to recover equality feasibility. This can reduce iterations, accelerate inference, and help recover the reference solution branch associated with the labels when multiple roots exist. The main contributions are as follows:

[1]Code: `https://anonymous.4open.science/r/EG-Opt-C46C`.



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.

1. **A linear surrogate framework for training and inference.** We propose a linear surrogate framework combining sample calibration and explicit gradients to accelerate training and inference. The design retains nonlinear equality recovery and helps recover the reference solution branch when nonlinear completion admits multiple roots.
2. **Explicit surrogate gradients and local validity analysis.** We derive an explicit gradient chain for the calibrated linear surrogate loss, along with bounds on state and gradient errors relative to exact nonlinear completion. These bounds provide local sufficient conditions for linear surrogate gradients to guide descent of the exact completion loss.
3. **Validation across domains on nonconvex NLPs.** We evaluate EG-Opt on alternate-current optimal power flow, water distribution optimization, and optimal gas flow, which involve distinct nonlinear equalities. Across domains and scales, EG-Opt achieves $100\%$ equality and inequality feasibility with competitive objective quality. It accelerates end-to-end training by up to $128\times$ over implicit differentiation and $2{,}160\times$ over two-point zeroth-order training, and inference by approximately $1.5\times$ over nonlinear-completion baselines.

## 2 Related Work

Our work is closely related to research on supervised prediction of optimization decisions, constraint handling in learning models, and acceleration of differentiable training. We discuss their main ideas, limitations, and connections to EG-Opt.

**Fast decision prediction via supervised learning.** For repeated parametric optimization, networks learn mappings from parameters to decisions using solver reference solutions. Applications span nonconvex nonlinear programs (Sun et al., 2018; Zamzam & Baker, 2020), linearly constrained quadratic programs (Pan et al., 2021), and finite horizon optimization with dynamical equalities (Hertneck et al., 2018). This moves computation offline, enabling fast candidate decisions. However, the prediction error, objective suboptimality, and constraint violation are distinct, and accurate fitting does not ensure feasibility. EG-Opt likewise uses reference solutions but couples learning with the original problem structure rather than equating label accuracy with deployment feasibility.

**Constraint handling in learning-based methods.** Learning for constrained optimization must address both objective quality and feasibility. Bounded mappings and softmax can encode interval, positivity, or simplex constraints, whereas general equations require constraint information in the training objective or prediction process. Physics-informed neural networks (Raissi et al., 2019) penalize equation residuals, but finite penalties cannot guarantee strict feasibility. Equality completion instead predicts some variables and solves the original equations for the rest. Zamzam & Baker (2020) use this at inference. DC3 (Donti et al., 2021) incorporates equality completion and inequality correction into training, propagating loss gradients through complete decisions. ENFORCE (Lastrucci & Schweidtmann, 2025), KKT-Hardnet (Iftakher et al., 2026), FSNet (Nguyen & Donti, 2025), and HardNet++ (Goertzen et al., 2026) use projection or iterative feasibility restoration. Repeated nonlinear solutions and corrections as predictions change can, however, bottleneck large-scale training. EG-Opt incorporates linear surrogates into both training and inference, avoiding repeated nonlinear solves during training while retaining equality recovery at inference.

**Gradient computation in end-to-end learning.** Training through optimization or constraint recovery requires propagating downstream gradients to the model (Donti et al., 2017). Differentiable optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019; Pineda et al., 2022), solver differentiation frameworks (Blondel et al., 2022; Gould et al., 2022), and implicit state models (Bai et al., 2019) enable this but can be costly. Implicit differentiation solves state-dependent linear systems, unrolling (Domke, 2012) backpropagates through solver trajectories, and zeroth order estimation (Pan et al., 2023) requires perturbed solves and loss evaluations. These costs recur as parameters change. To reduce the backward cost, Jacobian free backpropagation (JFB) (Fung et al., 2022) and phantom gradients (Geng et al., 2021) provide cheaper derivatives. However, both retain forward solutions: JFB and phantom gradients require implicit states. When forward solves dominate, approximate backpropagation cannot eliminate the repeated solve bottleneck. These studies show that exact gradients are not the only option: what matters is whether inexpensive approximations support effective updates. EG-Opt instead computes forward states and explicit gradients using the sample-calibrated linear surrogate, replacing nonlinear completion and implicit differentiation with reusable linear computations for efficient parameter updates.

## 3 EG-Opt: Explicit Gradient Training and Warm-Started Inference

For a given parameter vector $\boldsymbol{x}$, consider the following parametric NLP:

$$\min_{\boldsymbol{u},\boldsymbol{z}} F(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{h}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{g}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x}) \leq \boldsymbol{0}. \tag{1}$$

Here, the decision variables are partitioned into control variables $\boldsymbol{u} \in \mathbb{R}^p$ and state variables $\boldsymbol{z} \in \mathbb{R}^m$. $F(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x})$ is the objective function, while $\boldsymbol{h}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x}) = \boldsymbol{0}$ and $\boldsymbol{g}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x}) \leq \boldsymbol{0}$ define the equality and inequality constraints, respectively. The problem can be either convex or nonconvex.

A naive direct learning approach uses a neural network $N_\theta$ with trainable parameters $\theta$ to predict the outputs $(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{z}}) = N_\theta(\boldsymbol{x})$ and combines the objective and constraint violations in a soft penalty loss, with nonnegative weights $\lambda_h, \lambda_g$ for equality and inequality violations, respectively:

$$\ell_{\text{soft}}(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{z}};\boldsymbol{x}) = F(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{z}};\boldsymbol{x}) + \lambda_h \|\boldsymbol{h}(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{z}};\boldsymbol{x})\|^2 + \lambda_g \|[\boldsymbol{g}(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{z}};\boldsymbol{x})]_+\|^2. \tag{2}$$

Here, $[\cdot]_+$ denotes the componentwise positive part. A small soft-penalty loss does not ensure feasibility with finite penalty weights. To improve feasibility, DC3 (Donti et al., 2021) predicts some variables, recovers the rest through equality completion, and iteratively corrects inequality violations. However, changing predictions require repeated completion and correction, with gradients propagated through implicit differentiation and unrolling, respectively, making training costly.

EG-Opt addresses this bottleneck with an efficient linear surrogate framework for parametric NLPs. It predicts only the controls $\boldsymbol{u}$ and recovers the states $\boldsymbol{z}$ through equality completion, reducing the output dimension while preserving the original equality constraints. The linear surrogate accelerates both training and inference through two complementary designs:

**Efficient training with a sample-calibrated linear surrogate.** Training labels provide sample-specific calibration terms, allowing the linear surrogate to reproduce state variable labels at each labeled point. The sample-calibrated linear surrogate replaces nonlinear completion with an inexpensive linear solution and enables explicit gradient propagation consistent with the forward pass, avoiding repeated nonlinear solutions, implicit differentiation, and unrolling during training. Moreover, the fixed surrogate matrix and its factorization can be cached and reused in batched PyTorch computations, reducing synchronization and solver control flow overhead.

**Warm-started inference with a label-independent linear surrogate.** At inference, the label-independent surrogate provides an efficient initialization for the nonlinear completion solver, reducing the iterations required to recover equality feasible state variables. Thus, sample calibration enables efficient state recovery and explicit gradient propagation during training, while the label-independent surrogate accelerates nonlinear completion at inference.

Figure 1 illustrates the workflow, and Algorithm 1 summarizes offline preparation, training, and inference. The workflow begins with solvers generating labeled data $(\boldsymbol{x}_i, \boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star)$ within a predefined instance range, while a linear surrogate is constructed from the equality structure and frozen. Labels and the linear surrogate yield sample calibration terms. During training, the model predicts controls $\boldsymbol{u}$ with sigmoid output parameterization enforcing their bounds. The sample-calibrated linear surrogate computes state variables $\boldsymbol{z}$, and the task loss is backpropagated via explicit surrogate gradients. At inference, the trained model predicts $\hat{\boldsymbol{u}}$. Without using labels, the label-independent surrogate provides a warm start for nonlinear completion to recover the corresponding $\hat{\boldsymbol{z}}$.

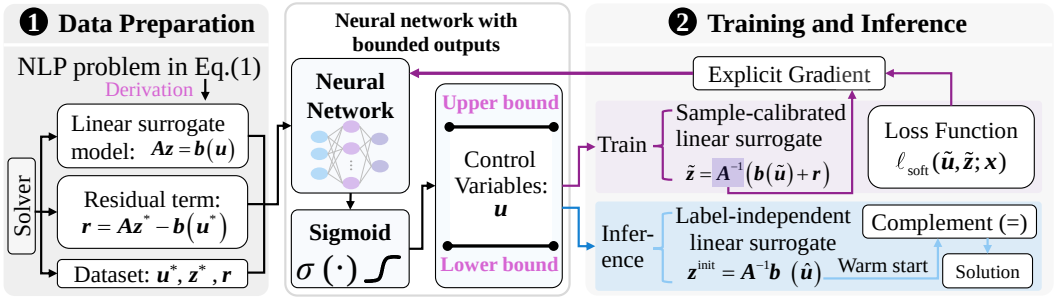



Figure 1: The EG-Opt framework

**Algorithm 1** EG-Opt: offline preparation, training, and warm-started inference

**Require:** Loss and weights, labels $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star)\}$
Construct a label-independent surrogate $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ and freeze it. Cache $\boldsymbol{A}$'s factorization
Reuse $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ with the same numerical settings during training and inference.
For each sample $i$, set $\boldsymbol{b}_i(\boldsymbol{u}) = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x}_i)$, and cache the residual $\boldsymbol{r}_i = \boldsymbol{A}\boldsymbol{z}_i^\star - \boldsymbol{b}_i(\boldsymbol{u}_i^\star)$
**procedure** TRAIN($\mathcal{D}$)
    Initialize model parameters $\boldsymbol{\theta}$
    **while** the fixed training update budget is not exhausted **do**
        Draw a batch $\mathcal{B}$ and set $\boldsymbol{u}_{\theta,i} = \boldsymbol{u}_{\min} + (\boldsymbol{u}_{\max} - \boldsymbol{u}_{\min}) \odot \mathrm{sigmoid}(N_\theta(\boldsymbol{x}_i))$, $i \in \mathcal{B}$
        Reuse factorizations to solve $\boldsymbol{A}\tilde{\boldsymbol{z}}_i = \boldsymbol{b}_i(\boldsymbol{u}_{\theta,i}) + \boldsymbol{r}_i$
        Evaluate the training loss on batch $\mathcal{B}$
        Backpropagate explicit gradients through the state and control paths to update $\boldsymbol{\theta}$
    **end while**
    **return** $\boldsymbol{\theta}$
**end procedure**
**procedure** INFER($\boldsymbol{x}, \boldsymbol{\theta}$)
    Use the trained model $N_\theta$ with the bounded output map on $\boldsymbol{x}$ to obtain $\hat{\boldsymbol{u}}$
    Compute the initial state $\boldsymbol{z}^{\text{init}}$ by solving $\boldsymbol{A}\boldsymbol{z}^{\text{init}} = \boldsymbol{b}(\hat{\boldsymbol{u}}; \boldsymbol{x})$ without calibration
    Given the predicted $\hat{\boldsymbol{u}}$, solve the original nonlinear equalities from $\boldsymbol{z}^{\text{init}}$ to obtain $\hat{\boldsymbol{z}}$
    **return** $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{z}})$ if completion succeeds, or a failure flag otherwise
**end procedure**

### 3.1 Linear Surrogate Construction, Training, and Inference

**Linear surrogate construction.** Learning with equality completion predicts the controls $\boldsymbol{u}$ and recovers the states $\boldsymbol{z}$ by solving $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}) = \boldsymbol{0}$. Let $\boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x})$ denote the selected regular solution branch. Each forward pass solves these nonlinear equalities, while backpropagation evaluates the constraint Jacobians $\boldsymbol{J}_{h,u} := \partial \boldsymbol{h}/\partial \boldsymbol{u}$ and $\boldsymbol{J}_{h,z} := \partial \boldsymbol{h}/\partial \boldsymbol{z}$ at the completed state $(\boldsymbol{u}, \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}))$. These matrices determine the completion sensitivity $\boldsymbol{J}_{z_h,u} := \partial \boldsymbol{z}_h/\partial \boldsymbol{u} = -\boldsymbol{J}_{h,z}^{-1}\boldsymbol{J}_{h,u}$. As predictions change during training, the nonlinear solutions and associated linear or adjoint sensitivity solutions must be repeated, creating a computational bottleneck. EG-Opt instead preconstructs a label-independent surrogate $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$, with a shared invertible matrix $\boldsymbol{A}$ and an explicit function $\boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$. It reuses the factorization for linear solutions and propagates explicit gradients through the same graph. This matrix can be constructed by first-order expansion. In fixed dimensionless coordinates (Appendix A), choose a feasible anchor $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{z}})$ on the reference solution branch associated with a labeled point, and denote the Jacobians there by $\bar{\boldsymbol{J}}_{h,u}$ and $\bar{\boldsymbol{J}}_{h,z}$. Expanding gives

$$\begin{aligned} &\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \bar{\boldsymbol{x}}) \approx \boldsymbol{h}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{z}}; \bar{\boldsymbol{x}}) + \bar{\boldsymbol{J}}_{h,u}(\boldsymbol{u} - \bar{\boldsymbol{u}}) + \bar{\boldsymbol{J}}_{h,z}(\boldsymbol{z} - \bar{\boldsymbol{z}}), \\ &\boldsymbol{A} = \bar{\boldsymbol{J}}_{h,z}, \quad \boldsymbol{b}(\boldsymbol{u}; \bar{\boldsymbol{x}}) = \boldsymbol{A}\bar{\boldsymbol{z}} - \boldsymbol{h}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{z}}; \bar{\boldsymbol{x}}) - \bar{\boldsymbol{J}}_{h,u}(\boldsymbol{u} - \bar{\boldsymbol{u}}). \end{aligned} \tag{3}$$

For varying instances, $\boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ retains parameter dependence while $\boldsymbol{A}$ remains fixed. Thus, $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ is linear only in $\boldsymbol{z}$, and $\boldsymbol{b}$ may be nonlinear in $(\boldsymbol{x}, \boldsymbol{u})$. Before training, we freeze $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ and factorize $\boldsymbol{A}$. This factorization aims to decouple the recovery of state variables from the prediction of control variables, obtain explicit gradients, and reduce inversion costs with smaller matrices (Appendix C details the factorization procedure). Those matrices are cached as constants on the GPU and reused throughout training and inference. With data and intermediate tensors kept on the device, forward and backward passes use batched linear solves and matrix operations, reducing matrix assembly, Python control overhead, and synchronization between the CPU and GPU (complexity: Appendix A.7). Appendix B discusses optional extensions to multiple linear surrogates.

**Sample-calibrated linear surrogate and explicit-gradient training.** Sample calibration removes the state bias of $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ at labeled points. For $(\boldsymbol{x}_i, \boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star)$, let $\boldsymbol{b}_i(\boldsymbol{u}) = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x}_i)$. The matrix $\boldsymbol{A}$ is shared by all samples. Define the calibration term and sample-calibrated linear surrogate as

$$\boldsymbol{r}_i = \boldsymbol{A}\boldsymbol{z}_i^\star - \boldsymbol{b}_i(\boldsymbol{u}_i^\star), \qquad \tilde{\boldsymbol{z}}_i(\boldsymbol{u}) = \boldsymbol{A}^{-1}[\boldsymbol{b}_i(\boldsymbol{u}) + \boldsymbol{r}_i]\,. \tag{4}$$

$\boldsymbol{r}_i$ is computed once before training and cached on the GPU as a constant. It ensures $\tilde{\boldsymbol{z}}_i(\boldsymbol{u}_i^\star) = \boldsymbol{z}_i^\star$ without changing sensitivity. The parameterization $\boldsymbol{u}_{\theta,i} = \boldsymbol{u}_{\min} + (\boldsymbol{u}_{\max} - \boldsymbol{u}_{\min}) \odot \mathrm{sigmoid}(N_\theta(\boldsymbol{x}_i))$ enforces control bounds. A basic training loss is

$$\mathcal{L}(\boldsymbol{\theta}) = \textstyle\sum_{i=1}^N \big[F(\boldsymbol{u}_{\theta,i}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}_{\theta,i}); \boldsymbol{x}_i) + \lambda_g \|[\boldsymbol{g}(\boldsymbol{u}_{\theta,i}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}_{\theta,i}); \boldsymbol{x}_i)]_+\|^2 + \lambda_s \|\boldsymbol{u}_{\theta,i} - \boldsymbol{u}_i^\star\|^2\big]/N. \tag{5}$$

$\lambda_g\|[\boldsymbol{g}]_+\|^2$ denotes the inequality penalty, and $\lambda_s\|\boldsymbol{u}_{\theta,i}-\boldsymbol{u}_i^\star\|^2$ supervises controls against reference labels. $\lambda_g$ and $\lambda_s$ are loss weights. Experimental losses are detailed in Appendix D.2.

Let $\ell_i(\boldsymbol{u},\boldsymbol{z})$ denote the sum of the three per-sample terms in Eq. (5), and let $\boldsymbol{J}_{\theta,i}=\partial\boldsymbol{u}_{\theta,i}/\partial\boldsymbol{\theta}$. Applying the chain rule and substituting $\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})=\boldsymbol{A}^{-1}\boldsymbol{J}_{b_i,u}(\boldsymbol{u})$ gives

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})=\textstyle\sum_{i=1}^{N}\boldsymbol{J}_{\theta,i}^\top\left[\nabla_{\boldsymbol{u}}\ell_i+\boldsymbol{J}_{b_i,u}(\boldsymbol{u}_{\theta,i})^\top\boldsymbol{A}^{-\top}\nabla_{\boldsymbol{z}}\ell_i\right]/N. \tag{6}$$

In Eq. (6), $\nabla_{\boldsymbol{u}}\ell_i$ represents the direct effect of the control variables on the loss. The second term accounts for their indirect effect through the sample-calibrated linear surrogate state. Both partial gradients are evaluated at $(\boldsymbol{u}_{\theta,i},\tilde{\boldsymbol{z}}_i(\boldsymbol{u}_{\theta,i}))$. Equation (6) thus allows direct computation of $\nabla_{\boldsymbol{u}}\ell_i+\boldsymbol{J}_{b_i,u}(\boldsymbol{u}_{\theta,i})^\top\boldsymbol{A}^{-\top}\nabla_{\boldsymbol{z}}\ell_i$ and backpropagation through $\boldsymbol{u}_{\theta,i}$, yielding the explicit gradient without explicitly forming $\partial\tilde{\boldsymbol{z}}_i/\partial\boldsymbol{\theta}$, differentiating through nonlinear completion, or unrolling solver iterations.

**Warm-started inference with a label-independent linear surrogate.** Inference retains $\boldsymbol{A}$ and $\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})$ frozen before training, removing only sample calibration. Given a test instance $\boldsymbol{x}$ and bounded prediction $\hat{\boldsymbol{u}}$, the label-independent surrogate initializes nonlinear completion. We denote by $\mathrm{Root}_h(\boldsymbol{u};\boldsymbol{x},\boldsymbol{z}^{\text{init}})$ a solution of $\boldsymbol{h}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x})=\boldsymbol{0}$ obtained from the initial state $\boldsymbol{z}^{\text{init}}$:

$$\boldsymbol{z}^{\text{init}}=\boldsymbol{A}^{-1}\boldsymbol{b}(\hat{\boldsymbol{u}};\boldsymbol{x}),\qquad\hat{\boldsymbol{z}}=\mathrm{Root}_h(\hat{\boldsymbol{u}};\boldsymbol{x},\boldsymbol{z}^{\text{init}}). \tag{7}$$

At inference, a single linear solve can provide a good initial guess for nonlinear completion without labels or calibration terms, reducing nonlinear iterations and improving inference efficiency. Moreover, when nonlinear completion admits multiple solutions, $\boldsymbol{z}^{\text{init}}$ can help guide the solver toward the reference solution branch associated with the labels, as demonstrated in Section 4.4. Convergence and cost analyses are given in Appendices A.6 and A.7.

### 3.2 Validity of surrogate gradients

Error bounds of state and gradient characterize when sample-calibrated gradients support effective training. Assume $\boldsymbol{J}_{h,z}$ and $\boldsymbol{A}$ are invertible and the nonlinear completion and linear surrogate maps are continuously differentiable. Their derivatives and the task loss partial derivatives are assumed locally Lipschitz.

Define label error $e_{0,i}=\|\boldsymbol{z}_i^\star-\boldsymbol{z}_h(\boldsymbol{u}_i^\star;\boldsymbol{x}_i)\|$, prediction displacement $d_i=\|\boldsymbol{u}-\boldsymbol{u}_i^\star\|$, and sensitivity mismatch at the label $\delta_{J,i}=\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}_i^\star)-\boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star;\boldsymbol{x}_i)\|$. Interpolation at the labeled point and Taylor expansion give $\|\tilde{\boldsymbol{z}}_i(\boldsymbol{u})-\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i)\|\le e_{0,i}+\delta_{J,i}d_i+C_id_i^2$, where $C_i$ measures the local curvature. Calibration removes zeroth-order bias relative to labels. The remaining state error depends on label error, sensitivity mismatch, and prediction displacement. Let $\boldsymbol{g}_{\mathrm{e},i}$ and $\boldsymbol{g}_{\mathrm{c},i}$ denote the gradients with respect to $\boldsymbol{u}$ of $\ell_i(\boldsymbol{u},\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i))$ and $\ell_i(\boldsymbol{u},\tilde{\boldsymbol{z}}_i(\boldsymbol{u}))$, respectively. For some $C_{z,i},C_{J,i}>0$,

$$\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|\le C_{z,i}\|\tilde{\boldsymbol{z}}_i(\boldsymbol{u})-\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i)\|+C_{J,i}\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})-\boldsymbol{J}_{z_h,u}(\boldsymbol{u};\boldsymbol{x}_i)\|. \tag{8}$$

Under these local conditions, let $\boldsymbol{G}_{\mathrm{c}}$ and $\boldsymbol{G}_{\mathrm{e}}$ denote linear surrogate and exact-completion parameter gradients at the same parameters on the same batch. If state and sensitivity errors propagated through network Jacobians and aggregated satisfy $\|\boldsymbol{G}_{\mathrm{c}}-\boldsymbol{G}_{\mathrm{e}}\|<\|\boldsymbol{G}_{\mathrm{e}}\|$, then $\boldsymbol{G}_{\mathrm{e}}^\top\boldsymbol{G}_{\mathrm{c}}>0$. Hence, $-\boldsymbol{G}_{\mathrm{c}}$ is a strict descent direction, decreasing the exact completion loss on that batch for a sufficiently small positive step. Full conditions, derivations of the error bounds, and the descent proof are given in Appendices A.1, A.2, and A.3.

## 4 Experiments

We assess EG-Opt's effectiveness across domains and scales. Gradient fidelity tests examine the quality of its training signals. Workload per update and time to target examine EG-Opt's training speedups. Tests with stronger nonlinearity, ill conditioning, and multiple roots probe its limits.

**Test cases.** We study AC optimal power flow (Case14/57/118), water distribution optimization (Net1/C-Town/Net6), and optimal gas flow (GasLib-11/135/4197). Their nonlinear equalities and couplings between control and state variables differ. Specifically, power networks involve voltage magnitude products and trigonometric functions of phase differences. In contrast, water networks combine resistance laws with fractional exponents and pressure boosts dependent on control variables, whereas gas networks use squared pressures as states and signed quadratic flow relations. Both

water and gas networks enforce flow conservation and use signed flows to represent transport direction, with resistance terms changing sign when flow reverses. Our Controlled Synthetic Constrained Problem (CSCP) varies nonlinearity, conditioning, and solution branches. Each case contains 15,000 instances (models/surrogates: Appendix C. Configurations/splits: Appendix D.1, Tables D1 and D2).

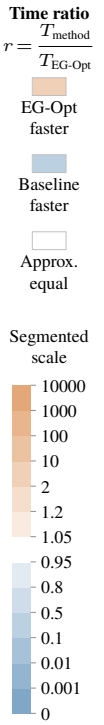
**Baselines.** Direct-NN (Sun et al., 2018) supervises all variables without completion or correction, whereas Partial-NN (Zamzam & Baker, 2020) learns control variables and recovers state variables through the original equalities at inference. Neither trains with nonlinear completion. Exact-ID, DC3, and ZO repeatedly complete original equalities in training: Exact-ID uses implicit gradients, DC3 (Donti et al., 2021) adds iterative inequality correction, and ZO (Pan et al., 2023) uses two-point differences with one direction per sample. They share labels, supervision terms, and loss weights with EG-Opt per case. All methods use the same training budget, three independent runs, and common evaluation and aggregation rules (Appendix D.2).

### 4.1 Cross-Domain and Multiscale Effectiveness

Table 1 averages three independent training seeds, with mean absolute relative objective gaps over the full evaluation set (Appendix D.2). End-to-end training includes offline preparation, and full training. Times are compared for the same case and timing scope (Appendix A.7).

**Table 1: Deployment quality and computation time across ten systems**

| Metric | Method | Power 14 | Power 57 | Power 118 | Gas 11 | Gas 135 | Gas 4197 | Water Net1 | Water C-Town | Water Net6 | Constructed CSCP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Objective gap (%) | **EG-Opt** | **0.0983** | **0.0706** | **0.0145** | **0.0859** | **0.1407** | **0.1124** | **0.2864** | **0.0624** | **0.1975** | **0.0704** |
| | Exact-ID | 0.0986 | 0.1057 | 0.0146 | 0.0868 | 0.1465 | 0.2058 | 0.2924 | 0.0640 | 0.1898 | 0.0704 |
| | DC3 | 0.1083 | 0.1061 | 0.0534 | 0.0708 | 6.3451 | 0.0597 | 0.2924 | 0.0640 | 0.1923 | 0.0704 |
| | ZO | 0.9325 | 1.2440 | 2.7064 | 0.0900 | 0.3850 | 0.1121 | 0.2924 | 0.0913 | 0.2429 | 0.0901 |
| | Direct-NN | 0.0154 | 0.0265 | 0.0274 | 0.0046 | 0.0129 | 0.0255 | 0.4529 | 0.0519 | 0.0263 | 0.0967 |
| | Partial-NN | $< 0.0001$ | 0.0002 | $< 0.0001$ | 0.0194 | 0.0258 | 0.0378 | 0.2998 | 0.0093 | 0.0371 | 0.0704 |
| Eq. feasibility (%) | **EG-Opt** | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** |
| | Exact-ID | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | DC3 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | ZO | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | Direct-NN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Partial-NN | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Ineq. feasibility (%) | **EG-Opt** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| | Exact-ID | 100.0 | 99.9 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | DC3 | 100.0 | 100.0 | 96.2 | 100.0 | 99.7 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | ZO | 100.0 | 80.0 | 26.7 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Direct-NN | 48.2 | 22.7 | 0.0 | 100.0 | 100.0 | 100.0 | 100.0 | 98.2 | 0.0 | 100.0 |
| | Partial-NN | 37.0 | 11.9 | 0.0 | 58.7 | 40.8 | 98.8 | 100.0 | 98.2 | 90.1 | 100.0 |
| E2E training (s) | **EG-Opt** | **227.7** | **236.1** | **238.8** | **308.2** | **422.6** | **870.3** | **94.1** | **246.0** | **810.4** | **138.4** |
| | Exact-ID | 16587 | 26538 | 30594 | 1588 | 4035 | 36212 | 632.5 | 5747 | 53168 | 174.3 |
| | DC3 | 16327 | 27321 | 31629 | 1676 | 4135 | 36610 | 654.9 | 5895 | 54783 | 174.4 |
| | ZO | 36045 | 59570 | 67248 | 8007 | 20419 | 350437 | 1404 | 46846 | 1750819 | 387.6 |
| | Direct-NN | 100.7 | 102.4 | 109.8 | 186.5 | 232.7 | 513.6 | 59.6 | 97.8 | 395.1 | 139.0 |
| | Partial-NN | 93.2 | 92.3 | 96.9 | 183.2 | 228.7 | 521.2 | 52.6 | 62.8 | 105.7 | 126.6 |
| E2E inference (ms/instance) | **EG-Opt** | **2.8179** | **3.1341** | **3.5844** | **0.2166** | **0.3171** | **4.3847** | **0.2319** | **2.9451** | **5.3428** | **0.0157** |
| | Exact-ID | 3.7657 | 3.6758 | 4.2505 | 0.2937 | 0.7201 | 9.9008 | 0.3004 | 4.4108 | 5.7649 | 0.0167 |
| | DC3 | 3.8077 | 3.8275 | 4.2136 | 3.0425 | 4.9979 | 16.2330 | 0.3178 | 4.4846 | 5.9887 | 0.0168 |
| | ZO | 3.7894 | 3.7256 | 4.2347 | 0.2958 | 0.7187 | 9.8818 | 0.2979 | 4.4349 | 5.7549 | 0.0171 |
| | Direct-NN | 0.0066 | 0.0065 | 0.0081 | 0.0064 | 0.0076 | 0.0319 | 0.0028 | 0.0125 | 0.0506 | 0.0117 |
| | Partial-NN | 3.0590 | 3.7356 | 4.2599 | 0.2923 | 0.7091 | 9.9217 | 0.2992 | 4.4754 | 5.7410 | 0.0177 |




In Table 1, EG-Opt attains full equality and inequality feasibility on all ten cases while maintaining competitive objective quality. Its mean absolute relative objective gap matches or improves on Exact-ID in nine cases, showing that the computational savings need not compromise objective quality. On GasLib-4197, DC3 achieves a smaller objective gap than EG-Opt (0.0597% versus 0.1124%), but its inference procedure includes additional iterative constraint correction and takes 3.70 times as long (16.23 versus 4.38 ms per instance). This additional inference cost can be a drawback in applications requiring rapid responses. Direct-NN and Partial-NN attain smaller objective gaps in several cases but do not consistently satisfy the constraints, so objective gaps alone do not establish superior solution quality. EG-Opt's end-to-end training and inference are both faster than Exact-ID, DC3, and ZO. Training speedups reach approximately 128× over Exact-ID and 2,160× over ZO, while inference is approximately 1.5× faster than the three baselines on most cases. Supplementary experiments in Appendix D.2.1 further show that preparation, including construction of the sample-calibrated linear surrogate, contributes little to EG-Opt's end-to-end training cost. Appendix D.2.2 tests a variant that retains nonlinear completion but uses explicit gradients from the sample-calibrated

linear surrogate. This variant achieves comparable solution quality but requires substantially more training time than EG-Opt, supporting the complementary roles of explicit gradients and linear state computations in training acceleration.

### 4.2 Gradient Fidelity and Descent Effectiveness

We assess the accuracy of gradients from the sample-calibrated linear surrogate and their ability to reduce the task loss. The sample calibration ablation is reported in Appendix D.5.

We use the high accuracy implicit gradients from Exact-ID as the reference. For a variable block $\boldsymbol{v} \in \{\boldsymbol{u}, \boldsymbol{\theta}\}$, let $\boldsymbol{g}_{\boldsymbol{v}}$ and $\boldsymbol{g}^{\star}_{\boldsymbol{v}}$ denote the evaluated and reference gradients, respectively. We define the angular difference $\iota_{\boldsymbol{v}}$ and relative gradient error $R_{\boldsymbol{v}}$ as

$$\iota_{\boldsymbol{v}} = \arccos\left(\boldsymbol{g}_{\boldsymbol{v}}^{\top}\boldsymbol{g}^{\star}_{\boldsymbol{v}} / \left[\|\boldsymbol{g}_{\boldsymbol{v}}\|_2 \|\boldsymbol{g}^{\star}_{\boldsymbol{v}}\|_2\right]\right)(180/\pi), \qquad R_{\boldsymbol{v}} = \|\boldsymbol{g}_{\boldsymbol{v}} - \boldsymbol{g}^{\star}_{\boldsymbol{v}}\|_2 / \|\boldsymbol{g}^{\star}_{\boldsymbol{v}}\|_2. \tag{9}$$

For each test case in Figure 2, all methods are evaluated using the same checkpoint, 64 validation samples, and task loss. They take steps of equal size along unit gradient directions in the same dimensionless control space, and loss is evaluated after the same bound projection and exact completion. ZO uses one random direction and two perturbed completion queries per sample. Its computational cost is reported in Section 4.3.

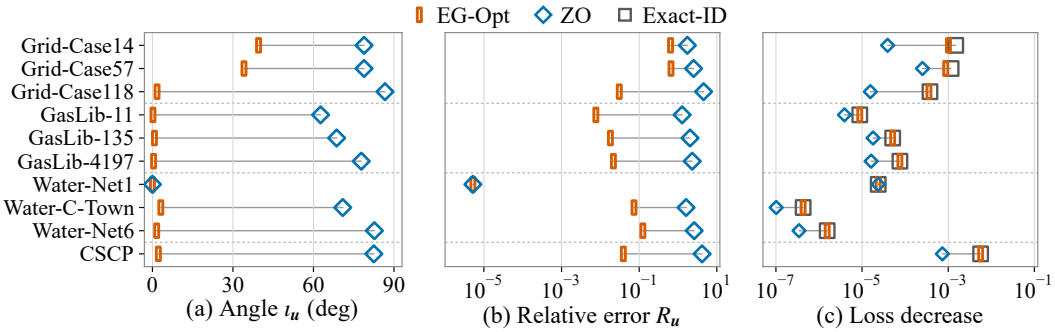



Figure 2: Gradient fidelity and one-step loss decrease across ten test cases

Figure 2 shows that EG-Opt achieves smaller angular differences and lower median relative gradient errors than ZO on all nine multidimensional test cases. At the same step size, EG-Opt also achieves a larger relative reduction in task loss than ZO. The Water-Net1 case has only one control dimension and is an exception: a single direction spans the control space, making ZO's central difference estimate nearly exact. These results indicate that in the tested multidimensional cases, EG-Opt provides training signals closer to the exact gradients and achieves greater loss reduction in a single step.

### 4.3 Why Training Is Faster: Explicit Gradients and Linear Solves

This section analyzes the sources of training speedups and the efficiency of meeting the target criteria using workload per update and time to target.

**Computational Workload per Update.** Table 2 lists state completion and implicit gradient counts for full batches across nine engineering cases. Figure 3 reports runtime per update and speedups using matched training phase states and batches.

Table 2: State completion and implicit gradient counts per full batch across nine engineering cases

| | | Power | | | Gas | | | Water | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Work | Method | 14 | 57 | 118 | 11 | 135 | 4197 | Net1 | C-Town | Net6 |
| State completion | **EG-Opt** (linear) | **256** | **256** | **256** | **200** | **200** | **200** | **64** | **64** | **64** |
| | **EG-Opt** (nonlinear) | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | Exact-ID (nonlinear) | 256 | 256 | 256 | 200 | 200 | 200 | 64 | 64 | 64 |
| | DC3 (nonlinear) | 256–512 | 256–512 | 256–512 | 200–400 | 200–400 | 200–400 | 64–128 | 64–128 | 64–128 |
| | ZO (nonlinear) | 768 | 768 | 768 | 600 | 600 | 600 | 192 | 192 | 192 |
| Implicit gradient | **EG-Opt** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | Exact-ID | 256 | 256 | 256 | 200 | 200 | 200 | 64 | 64 | 64 |
| | DC3 | 256–1792 | 256–1792 | 256–1792 | 200–1400 | 200–1400 | 200–1400 | 64–448 | 64–448 | 64–448 |
| | ZO | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

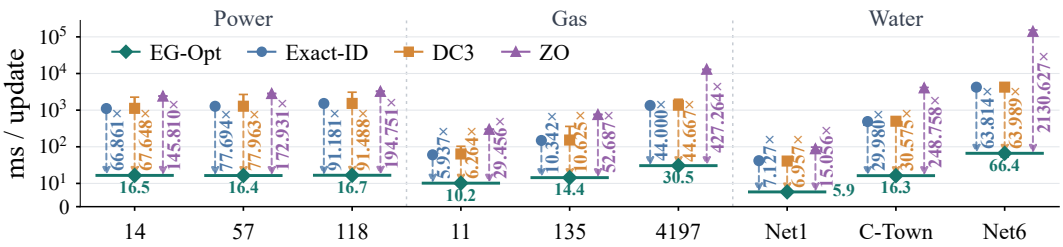



Figure 3: Computation counts and efficiency per update across nine engineering test cases

Table 2 and Figure 3 show that EG-Opt completes each update faster on all nine engineering test cases. Replacing the forward and backward passes with linear computations and reusing computations enabled by the fixed structure jointly reduce the overhead. Appendix D.2 further distinguishes the added benefit of device caching from the training gains on CSCP without GPU acceleration. The caching benefit is not equated with the full benefit of implementation optimizations.

**Time to Target.** All methods share the same data, evaluation frequency, hardware, and fixed training budget (Appendix D.2). A single checkpoint must attain 100% equality feasibility, at least 99% inequality feasibility, and a mean absolute relative objective gap of at most 1% over the full evaluation set, computed as in Table 1. Figure 4 reports the measured training update time to target (seconds), excluding validation and offline preparation, unlike the end-to-end times in Table 1 (Appendix A.7). *Not reached* denotes failure to meet these criteria throughout the full training run.

| | | EG-Opt | Exact-ID | DC3 | ZO | Direct-NN | Partial-NN |
|---|---|---|---|---|---|---|---|
| POWER | Grid-Case14 | 3.7 s | 202.0 s | 162.6 s | 5,856.7 s | Not reached | Not reached |
| | Grid-Case57 | 8.6 s | 382.2 s | 640.1 s | Not reached | Not reached | Not reached |
| | Grid-Case118 | 36.3 s | 3,329.9 s | 3,686.8 s | Not reached | Not reached | Not reached |
| GAS | GasLib-11 | 0.7 s | 3.9 s | 4.2 s | 20.0 s | Not reached | Not reached |
| | GasLib-135 | 1.0 s | 10.0 s | 10.3 s | 51.0 s | Not reached | Not reached |
| | GasLib-4197 | 2.1 s | 90.5 s | 91.5 s | 876.0 s | Not reached | 0.2 s |
| WATER | Water-Net1 | 0.7 s | 4.7 s | 5.0 s | 10.6 s | Not reached | 0.2 s |
| | Water-C-Town | 1.8 s | 28.9 s | 29.5 s | 234.6 s | Not reached | Not reached |
| | Water-Net6 | 10.0 s | 655.7 s | 812.5 s | 22,083.7 s | Not reached | 1.5 s |
| | CSCP | 3.8 s | 4.7 s | 4.7 s | 68.6 s | Not reached | 3.4 s |

Speedup $\frac{T_{\text{method}}}{T_{\text{EG-Opt}}}$ $10^3$ $10^2$ $10^1$ $10^0$ $10^{-1}$

Figure 4: Training update time to target ten test cases

Under the same protocol, the measured update times for Exact-ID and ZO are 1.24–91.81 and 15.39–2204.84 times those for EG-Opt, respectively. Thus, EG-Opt reduces both cost per update and the cumulative update time needed to reach the same deployment quality.

### 4.4 Gradient Reliability and Empirical Failure Modes

Tests on the CSCP probe EG-Opt's limits in gradient accuracy under stronger nonlinearity and matrix ill conditioning, descent under gradient errors, and linear warm start recovery of a reference branch when one control input admits multiple solutions. The settings are: (a) A nominal case with a single root and well conditioned exact completion Jacobian $\boldsymbol{J}_{h,z}$ and surrogate matrix $\boldsymbol{A}$ provides baseline gradient and warm start diagnostics. (b) Increasing nonlinear frequency while retaining a single root tests curvature effects on surrogate gradients and warm starts. (c) Varying nonlinear amplitude and reference state makes $\boldsymbol{J}_{h,z}$ nearly singular near the target, testing effects on gradient approximation and deployment. (d) Shifting only the linearization anchor makes $\boldsymbol{A}$ nearly singular while keeping $\boldsymbol{J}_{h,z}$ well conditioned, isolating its effects on gradients and warm starts. (e) Increasing nonlinear amplitude creates multiple roots with a prespecified left reference solution branch to evaluate loss reduction and solution branch recovery. (f) With exact completion kept well conditioned, joint sweeps over the minimum singular value $\mu_A$ of $\boldsymbol{A}$, which measures its proximity to singularity, and prediction perturbation $\delta$ quantify error amplification using the relative error $R_{\boldsymbol{u}}$ of surrogate control gradients. (g) Comparing normalized surrogate and exact gradients through one step descent success rates across prediction perturbations and surrogate conditioning tests whether surrogate directions remain effective away from the reference state. (h) Under multiple roots, varying the reference state's distance to the fold point tests recovery of the reference solution branch associated with the labeled

point using linear warm starts versus standard initialization. Tests (a–e) diagnose parameter gradients and deployment of models trained with the boundary study loss, comprising supervision, objective, and auxiliary equality terms rather than the base configuration in Eq. (5). Tests (f–g) involve no network training and instead analyze control gradients of the original objective and their descent effectiveness. Appendix C.4 gives the formulation, while Appendix D.3 details the model and sample settings, parameter grids, and generation procedures.

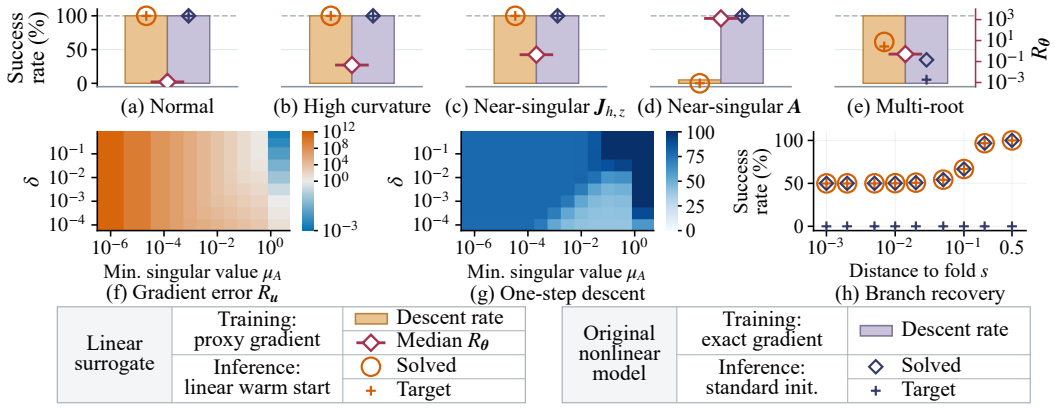



Figure 5: CSCP gradient and solution branch diagnostics

Figure 5(a–e) shows that under nominal and high curvature settings, surrogate and exact gradients remain close, and both directions effectively reduce the training loss. When the completion Jacobian becomes nearly singular, gradient errors increase, but some descent capability is retained. In contrast, a nearly singular surrogate matrix degrades both the parameter gradient approximation and linear warm starts. These results suggest that high curvature or ill conditioning of the exact completion Jacobian alone does not necessarily cause EG-Opt to fail, whereas severe ill conditioning of the frozen surrogate matrix $\boldsymbol{A}$ can substantially amplify approximation errors and constitutes a key failure mode observed in these experiments. Figures 5(f,g) further show that as the minimum singular value $\mu_A$ decreases, amplification through $\boldsymbol{A}^{-1}$ increases, leading to larger surrogate gradient errors relative to the exact gradient. Although normalized surrogate gradient directions still decrease the objective in some ill-conditioned settings, their descent success rate is lower than that of exact directions. Within the tested perturbation range, when $\boldsymbol{A}$ is well conditioned, larger perturbations move the exact gradient away from the near zero regime, reducing the relative surrogate gradient error and improving the one-step descent success. For smaller perturbations, the exact gradient approaches zero, making residual sensitivity errors more prominent in relative terms. Once $\boldsymbol{A}$ becomes severely ill conditioned, varying $\delta$ provides little recovery in gradient accuracy. Finally, Figure 5(e,h) shows that under the multiroot settings, linear warm starts recover the prescribed reference solution branch more reliably than standard diagonal initialization when the original equations and solver are held fixed and only the initialization is changed. Thus, the benefit of EG-Opt extends beyond completion success to solution branch selection.

## 5 Conclusion

EG-Opt accelerates training while maintaining competitive solution quality. Its warm start improves inference efficiency while recovering equality feasibility and helping the solver reach the reference solution branch associated with the labels when nonlinear completion admits multiple roots. Our findings suggest that constraint structure can act as a computational prior, shaping how learning is performed rather than serving only as a penalty. Across the tested problems, achieving competitive solution quality does not require nonlinear completion and exact implicit gradients at every training update. Instead, effective training can rely on cheaper, consistent state and gradient computations. This separation makes computational efficiency a property of the learning design, beyond implementation optimization alone. Under limited computational budgets, such savings create room for more training or broader model exploration on the same hardware. The broader lesson is to allocate computational effort according to the needs of training and inference, using mathematical and physical structure to reduce repeated work. These benefits depend on linear surrogate accuracy and conditioning, motivating adaptive construction and extensions to multiple operating modes.

## 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*, volume 32, 2019.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145, 2017.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5230–5242, 2022.

Justin Domke. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 318–326, 2012.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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.

Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. JFB: Jacobian-free backpropagation for implicit networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6648–6656, 2022.

Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24247–24260, 2021.

Andrea Goertzen, Kaveh Alim, Youngjae Min, and Navid Azizan. HardNet++: Nonlinear constraint enforcement in neural networks. *arXiv preprint arXiv:2604.19669*, 2026. doi: 10.48550/arXiv.2604.19669.

Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):3988–4004, 2022.

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.

Ashfaq Iftakher, Rahul Golder, Bimol Nath Roy, and M. M. Faruque Hasan. Physics-informed neural networks with hard nonlinear equality and inequality constraints. *Computers & Chemical Engineering*, 204:109418, 2026. doi: 10.1016/j.compchemeng.2025.109418.

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

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, pp. 44366–44404, 2025. doi: 10.52202/085713-1323.

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. doi: 10.1109/TPWRS.2020.3026379.

Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H. Low. DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. *IEEE Systems Journal*, 17 (1):673–683, 2023. doi: 10.1109/JSYST.2022.3201041.

Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky T. Q. Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, Jing Dong, Brandon Amos, and Mustafa Mukadam. Theseus: A library for differentiable nonlinear optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3801–3818, 2022.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

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.

Ahmed S. Zamzam and Kyri Baker. Learning optimal solutions for extremely fast AC optimal power flow. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2020. doi: 10.1109/SmartGridComm47815.2020.9303008.

## Overview of the Appendix

- Appendix A (theoretical analysis and proofs) analyzes sample calibration, state and sensitivity errors, and task and parameter gradient errors, and provides conditions and proofs for numerical stability, optimization with the sample-calibrated linear surrogate, convergence from warm starts, and computational complexity.
- Appendix B collects the construction, routing, and selection of multiple label-independent surrogates in one subsection. The primary method shares one frozen matrix and explicit map between the training and inference forms.
- Appendix C (problem formulations and reductions) presents the underlying optimization problems, partitions into control and state variables, reduced equations, and derivations of the label-independent surrogate and its sample-calibrated training form for AC optimal power flow, water and gas networks in steady state, and CSCP.
- Appendix D (experimental settings and reproducibility) summarizes system configurations, data splits, baselines, numerical gradient checks, and comparison protocols. It supplements the CSCP gradient, descent, and solution branch recovery diagnostics, distinguishes independent training repetitions from timing fluctuations at fixed states, reports the sample calibration ablation, and provides access to the code.
- Appendix E discloses the use of AI tools for writing assistance and language polishing.

## A Theoretical and Numerical Analysis with Proofs

This appendix provides the complete conditions and proofs for Sections 3.1 and 3.2, covering state and gradient errors, numerical stability, optimization of the sample-calibrated linear surrogate objective, convergence from warm starts, and amortized computational cost. Here, $\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x})$ denotes the state obtained by solving $\boldsymbol{h}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x}) = \boldsymbol{0}$ on a fixed regular solution branch. For sample $i$, use $\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i)$. Throughout the appendix, the sample-calibrated linear surrogate $\tilde{\boldsymbol{z}}_i(\boldsymbol{u})$ is used for training, while the label-independent surrogate $\boldsymbol{A}^{-1}\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})$ supplies warm starts at inference. Both forms share the frozen matrix and explicit map, but only the former includes $\boldsymbol{r}_i$. With $\boldsymbol{J}_{f,v} := \partial f/\partial \boldsymbol{v}$, the exact completion and sample-calibrated linear surrogate sensitivities are $\boldsymbol{J}_{z_h,u}$ and $\boldsymbol{J}_{\tilde{z}_i,u}$, respectively. $D^2$ denotes a second derivative, not a Jacobian.

**Scaling convention.** To make the mapping from physical units reproducible, denote the fixed, invertible, positive diagonal scaling matrices in Section 3.1 by $\boldsymbol{W}_x, \boldsymbol{W}_u, \boldsymbol{W}_z, \boldsymbol{W}_h, \boldsymbol{W}_g$. They act on the continuous instance components, controls, state variables, equality residuals, and general inequality residuals, respectively. To distinguish explicitly scaled quantities from anchor notation, use the superscript nd: $\boldsymbol{x}^{\mathrm{nd}} = \boldsymbol{W}_x\boldsymbol{x}_{\mathrm{c}}$, $\boldsymbol{u}^{\mathrm{nd}} = \boldsymbol{W}_u\boldsymbol{u}$, $\boldsymbol{z}^{\mathrm{nd}} = \boldsymbol{W}_z\boldsymbol{z}$, $\boldsymbol{h}^{\mathrm{nd}} = \boldsymbol{W}_h\boldsymbol{h}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x})$, and $\boldsymbol{g}^{\mathrm{nd}} = \boldsymbol{W}_g\boldsymbol{g}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x})$. Vectors use Euclidean norms, and matrices use the induced norms of the corresponding input and output spaces. If the learning model directly outputs physical coordinates

$\boldsymbol{u}$, the fixed inverse scaling must be included in the output layer and the automatic differentiation chain for $\boldsymbol{J}_{\theta,i}$. Scales are determined from the training set for constructing candidates or engineering base values and frozen before training. Model checkpoint thresholds may be chosen on a separate independent validation set. No scale may be adjusted using the final test set.

### A.1 Exact Completion and Interpolation with Sample-Calibrated Linear Surrogates

The frozen coefficients shared by both forms satisfy $\boldsymbol{A}_i = \boldsymbol{A}$ and $\boldsymbol{b}_i(\boldsymbol{u}) = \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x}_i)$ for every sample.

$$\boldsymbol{J}_{h,u}(\boldsymbol{u}, \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}); \boldsymbol{x}) + \boldsymbol{J}_{h,z}(\boldsymbol{u}, \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}); \boldsymbol{x}) \boldsymbol{J}_{z_h,u}(\boldsymbol{u}; \boldsymbol{x}) = 0, \tag{A.1a}$$

and hence

$$\boldsymbol{J}_{z_h,u}(\boldsymbol{u}; \boldsymbol{x}) = -\boldsymbol{J}_{h,z}(\boldsymbol{u}, \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}); \boldsymbol{x})^{-1} \boldsymbol{J}_{h,u}(\boldsymbol{u}, \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}); \boldsymbol{x}). \tag{A.1b}$$

This result holds only on the selected local regular solution branch. The description as a smooth map is no longer valid if $\boldsymbol{J}_{h,z}$ becomes singular, the reference solution branch associated with the labeled point ceases to exist, or a solution branch switch occurs.

For sample calibration, substituting $\boldsymbol{r}_i = \boldsymbol{A}_i \boldsymbol{z}_i^\star - \boldsymbol{b}_i(\boldsymbol{u}_i^\star)$ into the sample-calibrated linear surrogate and setting $\boldsymbol{u} = \boldsymbol{u}_i^\star$ yields

$$\tilde{\boldsymbol{z}}_i(\boldsymbol{u}_i^\star) = \boldsymbol{A}_i^{-1}[\boldsymbol{b}_i(\boldsymbol{u}_i^\star) + \boldsymbol{r}_i] = \boldsymbol{A}_i^{-1}[\boldsymbol{b}_i(\boldsymbol{u}_i^\star) + \boldsymbol{A}_i \boldsymbol{z}_i^\star - \boldsymbol{b}_i(\boldsymbol{u}_i^\star)] = \boldsymbol{z}_i^\star. \tag{A.2}$$

Thus, interpolation at the labeled point is an algebraic identity, but only for the sample-calibrated linear surrogate for that training sample. It neither establishes that the label is an exact root nor ensures that the original equalities hold in a neighborhood, and it cannot be applied directly to test samples. If the linear system is solved approximately, the actual interpolation error also includes a numerical term arising from amplification of the linear residual by $\boldsymbol{A}_i^{-1}$ (see Appendix A.4).

### A.2 Local State, Sensitivity, and Task Gradient Errors

This section supplies the explicit constants and the complete error propagation chain omitted from Section 3.2. The main text requires only locally Lipschitz Jacobians. Here, bounded second derivatives provide a stronger but more transparent sufficient condition. For models that are only $C^{1,1}$, the Hessian suprema below can be replaced by the corresponding Jacobian Lipschitz constants. Write $\delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_i^\star$ and $e_{0,i} = \|\boldsymbol{z}_i^\star - \boldsymbol{z}_h(\boldsymbol{u}_i^\star; \boldsymbol{x}_i)\|$. Algebraic interpolation at the labeled point does not eliminate $e_{0,i}$. If, with $\boldsymbol{u}_i^\star$ fixed, the label state and target root lie in the same regular ball described in Appendix A.6, and $\|\boldsymbol{J}_{h,z}(\boldsymbol{u}_i^\star, \boldsymbol{z}_h(\boldsymbol{u}_i^\star; \boldsymbol{x}_i))^{-1}\| \le \gamma_i$, the argument relating residuals to state errors in that appendix gives

$$e_{0,i} \le 2\gamma_i \|\boldsymbol{h}(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star; \boldsymbol{x}_i)\|_h \le 2\gamma_i \varepsilon_{\text{label},i}. \tag{A.2a}$$

This bound still requires the label to lie in the same local ball as the target root. A small label residual alone cannot identify the solution branch in a region with multiple roots.

A second-order Taylor expansion of $\boldsymbol{b}_i$ gives

$$\tilde{\boldsymbol{z}}_i(\boldsymbol{u}) = \boldsymbol{z}_i^\star + \boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}_i^\star)\delta\boldsymbol{u} + \boldsymbol{R}_{\text{c},i}(\boldsymbol{u}), \tag{A.2b}$$

where

$$\|\boldsymbol{R}_{\text{c},i}(\boldsymbol{u})\| \le \frac{1}{2}\|\boldsymbol{A}_i^{-1}\| M_{b,i} \|\delta\boldsymbol{u}\|^2, \qquad M_{b,i} = \sup_{\boldsymbol{v} \in B_{\rho_i}(\boldsymbol{u}_i^\star)} \|D^2 \boldsymbol{b}_i(\boldsymbol{v})\|_{\text{op}}. \tag{A.2c}$$

On the same exact solution branch,

$$\boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}_i) = \boldsymbol{z}_h(\boldsymbol{u}_i^\star; \boldsymbol{x}_i) + \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i)\delta\boldsymbol{u} + \boldsymbol{R}_{\text{e},i}(\boldsymbol{u}), \tag{A.2d}$$

with

$$\|\boldsymbol{R}_{\text{e},i}(\boldsymbol{u})\| \le \frac{1}{2} M_{\text{ex},i} \|\delta\boldsymbol{u}\|^2, \qquad M_{\text{ex},i} = \sup_{\boldsymbol{v} \in B_{\rho_i}(\boldsymbol{u}_i^\star)} \|D^2 \boldsymbol{z}_h(\boldsymbol{v}; \boldsymbol{x}_i)\|_{\text{op}}. \tag{A.2e}$$

Subtracting the two expansions and applying the triangle inequality gives the full state error bound used in the main text:

$$\|\tilde{\boldsymbol{z}}_i(\boldsymbol{u})-\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i)\|\le e_{0,i}+\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}_i^\star)-\boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star;\boldsymbol{x}_i)\|\|\delta\boldsymbol{u}\|+\frac{1}{2}\left(M_{\mathrm{ex},i}+\|\boldsymbol{A}_i^{-1}\|M_{b,i}\right)\|\delta\boldsymbol{u}\|^2. \tag{A.3}$$

Exact labels give $e_{0,i}=0$, and an affine $\boldsymbol{b}_i$ gives $M_{b,i}=0$. Sample calibration therefore removes the zeroth-order intercept error that is independent of prediction distance. Away from the labeled point, sensitivity mismatch and nonlinear curvature continue to govern the error.

The curvature of exact completion is itself amplified by problem conditioning. Fix two control directions $\boldsymbol{v},\boldsymbol{w}$. For this local derivation, evaluate quantities that depend on the state at $\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x})$ and write $\boldsymbol{J}_{z_h,u}$ for $\boldsymbol{J}_{z_h,u}(\boldsymbol{u};\boldsymbol{x})$. Taking two directional derivatives of $\boldsymbol{h}(\boldsymbol{u},\boldsymbol{z};\boldsymbol{x})=0$ yields

$$\boldsymbol{h}_{uu}[\boldsymbol{v},\boldsymbol{w}]+\boldsymbol{h}_{uz}[\boldsymbol{v},\boldsymbol{J}_{z_h,u}\boldsymbol{w}]+\boldsymbol{h}_{zu}[\boldsymbol{J}_{z_h,u}\boldsymbol{v},\boldsymbol{w}]+\boldsymbol{h}_{zz}[\boldsymbol{J}_{z_h,u}\boldsymbol{v},\boldsymbol{J}_{z_h,u}\boldsymbol{w}]+\boldsymbol{J}_{h,z}D^2\boldsymbol{z}_h[\boldsymbol{v},\boldsymbol{w}]=0. \tag{A.3a}$$

Therefore,

$$D^2\boldsymbol{z}_h[\boldsymbol{v},\boldsymbol{w}]=-\boldsymbol{J}_{h,z}^{-1}\big(\boldsymbol{h}_{uu}[\boldsymbol{v},\boldsymbol{w}]+\boldsymbol{h}_{uz}[\boldsymbol{v},\boldsymbol{J}_{z_h,u}\boldsymbol{w}]+\boldsymbol{h}_{zu}[\boldsymbol{J}_{z_h,u}\boldsymbol{v},\boldsymbol{w}]+\boldsymbol{h}_{zz}[\boldsymbol{J}_{z_h,u}\boldsymbol{v},\boldsymbol{J}_{z_h,u}\boldsymbol{w}]\big). \tag{A.3b}$$

If, throughout the relevant neighborhood, $\|\boldsymbol{J}_{h,z}^{-1}\|\le\gamma$, $\|\boldsymbol{J}_{h,u}\|\le\beta$, $\|\boldsymbol{h}_{uu}\|\le M_{uu}$, $\|\boldsymbol{h}_{uz}\|\le M_{uz}$, $\|\boldsymbol{h}_{zu}\|\le M_{zu}$, and $\|\boldsymbol{h}_{zz}\|\le M_{zz}$, then $\|\boldsymbol{J}_{z_h,u}\|\le\gamma\beta$ implies

$$M_{\mathrm{ex},i}\le\gamma\left[M_{uu}+(M_{uz}+M_{zu})\gamma\beta+M_{zz}(\gamma\beta)^2\right]. \tag{A.4}$$

Thus, as $\boldsymbol{J}_{h,z}$ approaches singularity, both the first-order sensitivity and the second-order remainder can grow.

We next identify the matrix sources of sensitivity mismatch. Fix sample $i$ and the current $\boldsymbol{u}$, and use the following notation only within this paragraph: $\boldsymbol{J}=\boldsymbol{J}_{h,z}(\boldsymbol{u},\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i);\boldsymbol{x}_i)$, $\boldsymbol{B}=-\boldsymbol{J}_{h,u}(\boldsymbol{u},\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i);\boldsymbol{x}_i)$, and $\boldsymbol{B}_{\mathrm{c}}=\boldsymbol{J}_{b_i,u}(\boldsymbol{u})$. The exact completion and sample-calibrated linear surrogate sensitivities are then $\boldsymbol{J}^{-1}\boldsymbol{B}$ and $\boldsymbol{A}_i^{-1}\boldsymbol{B}_{\mathrm{c}}$, respectively. The identity for matrix inverses gives

$$\boldsymbol{A}_i^{-1}\boldsymbol{B}_{\mathrm{c}}-\boldsymbol{J}^{-1}\boldsymbol{B}=\boldsymbol{A}_i^{-1}(\boldsymbol{B}_{\mathrm{c}}-\boldsymbol{B})-\boldsymbol{A}_i^{-1}(\boldsymbol{A}_i-\boldsymbol{J})\boldsymbol{J}^{-1}\boldsymbol{B}, \tag{A.4a}$$

and hence

$$\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})-\boldsymbol{J}_{z_h,u}(\boldsymbol{u};\boldsymbol{x}_i)\|\le\|\boldsymbol{A}_i^{-1}\|\|\boldsymbol{B}_{\mathrm{c}}-\boldsymbol{B}\|+\|\boldsymbol{A}_i^{-1}\|\|\boldsymbol{A}_i-\boldsymbol{J}\|\|\boldsymbol{J}^{-1}\|\|\boldsymbol{B}\|. \tag{A.5}$$

Equation (A.5) separates the error in the derivative $\boldsymbol{J}_{b_i,u}(\boldsymbol{u})$, the error in the state variable Jacobian, and amplification by the frozen and exact inverse operators. A matrix with good conditioning but an incorrect derivative may be stable but ineffective. A matrix close to the current Jacobian may still produce unstable gradients if it is nearly singular. Arbitrary rescaling of variables and equations changes matrix norms and condition numbers, so comparisons between candidates must first use dimensionless coordinates.

For the task gradients, let $\boldsymbol{\omega}_{\mathrm{e}}=(\boldsymbol{u},\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i))$ and $\boldsymbol{\omega}_{\mathrm{c}}=(\boldsymbol{u},\tilde{\boldsymbol{z}}_i(\boldsymbol{u}))$. Then

$$\boldsymbol{g}_{\mathrm{e},i}=\ell_u(\boldsymbol{\omega}_{\mathrm{e}})+\boldsymbol{J}_{z_h,u}(\boldsymbol{u};\boldsymbol{x}_i)^\top\ell_z(\boldsymbol{\omega}_{\mathrm{e}}), \tag{A.5a}$$

$$\boldsymbol{g}_{\mathrm{c},i}=\ell_u(\boldsymbol{\omega}_{\mathrm{c}})+\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})^\top\ell_z(\boldsymbol{\omega}_{\mathrm{c}}). \tag{A.5b}$$

Add and subtract $\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})^\top\ell_z(\boldsymbol{\omega}_{\mathrm{e}})$ in the difference between these expressions. If $\|\ell_u(\boldsymbol{u},\boldsymbol{z}_1)-\ell_u(\boldsymbol{u},\boldsymbol{z}_2)\|\le L_u^z\|\boldsymbol{z}_1-\boldsymbol{z}_2\|$, $\|\ell_z(\boldsymbol{u},\boldsymbol{z}_1)-\ell_z(\boldsymbol{u},\boldsymbol{z}_2)\|\le L_z^z\|\boldsymbol{z}_1-\boldsymbol{z}_2\|$, and $\|\ell_z(\boldsymbol{u},\boldsymbol{z})\|\le M_z$, then

$$\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|\le\left(L_u^z+\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})\|L_z^z\right)\|\tilde{\boldsymbol{z}}_i(\boldsymbol{u})-\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x}_i)\|+M_z\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})-\boldsymbol{J}_{z_h,u}(\boldsymbol{u};\boldsymbol{x}_i)\|. \tag{A.6}$$

Substituting (A.3) and (A.5) into (A.6), and retaining the local notation $\boldsymbol{J},\boldsymbol{B},\boldsymbol{B}_{\mathrm{c}}$ from (A.5), gives a fully explicit local bound:

$$\begin{aligned}\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\| \le \big(L_u^z+\|\boldsymbol{A}_i^{-1}\boldsymbol{B}_{\mathrm{c}}\|L_z^z\big)\Big[e_{0,i}+\|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}_i^\star)-\boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star;\boldsymbol{x}_i)\|\|\delta\boldsymbol{u}\|\Big]\\ +\tfrac{1}{2}\big(L_u^z+\|\boldsymbol{A}_i^{-1}\boldsymbol{B}_{\mathrm{c}}\|L_z^z\big)\big(M_{\mathrm{ex},i}+\|\boldsymbol{A}_i^{-1}\|M_{b,i}\big)\|\delta\boldsymbol{u}\|^2\\ +M_z\|\boldsymbol{A}_i^{-1}\|\Big[\|\boldsymbol{B}_{\mathrm{c}}-\boldsymbol{B}\|+\|\boldsymbol{A}_i-\boldsymbol{J}\|\|\boldsymbol{J}^{-1}\|\|\boldsymbol{B}\|\Big].\end{aligned} \tag{A.7}$$

Equation (A.7) explicitly connects numerical label error, distance from the prediction to the label, nonlinear curvature, matrix approximation, the derivative $\boldsymbol{J}_{b_i,u}(\boldsymbol{u})$, and amplification by both inverse operators to task gradient error. This explains why the main text presents the interpretation in terms of state and sensitivity errors, while reserving the full expression for the appendix.

Even with exact labels and $\boldsymbol{u}=\boldsymbol{u}_i^\star$, so that both states are evaluated at the same point, we still have

$$\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}=\left[\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}_i^\star)-\boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star;\boldsymbol{x}_i)\right]^\top \ell_z(\boldsymbol{u}_i^\star,\boldsymbol{z}_i^\star), \tag{A.8}$$

Thus, zeroth-order state interpolation does not imply first-order gradient accuracy.

Finally, if $\boldsymbol{g}_{\mathrm{e},i}\neq 0$, then

$$\boldsymbol{g}_{\mathrm{e},i}^\top\boldsymbol{g}_{\mathrm{c},i}=\|\boldsymbol{g}_{\mathrm{e},i}\|^2+\boldsymbol{g}_{\mathrm{e},i}^\top(\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i})\ge\|\boldsymbol{g}_{\mathrm{e},i}\|\left(\|\boldsymbol{g}_{\mathrm{e},i}\|-\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|\right). \tag{A.8a}$$

Consequently, $\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|/\|\boldsymbol{g}_{\mathrm{e},i}\|<1$ suffices to ensure a positive inner product. If the exact task gradient is $L$-Lipschitz, the descent lemma gives, for any $0<\eta<\frac{2\boldsymbol{g}_{\mathrm{c},i}^\top\boldsymbol{g}_{\mathrm{c},i}}{L\|\boldsymbol{g}_{\mathrm{c},i}\|^2}$,

$$L_{\mathrm{e},i}^{\mathrm{task}}(\boldsymbol{u}-\eta\boldsymbol{g}_{\mathrm{c},i})<L_{\mathrm{e},i}^{\mathrm{task}}(\boldsymbol{u}). \tag{A.10}$$

### A.3 Parameter Gradients, Auxiliary Terms, and Empirical Objectives

This section propagates the control gradient error bounds of Section 3.2 to model parameter space and derives explicit bounds for minibatch aggregation and an optional auxiliary equality term.

For a single sample,

$$\nabla_{\boldsymbol{\theta}}L_{\mathrm{c},i}^{\mathrm{task}}=\boldsymbol{J}_{\theta,i}^\top\boldsymbol{g}_{\mathrm{c},i},\qquad\nabla_{\boldsymbol{\theta}}L_{\mathrm{e},i}^{\mathrm{task}}=\boldsymbol{J}_{\theta,i}^\top\boldsymbol{g}_{\mathrm{e},i}. \tag{A.10a}$$

A positive inner product in output space does not automatically imply a positive inner product in parameter space, because $\boldsymbol{J}_{\theta,i}$ can change both angles and scales. Write $n_\theta=\dim\boldsymbol{\theta}$. If $\boldsymbol{J}_{\theta,i}\in\mathbb{R}^{p\times n_\theta}$ has full row rank and $\boldsymbol{g}_{\mathrm{e},i}\neq 0$, then

$$\|\boldsymbol{J}_{\theta,i}^\top(\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i})\|\le\sigma_{\max}(\boldsymbol{J}_{\theta,i})\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|, \tag{A.10b}$$

$$\|\boldsymbol{J}_{\theta,i}^\top\boldsymbol{g}_{\mathrm{e},i}\|\ge\sigma_{\min}(\boldsymbol{J}_{\theta,i})\|\boldsymbol{g}_{\mathrm{e},i}\|, \tag{A.10c}$$

Therefore,

$$\frac{\|\nabla_{\boldsymbol{\theta}}L_{\mathrm{c},i}^{\mathrm{task}}-\nabla_{\boldsymbol{\theta}}L_{\mathrm{e},i}^{\mathrm{task}}\|}{\|\nabla_{\boldsymbol{\theta}}L_{\mathrm{e},i}^{\mathrm{task}}\|}\le\kappa_2(\boldsymbol{J}_{\theta,i})\frac{\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|}{\|\boldsymbol{g}_{\mathrm{e},i}\|}. \tag{A.11}$$

Hence,

$$\kappa_2(\boldsymbol{J}_{\theta,i})\frac{\|\boldsymbol{g}_{\mathrm{c},i}-\boldsymbol{g}_{\mathrm{e},i}\|}{\|\boldsymbol{g}_{\mathrm{e},i}\|}<1 \tag{A.12}$$

is a sufficient condition for strict descent in parameter space for a single sample. General networks are not guaranteed to have full row rank at every point. Experiments should therefore report parameter space gradient metrics directly, rather than relying solely on (A.12).

For a minibatch $\mathcal{B}$, using the bounded controls $\boldsymbol{u}_{\theta,i}$ from Section 3.1, define

$$\mathcal{J}_{\mathrm{e},\mathcal{B}}^{\mathrm{task}}(\boldsymbol{\theta})=\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}L_{\mathrm{e},i}^{\mathrm{task}}(\boldsymbol{u}_{\theta,i}), \tag{A.12a}$$

$$\mathcal{J}_{\mathrm{c},\mathcal{B}}^{\mathrm{task}}(\boldsymbol{\theta})=\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}L_{\mathrm{c},i}^{\mathrm{task}}(\boldsymbol{u}_{\theta,i}). \tag{A.12b}$$

The chain rule directly gives

$$\nabla_{\boldsymbol{\theta}} \mathcal{J}^{\text{task}}_{\text{c},\mathcal{B}} - \nabla_{\boldsymbol{\theta}} \mathcal{J}^{\text{task}}_{\text{e},\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \boldsymbol{J}_{\theta,i}^{\top} (\boldsymbol{g}_{\text{c},i} - \boldsymbol{g}_{\text{e},i}), \tag{A.13}$$

and hence

$$\left\| \nabla_{\boldsymbol{\theta}} \mathcal{J}^{\text{task}}_{\text{c},\mathcal{B}} - \nabla_{\boldsymbol{\theta}} \mathcal{J}^{\text{task}}_{\text{e},\mathcal{B}} \right\| \leq \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \|\boldsymbol{J}_{\theta,i}\| \|\boldsymbol{g}_{\text{c},i} - \boldsymbol{g}_{\text{e},i}\|. \tag{A.14}$$

This bound does not require gradients from different samples to point in the same direction. However, the triangle inequality can discard favorable cancellation between samples. Accordingly, the main text also recommends measuring the aggregated parameter gradient error, rather than reporting only results for individual samples.

The basic loss in Eq. (5) contains supervision but no auxiliary equality penalty. The power, water, and CSCP experiments use such a penalty, with case-specific scaling and weights specified in Appendix D.2. We now analyze this extension. The basic example corresponds to $\lambda_h = 0$. The derivation uses the dimensionless coordinates frozen in Section 3.1 and reuses the notation $\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{h}, \boldsymbol{J}_{h,u}, \boldsymbol{J}_{h,z}, \boldsymbol{J}_{\tilde{z}_i,u}$. For the squared auxiliary equality term $R_i^{\text{eq}}(\boldsymbol{u}, \boldsymbol{z}) = \lambda_h \|\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}_i)\|_2^2$, exact completion gives $R_i^{\text{eq}}(\boldsymbol{u}, \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}_i)) \equiv 0$, whereas the control gradient evaluated at the sample-calibrated linear surrogate state is

$$\boldsymbol{q}^{\text{eq}}_{\text{c},i} = 2\lambda_h \left[ \boldsymbol{J}_{h,u}(\boldsymbol{u}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}); \boldsymbol{x}_i) + \boldsymbol{J}_{h,z}(\boldsymbol{u}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}); \boldsymbol{x}_i) \boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}) \right]^{\top} \boldsymbol{h}(\boldsymbol{u}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}); \boldsymbol{x}_i). \tag{A.15}$$

Let $e_{z,i} = \|\tilde{\boldsymbol{z}}_i(\boldsymbol{u}) - \boldsymbol{z}_h(\boldsymbol{u}; \boldsymbol{x}_i)\|$ and $e_{J,i} = \|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u}) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}; \boldsymbol{x}_i)\|$. Assume that the line segment joining the two states lies within the same regular tubular neighborhood and that $\|\boldsymbol{J}_{h,z}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}_i)\| \leq \overline{J}_{h,z,i}$, $\|\boldsymbol{J}_{h,u}(\boldsymbol{u}, \boldsymbol{z}_1; \boldsymbol{x}_i) - \boldsymbol{J}_{h,u}(\boldsymbol{u}, \boldsymbol{z}_2; \boldsymbol{x}_i)\| \leq L^z_{J_{h,u},i} \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|$, and $\|\boldsymbol{J}_{h,z}(\boldsymbol{u}, \boldsymbol{z}_1; \boldsymbol{x}_i) - \boldsymbol{J}_{h,z}(\boldsymbol{u}, \boldsymbol{z}_2; \boldsymbol{x}_i)\| \leq L^z_{J_{h,z},i} \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|$. The mean value theorem and the exact tangent identity (A.1a) imply

$$\|\boldsymbol{h}(\boldsymbol{u}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}); \boldsymbol{x}_i)\| \leq \overline{J}_{h,z,i} e_{z,i}, \tag{A.16}$$

and

$$\begin{aligned} &\|\boldsymbol{J}_{h,u}(\boldsymbol{u}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}); \boldsymbol{x}_i) + \boldsymbol{J}_{h,z}(\boldsymbol{u}, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}); \boldsymbol{x}_i) \boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})\| \\ &\quad \leq \left( L^z_{J_{h,u},i} + L^z_{J_{h,z},i} \|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})\| \right) e_{z,i} + \overline{J}_{h,z,i} e_{J,i}. \end{aligned} \tag{A.17}$$

Therefore,

$$\|\boldsymbol{q}^{\text{eq}}_{\text{c},i}\| \leq 2\lambda_h \overline{J}_{h,z,i} e_{z,i} \Big[ \left( L^z_{J_{h,u},i} + L^z_{J_{h,z},i} \|\boldsymbol{J}_{\tilde{z}_i,u}(\boldsymbol{u})\| \right) e_{z,i} + \overline{J}_{h,z,i} e_{J,i} \Big]. \tag{A.18}$$

The control gradient bias contributed by the squared auxiliary equality term is thus $O(e_{z,i}^2) + O(e_{z,i} e_{J,i})$. In particular, at an exact labeled point with $e_{z,i} = 0$, this auxiliary gradient vanishes, although the task gradient may remain inaccurate because of the sensitivity mismatch in (A.8).

Define the full minibatch objectives of the sample-calibrated linear surrogate and exact counterfactual completion, respectively, as

$$\mathcal{J}^{\text{train}}_{\text{c},\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left[ L^{\text{task}}_{\text{c},i} + R_i^{\text{sup}} + R_i^{\text{eq}}(\boldsymbol{u}_i, \tilde{\boldsymbol{z}}_i(\boldsymbol{u}_i)) \right], \tag{A.18a}$$

$$\mathcal{J}^{\text{cf}}_{\text{e},\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left[ L^{\text{task}}_{\text{e},i} + R_i^{\text{sup}} + R_i^{\text{eq}}(\boldsymbol{u}_i, \boldsymbol{z}_h(\boldsymbol{u}_i; \boldsymbol{x}_i)) \right]. \tag{A.18b}$$

The identical supervision terms cancel exactly in their difference, yielding

$$\left\| \nabla_{\boldsymbol{\theta}} \mathcal{J}^{\text{train}}_{\text{c},\mathcal{B}} - \nabla_{\boldsymbol{\theta}} \mathcal{J}^{\text{cf}}_{\text{e},\mathcal{B}} \right\| \leq \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \|\boldsymbol{J}_{\theta,i}\| \left[ \|\boldsymbol{g}_{\text{c},i} - \boldsymbol{g}_{\text{e},i}\| + \|\boldsymbol{q}^{\text{eq}}_{\text{c},i}\| \right]. \tag{A.19}$$

If the bound in (A.19) is strictly smaller than the norm of the exact counterfactual full minibatch gradient, the negative parameter gradient of the sample-calibrated linear surrogate loss remains a local descent direction for that counterfactual objective. Using the full training set gives the corresponding deterministic gradient bias bound for the empirical objective. This is still not a statistical generalization guarantee for the test distribution.

### A.4 Numerical Stability

All matrix and vector norms in this section are interpreted in the dimensionless coordinates frozen in Section 3.1. In particular, if the linear system in physical coordinates is $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{\varrho}$, conditioning and regularization must be assessed for $\boldsymbol{A}_{\text{nd}} = \boldsymbol{W}_h \boldsymbol{A} \boldsymbol{W}_z^{-1}$. Direct comparisons of matrices with mixed physical units would be dimensionally inconsistent. For simplicity, we continue to write $\boldsymbol{A}$ below.

Consider the exact relation $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{\varrho}$, while the computation actually uses $\boldsymbol{A} + \Delta\boldsymbol{A}$ and $\boldsymbol{\varrho} + \Delta\boldsymbol{\varrho}$. If $\eta_A = \|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\| < 1$, the Banach lemma gives $\|(\boldsymbol{A}+\Delta\boldsymbol{A})^{-1}\| \le \frac{\|\boldsymbol{A}^{-1}\|}{1-\eta_A}$. From $(\boldsymbol{A}+\Delta\boldsymbol{A})\Delta\boldsymbol{z} = \Delta\boldsymbol{\varrho} - \Delta\boldsymbol{A}\boldsymbol{z}$, we obtain

$$\|\Delta\boldsymbol{z}\| \le \frac{\|\boldsymbol{A}^{-1}\|}{1-\eta_A}\left(\|\Delta\boldsymbol{\varrho}\| + \|\Delta\boldsymbol{A}\|\|\boldsymbol{z}\|\right). \tag{A.20}$$

The sensitivity relation $\boldsymbol{A}\boldsymbol{S}_{\text{lin}} = \boldsymbol{B}_{\text{rhs}}$, where $\boldsymbol{S}_{\text{lin}}$ is the sensitivity and $\boldsymbol{B}_{\text{rhs}} = \boldsymbol{J}_{b,u}(\boldsymbol{u};\boldsymbol{x})$, similarly gives

$$\|\Delta\boldsymbol{S}_{\text{lin}}\| \le \frac{\|\boldsymbol{A}^{-1}\|}{1-\eta_A}\left(\|\Delta\boldsymbol{B}_{\text{rhs}}\| + \|\Delta\boldsymbol{A}\|\|\boldsymbol{S}_{\text{lin}}\|\right). \tag{A.21}$$

For a sparse LU or QR solve that is backward stable, the typical relative forward error is

$$O(\kappa(\boldsymbol{A})\epsilon_{\text{mach}}), \qquad \kappa(\boldsymbol{A})\epsilon_{\text{mach}} \ll 1. \tag{A.22}$$

The transposed triangular solve has the same condition number. Computing $\boldsymbol{r}_i = \boldsymbol{A}_i\boldsymbol{z}_i^\star - \boldsymbol{b}_i(\boldsymbol{u}_i^\star)$ can also involve cancellation between large quantities. Practical diagnostics should therefore include, at a minimum, interpolation error at labeled points, linear residuals, condition estimates, factor growth, and differences between single and double precision.

If $\boldsymbol{A}$ is symmetric positive definite in dimensionless coordinates, then, for $\lambda > 0$,

$$\kappa_2(\boldsymbol{A}+\lambda\boldsymbol{I}) = \frac{\lambda_{\max}(\boldsymbol{A})+\lambda}{\lambda_{\min}(\boldsymbol{A})+\lambda} \le \kappa_2(\boldsymbol{A}). \tag{A.23}$$

However, using the regularized matrix $\boldsymbol{A}_\lambda = \boldsymbol{A} + \lambda\boldsymbol{I}$ requires recomputing $\boldsymbol{r}_{i,\lambda} = \boldsymbol{A}_\lambda\boldsymbol{z}_i^\star - \boldsymbol{b}_i(\boldsymbol{u}_i^\star)$ to preserve interpolation at labeled points. The sample-calibrated linear surrogate derivative also becomes $\boldsymbol{A}_\lambda^{-1}\boldsymbol{J}_{b_i,u}$, so gradient fidelity must be reassessed. For a general nonsymmetric or nonnormal matrix, (A.23) provides no analogous direct guarantee. Row and column equilibration, preconditioning, or structured regularization should be considered.

The same errors due to finite precision affect initialization by the label-independent surrogate. If the computed linear solution is $\widehat{\boldsymbol{z}}^{\text{init}} = \boldsymbol{A}^{-1}\boldsymbol{b}(\hat{\boldsymbol{u}};\boldsymbol{x}) + \delta\boldsymbol{z}_{\text{lin}}$, then

$$\|\widehat{\boldsymbol{z}}^{\text{init}} - \boldsymbol{z}_h(\hat{\boldsymbol{u}};\boldsymbol{x})\| \le \|\boldsymbol{A}^{-1}\boldsymbol{b}(\hat{\boldsymbol{u}};\boldsymbol{x}) - \boldsymbol{z}_h(\hat{\boldsymbol{u}};\boldsymbol{x})\| + \|\delta\boldsymbol{z}_{\text{lin}}\|. \tag{A.24}$$

Thus, ill conditioning of $\boldsymbol{A}$ affects training states and gradients and can also push a test initialization computed in finite precision outside the local convergence region considered below.

#### A.4.1 Forward Residuals, Transposed Adjoint Residuals, and Implemented Gradients

Equations (A.20)–(A.24) describe how finite precision affects states and sensitivities. In an implementation, the forward linear residual and transposed adjoint residual are more directly observable. Fix a sample and a control value, and abbreviate $\boldsymbol{\rho} = \boldsymbol{b}_i(\boldsymbol{u}) + \boldsymbol{r}_i$, $\boldsymbol{z} = \boldsymbol{A}_i^{-1}\boldsymbol{\rho}$, and $\boldsymbol{v} = \boldsymbol{A}_i^{-\top}\ell_z(\boldsymbol{u},\boldsymbol{z})$. Define the residuals of the computed forward state $\widehat{\boldsymbol{z}}$ and adjoint $\widehat{\boldsymbol{v}}$ as

$$\boldsymbol{r}_{\text{p}} = \boldsymbol{A}_i\widehat{\boldsymbol{z}} - \boldsymbol{\rho}, \qquad \boldsymbol{r}_{\text{a}} = \boldsymbol{A}_i^\top\widehat{\boldsymbol{v}} - \ell_z(\boldsymbol{u},\widehat{\boldsymbol{z}}). \tag{A.24a}$$

Then

$$\|\widehat{\boldsymbol{z}} - \boldsymbol{z}\| \le \|\boldsymbol{A}_i^{-1}\|\,\|\boldsymbol{r}_{\text{p}}\|. \tag{A.24b}$$

Suppose the implemented control gradient is $\widehat{\boldsymbol{g}}_u = \ell_u(\boldsymbol{u},\widehat{\boldsymbol{z}}) + \boldsymbol{J}_{b_i,u}(\boldsymbol{u})^\top\widehat{\boldsymbol{v}}$, whereas the algebraically exact gradient of the sample-calibrated linear surrogate loss is $\boldsymbol{g}_u = \ell_u(\boldsymbol{u},\boldsymbol{z}) +$

$\boldsymbol{J}_{b_i,u}(\boldsymbol{u})^\top \boldsymbol{A}_i^{-\top} \ell_z(\boldsymbol{u}, \boldsymbol{z})$. If $\ell_u, \ell_z$ have Lipschitz constants $L_u^z, L_z^z$ with respect to $\boldsymbol{z}$, respectively, then

$$\|\widehat{\boldsymbol{g}}_u - \boldsymbol{g}_u\| \le \left[L_u^z + \|\boldsymbol{J}_{b_i,u}(\boldsymbol{u})\|\, \|\boldsymbol{A}_i^{-\top}\| L_z^z\right] \|\boldsymbol{A}_i^{-1}\|\, \|\boldsymbol{r}_{\mathrm{p}}\| + \|\boldsymbol{J}_{b_i,u}(\boldsymbol{u})\|\, \|\boldsymbol{A}_i^{-\top}\|\, \|\boldsymbol{r}_{\mathrm{a}}\|. \quad \text{(A.24c)}$$

*Proof.* Use $\widehat{\boldsymbol{z}} - \boldsymbol{z} = \boldsymbol{A}_i^{-1}\boldsymbol{r}_{\mathrm{p}}$ and

$$\widehat{\boldsymbol{v}} - \boldsymbol{v} = \boldsymbol{A}_i^{-\top}\left[\ell_z(\boldsymbol{u}, \widehat{\boldsymbol{z}}) - \ell_z(\boldsymbol{u}, \boldsymbol{z}) + \boldsymbol{r}_{\mathrm{a}}\right], \quad \text{(A.24d)}$$

Substitution into the difference between the two gradients, followed by the Lipschitz bounds, proves the result.

Equation (A.24c) therefore applies when the linear system is treated as an algebraic solve with a custom vector–Jacobian product (VJP). Differentiating through a fixed number of iterations of a linear solver instead yields the derivative of that finite algorithmic map. Unless its proximity to the derivative of the algebraic map is established separately, (A.24c) cannot replace an analysis of the algorithmic map.

### A.5 Optimization with the Sample-Calibrated Linear Surrogate and Biased Stochastic Updates

We distinguish two objects: the sample-calibrated linear surrogate empirical objective $J_{\mathrm{c}}$ that is actually optimized, and the counterfactual empirical objective $J_{\mathrm{e}}^{\mathrm{tr}}$ obtained by replacing each sample-calibrated linear surrogate state with the state on the selected exact solution branch. Under standard sampling, stochastic gradients can be unbiased for the former. Relative to the latter, they generally carry systematic bias caused by this approximate state computation. The following results concern standard updates of the SGD type only. They do not claim global convergence for Adam or for general sequences of nonconvex network parameters.

**Theorem A.1 (average gradient bound for stochastic gradients of the sample-calibrated linear surrogate loss).** Assume that $J_{\mathrm{c}}$ is bounded below by $J_{\inf} > -\infty$ and that $\nabla J_{\mathrm{c}}$ is $L_J$-Lipschitz. Let $\mathcal{F}_t$ denote the history before step $t$, and suppose the minibatch gradient satisfies

$$\mathbb{E}[\widehat{\boldsymbol{d}}_t \mid \mathcal{F}_t] = \nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t), \qquad \mathbb{E}[\|\widehat{\boldsymbol{d}}_t - \nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t)\|^2 \mid \mathcal{F}_t] \le \frac{\sigma_{\mathrm{SGD}}^2}{B_{\mathrm{mb}}}. \quad \text{(A.24e)}$$

With a constant step size $0 < \eta \le 1/L_J$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t)\|^2 \le \frac{2[J_{\mathrm{c}}(\boldsymbol{\theta}_0) - J_{\inf}]}{\eta T} + \frac{L_J \eta \sigma_{\mathrm{SGD}}^2}{B_{\mathrm{mb}}}. \quad \text{(A.25)}$$

*Proof.* By $L_J$-smoothness and the update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\widehat{\boldsymbol{d}}_t$,

$$J_{\mathrm{c}}(\boldsymbol{\theta}_{t+1}) \le J_{\mathrm{c}}(\boldsymbol{\theta}_t) - \eta \nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t)^\top \widehat{\boldsymbol{d}}_t + \frac{L_J \eta^2}{2}\|\widehat{\boldsymbol{d}}_t\|^2. \quad \text{(A.25a)}$$

Take the conditional expectation given $\mathcal{F}_t$, use unbiasedness, the variance bound, and $\mathbb{E}_t\|\widehat{\boldsymbol{d}}_t\|^2 = \|\nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t)\|^2 + \mathbb{E}_t\|\widehat{\boldsymbol{d}}_t - \nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t)\|^2$, and then apply $\eta \le 1/L_J$ and sum to obtain (A.25). With a fixed step size, the variance term does not vanish. The conclusion is therefore an average first-order optimality bound, rather than unconditional convergence of the entire sequence. □

Define the exact counterfactual empirical objective by

$$J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left[\ell_i^{\mathrm{task}}(\boldsymbol{u}_{\theta,i}, \boldsymbol{z}_h(\boldsymbol{u}_{\theta,i}; \boldsymbol{x}_i)) + R_i^{\mathrm{sup}}(\boldsymbol{u}_{\theta,i}) + R_i^{\mathrm{eq}}(\boldsymbol{u}_{\theta,i}, \boldsymbol{z}_h(\boldsymbol{u}_{\theta,i}; \boldsymbol{x}_i))\right]. \quad \text{(A.26)}$$

This is not test risk: it is defined only on the selected regular solution branches and valid domains of the training samples. For a squared equality residual, the final term vanishes identically under exact completion. Write the parameter gradient bias as

$$\boldsymbol{e}_t = \nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t) - \nabla J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_t). \quad \text{(A.27)}$$

Equation (A.19) provides a class of local, verifiable upper bounds. If the training trajectory stays within a common regular tubular neighborhood, and the network Jacobians, task gradient errors, and errors in the gradient of the auxiliary term remain controlled, these quantities jointly bound $\|\boldsymbol{e}_t\|$.

**Theorem A.2 (exact counterfactual first-order optimality under biased minibatch updates using the sample-calibrated linear surrogate).** Assume that $J_{\mathrm{e}}^{\mathrm{tr}}$ is bounded below by $J_{\mathrm{e,inf}}$ and has an $L_{J,\mathrm{e}}$-Lipschitz gradient. The minibatch gradient still satisfies Eq. (A.24e).

For the update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\widehat{\boldsymbol{d}}_t$, with $0 < \eta \le 1/(4L_{J,\mathrm{e}})$,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_t)\|^2 \le \frac{4[J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_0) - J_{\mathrm{e,inf}}]}{\eta T} + \frac{3}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{e}_t\|^2 + \frac{2L_{J,\mathrm{e}}\eta\sigma_{\mathrm{SGD}}^2}{B_{\mathrm{mb}}}. \tag{A.28}$$

*Proof.* Abbreviate $\boldsymbol{g}_t = \nabla J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_t)$, $\boldsymbol{d}_t = \nabla J_{\mathrm{c}}(\boldsymbol{\theta}_t) = \boldsymbol{g}_t + \boldsymbol{e}_t$, and $\boldsymbol{\xi}_t = \widehat{\boldsymbol{d}}_t - \boldsymbol{d}_t$. Conditional unbiasedness gives $\mathbb{E}_t\boldsymbol{\xi}_t = 0$ and $\mathbb{E}_t\|\boldsymbol{\xi}_t\|^2 \le \sigma_{\mathrm{SGD}}^2/B_{\mathrm{mb}}$. Smoothness of the exact counterfactual objective implies

$$\mathbb{E}_t J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_{t+1}) \le J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_t) - \eta\boldsymbol{g}_t^\top\boldsymbol{d}_t + \frac{L_{J,\mathrm{e}}\eta^2}{2}\left(\|\boldsymbol{d}_t\|^2 + \frac{\sigma_{\mathrm{SGD}}^2}{B_{\mathrm{mb}}}\right). \tag{A.28a}$$

Young's inequality gives

$$\boldsymbol{g}_t^\top\boldsymbol{d}_t = \|\boldsymbol{g}_t\|^2 + \boldsymbol{g}_t^\top\boldsymbol{e}_t \ge \frac{1}{2}\|\boldsymbol{g}_t\|^2 - \frac{1}{2}\|\boldsymbol{e}_t\|^2, \tag{A.28b}$$

Also, $\|\boldsymbol{d}_t\|^2 = \|\boldsymbol{g}_t + \boldsymbol{e}_t\|^2 \le 2\|\boldsymbol{g}_t\|^2 + 2\|\boldsymbol{e}_t\|^2$. Substitution yields

$$\mathbb{E}_t J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_{t+1}) \le J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta}_t) - \left(\frac{\eta}{2} - L_{J,\mathrm{e}}\eta^2\right)\|\boldsymbol{g}_t\|^2 + \left(\frac{\eta}{2} + L_{J,\mathrm{e}}\eta^2\right)\|\boldsymbol{e}_t\|^2 + \frac{L_{J,\mathrm{e}}\eta^2\sigma_{\mathrm{SGD}}^2}{2B_{\mathrm{mb}}}. \tag{A.28c}$$

When $\eta \le 1/(4L_{J,\mathrm{e}})$, the descent coefficient is at least $\eta/4$, and the bias coefficient is at most $3\eta/4$. Sum over $t = 0, \dots, T-1$, take the full expectation, apply the lower bound, and divide by $\eta T/4$ to obtain (A.28). □

If $\|\boldsymbol{e}_t\| \le \varepsilon$ along the trajectory, the second term of (A.28) is at most $3\varepsilon^2$. Deterministic updates correspond to $\sigma_{\mathrm{SGD}} = 0$. Choosing $\eta = O(T^{-1/2})$ gives an average squared exact counterfactual gradient of $O(T^{-1/2}) + O(\varepsilon^2)$. Thus, nonvanishing gradient bias from the sample-calibrated linear surrogate permits a guarantee only of reaching a first-order neighborhood determined by that bias. The result does not control test risk, establish the same constants for Adam, or imply global optimality for the original nonconvex problem.

### A.6 Warm Starts from the Label-Independent Surrogate, Local Newton Convergence, and Bounds Relating Residuals to State Errors

At test time, the label-independent surrogate computes the initial state $\boldsymbol{A}^{-1}\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})$, without the calibration term in the sample-calibrated linear surrogate $\tilde{\boldsymbol{z}}_i$. The matrix is fixed before deployment and does not depend on the current network output.

Fix a reference $\bar{\boldsymbol{u}}$ on the same exact solution branch and write $d_u = \|\boldsymbol{u} - \bar{\boldsymbol{u}}\|$. If $\boldsymbol{A}^{-1}\boldsymbol{J}_{b,u}(\boldsymbol{u};\boldsymbol{x})$ and $\boldsymbol{J}_{z_h,u}(\boldsymbol{u};\boldsymbol{x})$ are Lipschitz with constants $L_{\mathrm{lin}}$ and $L_{J_{z_h,u}}$, respectively, the integral Taylor remainder gives

$$\begin{aligned}\|\boldsymbol{A}^{-1}\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x}) - \boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x})\| \le{}& \|\boldsymbol{A}^{-1}\boldsymbol{b}(\bar{\boldsymbol{u}};\boldsymbol{x}) - \boldsymbol{z}_h(\bar{\boldsymbol{u}};\boldsymbol{x})\| \\ &+ \|\boldsymbol{A}^{-1}\boldsymbol{J}_{b,u}(\bar{\boldsymbol{u}};\boldsymbol{x}) - \boldsymbol{J}_{z_h,u}(\bar{\boldsymbol{u}};\boldsymbol{x})\|d_u + \frac{L_{J_{z_h,u}} + L_{\mathrm{lin}}}{2}d_u^2.\end{aligned} \tag{A.29}$$

Under $C^2$ regularity, the corresponding Lipschitz constants can be replaced by $\sup\|D^2\boldsymbol{z}_h(\boldsymbol{u};\boldsymbol{x})\|$ and $\sup\|D_{\boldsymbol{u}}^2[\boldsymbol{A}^{-1}\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})]\|$. If the label-independent surrogate is a first-order Taylor model constructed from the full Jacobian at a feasible anchor on the same solution branch of the same instance,

both the zeroth-order and first-order errors at the reference point vanish. Shared or clustered label-independent surrogates, physical approximations, and infeasible anchors generally do not have this property.

For a linear solution computed in finite precision $\widehat{\boldsymbol{z}}^{\text{init}} = \boldsymbol{A}^{-1}\boldsymbol{b}(\hat{\boldsymbol{u}}; \boldsymbol{x}) + \delta\boldsymbol{z}_{\text{lin}}$, the initial distance to the target root $\boldsymbol{z}^\star = \boldsymbol{z}_h(\hat{\boldsymbol{u}}; \boldsymbol{x})$ is bounded by the bound in (A.29) plus $\|\delta\boldsymbol{z}_{\text{lin}}\|$. This numerical term can be estimated using the bounds on backward stability in Appendix A.4 or the measured linear residual.

Let $\boldsymbol{F}_{\text{N}}(\boldsymbol{z}) = \boldsymbol{h}(\hat{\boldsymbol{u}}, \boldsymbol{z}; \boldsymbol{x})$. The following proof uses the dimensionless coordinates frozen in Section 3.1: set $\boldsymbol{z}^{\text{nd}} = \boldsymbol{W}_z\boldsymbol{z}$ and $\boldsymbol{F}_{\text{N}}^{\text{nd}}(\boldsymbol{z}^{\text{nd}}) = \boldsymbol{W}_h\boldsymbol{F}_{\text{N}}(\boldsymbol{W}_z^{-1}\boldsymbol{z}^{\text{nd}})$, and continue to write $\boldsymbol{z}, \boldsymbol{F}_{\text{N}}$ to avoid duplicate notation. This linear transformation of variables and residuals leaves the exact Newton step unchanged after mapping back to the original variables. Assume that the target root $\boldsymbol{z}^\star$ exists and that the following conditions hold on $B_r(\boldsymbol{z}^\star)$:

$$\|\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star)^{-1}\| \le \gamma_N, \qquad \|\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}_1) - \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}_2)\| \le L_N\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|, \qquad \gamma_N L_N r \le \frac{1}{2}. \tag{A.30}$$

If the actual initial distance is at most $r$, standard Newton iterations have the following local properties. For any $\boldsymbol{z} \in B_r(\boldsymbol{z}^\star)$,

$$\|\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star)^{-1}[\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}) - \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star)]\| \le \gamma_N L_N r \le \frac{1}{2}. \tag{A.30a}$$

The Banach lemma gives

$$\|\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z})^{-1}\| \le \frac{\gamma_N}{1 - \gamma_N L_N r}. \tag{A.31}$$

For the Newton update

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^{(k)})^{-1}\boldsymbol{F}_{\text{N}}(\boldsymbol{z}^{(k)}), \tag{A.31a}$$

write $\boldsymbol{e}_k = \boldsymbol{z}^{(k)} - \boldsymbol{z}^\star$. The integral remainder gives

$$\boldsymbol{e}_{k+1} = \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^{(k)})^{-1}\int_0^1 \left[\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^{(k)}) - \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star + t\boldsymbol{e}_k)\right]\boldsymbol{e}_k\, dt. \tag{A.31b}$$

Therefore,

$$\|\boldsymbol{e}_{k+1}\| \le \frac{\gamma_N L_N}{2(1 - \gamma_N L_N r)}\|\boldsymbol{e}_k\|^2. \tag{A.32}$$

When $\|\boldsymbol{e}_k\| \le r$, the bound in (A.32) is at most $r/2$. The iterates therefore remain in the ball and converge locally at a quadratic rate. This proof applies only to standard Newton iterations. Damping, line searches, trust regions, switching between PV and PQ bus types, and other practical root solvers require separate analyses.

Equality feasibility at deployment can be assessed without relying on the Newton proof above. Suppose the solver returns a state $\hat{\boldsymbol{z}}$ and a status flag $s_{\text{solve}}$. If $\hat{\boldsymbol{z}}$ is finite, $s_{\text{solve}} = 1$, and the stopping criterion includes a check of equality feasibility under the frozen scaling, then

$$\|\boldsymbol{W}_h\boldsymbol{h}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{z}}; \boldsymbol{x})\|_2 \le \varepsilon_{\text{solve}}. \tag{A.33}$$

This is a verifiable numerical feasibility criterion, not a guarantee of root existence, recovery of the reference solution branch associated with the labeled point, or global convergence.

We next prove the bound relating residuals to state errors within a regular neighborhood on the same solution branch. Assume $\hat{\boldsymbol{z}} \in B_r(\boldsymbol{z}^\star)$. From

$$\boldsymbol{F}_{\text{N}}(\hat{\boldsymbol{z}}) - \boldsymbol{F}_{\text{N}}(\boldsymbol{z}^\star) = \left[\int_0^1 \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star + t(\hat{\boldsymbol{z}} - \boldsymbol{z}^\star))\, dt\right](\hat{\boldsymbol{z}} - \boldsymbol{z}^\star) \tag{A.33a}$$

define the Jacobian averaged along the path $\bar{\boldsymbol{J}}_{F_{\text{N}},z}$. We have

$$\|\boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star)^{-1}(\bar{\boldsymbol{J}}_{F_{\text{N}},z} - \boldsymbol{J}_{F_{\text{N}},z}(\boldsymbol{z}^\star))\| \le \frac{1}{2}\gamma_N L_N\|\hat{\boldsymbol{z}} - \boldsymbol{z}^\star\| \le \frac{1}{4}. \tag{A.33b}$$

Applying the Banach lemma again shows that $\bar{\boldsymbol{J}}_{F_{\mathrm{N}},z}$ is invertible, with $\|\bar{\boldsymbol{J}}_{F_{\mathrm{N}},z}^{-1}\| \le \gamma_N/(1-1/4) < 2\gamma_N$. Since $\boldsymbol{F}_{\mathrm{N}}(\boldsymbol{z}^\star) = 0$,

$$\|\hat{\boldsymbol{z}} - \boldsymbol{z}^\star\|_z \le 2\gamma_N \|\boldsymbol{h}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{z}}; \boldsymbol{x})\|_h \le 2\gamma_N \varepsilon_{\text{solve}}. \tag{A.34}$$

If, within this ball, the task loss is $L_{\ell,z}$-Lipschitz in $\boldsymbol{z}$ under $\|\cdot\|_z$, then

$$|\ell_x^{\text{task}}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{z}}) - \ell_x^{\text{task}}(\hat{\boldsymbol{u}}, \boldsymbol{z}^\star)| \le 2L_{\ell,z}\gamma_N \varepsilon_{\text{solve}}. \tag{A.35}$$

Equation (A.34) also controls the label error in Appendix A.2 when the label state and target root lie in the same ball. Globally, in the presence of multiple roots, a small equality violation does not imply recovery of the reference solution branch associated with the labeled point: a solver may return another exact root with zero equality violation while remaining far from the target root. Equations (A.33)–(A.35) therefore cannot be invoked when the root does not exist, mode information is incorrect, the wrong solution branch is selected, the linear solve fails, or the stopping criterion is not met.

### A.7 Computational Complexity, Net Inference Gains, Validation Cost, and Amortization

Let $a$ index the method configurations being compared. Table A1 summarizes the operation costs, counts, and timing quantities used below. The main method uses one frozen matrix and explicit map for both forms, so its cost contains no matrix selection or rebuilding term.

Table A1: Notation for the computational complexity analysis

| Operation costs | | Counts and accumulated costs | |
|---|---|---|---|
| $C_{\mathrm{NN}}$ | One network forward pass | $N, E$ | Training sample count and epoch count |
| $C_h, C_J$ | Equality residual evaluation and completion Jacobian assembly | $I_{\text{test}}, I_{\text{cold}}$ | Mean completion iterations with label-independent surrogate and default initialization |
| $F_J, S_J$ | Factorization and triangular solve for the current $\boldsymbol{J}_{h,z}$ | $N_{\text{val}}^{(a)}$ | Deployment validation runs during training |
| $F_A, S_A$ | Factorization and triangular solve for the frozen $\boldsymbol{A}$ | $C_{\text{val},v}^{(a)}$ | Cost of validation run $v$ |
| $C_b$ | Evaluation of $\boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$ | $T_{\text{sync/io}}^{(a)}$ | Synchronization, input, and output operations not counted in updates or validation |
| $C_{\text{cal}}$ | Calibration construction for one training sample | $T_{\text{label,raw}}^{\text{inc}}$ | Incremental labeling cost relative to matched Exact-ID |
| $C_{\text{exact}}, C_{\text{frame}}$ | Completion cost per sample per epoch for Exact-ID and EG-Opt | $T_{\text{off,pool}}^{\text{inc}}$ | Incremental preparation cost for both forms for an existing labeled pool |

EG-Opt training updates neither invoke the original nonlinear completion nor construct its completion Jacobian at the current state. They still require ordinary VJPs for the network, objective, constraints, and $\boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x})$. The sample-calibrated linear surrogate eliminates nonlinear iterations that depend on predictions and implicit differentiation. Freezing the matrices additionally enables factorization reuse. These provide algorithmic and implementation gains, respectively. Without factorization reuse, each update must include the corresponding $F_A$, but the iterative cost of the original nonlinear completion is not reintroduced. If $\boldsymbol{A}$ is solved with a finite number of iterations, $S_A$ must reflect the actual iterative workload, and the backward cost must reflect the differentiation scheme used.

At inference, the primary method performs a network forward pass, one linear solve with the label-independent surrogate, and one original nonlinear completion. If the completion solver performs $I_{\text{test}}$ iterations on average, then

$$C_{\text{infer}}^{\text{frame}} \approx C_{\mathrm{NN}} + C_b + S_A + I_{\text{test}}(C_h + C_J + F_J + S_J). \tag{A.36}$$

If the same completion solver requires $I_{\text{cold}}$ iterations on average from its default initialization, a simplified sufficient comparison condition for a net latency gain from the label-independent surrogate is

$$C_b + S_A < (I_{\text{cold}} - I_{\text{test}})(C_h + C_J + F_J + S_J). \tag{A.37}$$

Even when (A.37) does not hold, the label-independent surrogate may improve completion success or recovery of the reference solution branch associated with the labeled point. Latency, success rate, and solution branch reliability must therefore be reported separately.

Training wall clock time must include more than parameter updates. For the selected configuration $a$, we define three auditable timing scopes. Formal Training is defined as

$$T^{(a)}_{\text{target,formal}} = T^{(a)}_{\text{update}} + \sum_{v=1}^{N^{(a)}_{\text{val}}} C^{(a)}_{\text{val},v} + T^{(a)}_{\text{sync/io}}, \tag{A.38}$$

Here, $T^{(a)}_{\text{sync/io}}$ includes only synchronization, input, and output operations not already counted within updates or validation, avoiding double counting.

When the training instances and required labels already exist in a data pool, define

$$T^{(a)}_{\text{target,pool}} = T^{(a)}_{\text{target,formal}} + T^{(a)}_{\text{off,pool}}, \tag{A.38a}$$

Here, $T^{(a)}_{\text{off,pool}}$ includes one matrix factorization, sample calibration construction, and other preprocessing of the existing pool specific to each method. Starting from raw problem instances gives the further definition

$$T^{(a)}_{\text{target,raw}} = T^{(a)}_{\text{target,pool}} + T^{(a)}_{\text{label,raw}}. \tag{A.38b}$$

These three scopes correspond to formal training, end-to-end training from an existing labeled pool, and end-to-end training from raw instances. Table 1 uses the second scope, Eq. (A.38a), for all methods and cases. Timing starts with reading existing data and includes preprocessing, initialization, method-specific preparation, training and validation, synchronization, and model saving. Data and reference-label generation, framework imports, resource configuration, original-model integrity checks, and independent final testing are excluded. The fixed-budget endpoint replaces target attainment in the expressions above. All methods use the same labeled pool and timing boundaries. Equation (A.38b) defines the additional cost of starting from raw instances, but this is not the timing scope of Table 1.

All speedup ratios use EG-Opt as the reference and are defined as $T^{(a)}/T^{(\text{EG-Opt})}$ for the same case and timing scope. Table 1 compares end-to-end training and inference separately. Figure 3 compares synchronized individual updates. Figure 4 compares measured cumulative update time to target. Inference includes network prediction, initialization, nonlinear completion, and any correction or fallback actually used. Steps not performed by Direct-NN are not added to its cost. Numerator and denominator are computed separately for each timing scope, and speedups per update are not directly interpreted as end-to-end training speedups. Validation, failures, device synchronization, and other costs are assigned to their respective stages.

Figure 4 separately compares measured update time to target, $T^{(a)}_{\text{update}} = \sum_{e=1}^{E^{(a)}_{\text{hit}}} t^{(a)}_{\text{update},e}$, where $E^{(a)}_{\text{hit}}$ is the epoch of the first qualifying checkpoint and $t^{(a)}_{\text{update},e}$ is the measured duration of the update stage of epoch $e$ under the same protocol. This total excludes offline preparation, checkpoint evaluation, and disk writes. No case or method uses pretraining, including gas networks. It therefore compares only costs of the update stage and is not equivalent to the formal training or end-to-end totals in (A.38)–(A.38b). Comparisons of full training and end-to-end costs must add the relevant overheads. Using the same evaluation protocol does not justify assuming that these costs cancel. *Not reached* indicates that no checkpoint meets the joint target during the full training run.

Finally, consider the offline investment. Relative to a matched Exact-ID configuration, let the incremental cost of labeling raw instances be $T^{\text{inc}}_{\text{label,raw}}$, and define the incremental offline cost of preparing both forms on an existing pool as $T^{\text{inc}}_{\text{off,pool}} = F_A + NC_{\text{cal}}$.

Suppose the completion costs per training sample per epoch are $C_{\text{exact}}$ and $C_{\text{frame}}$, training runs for $E$ epochs, and the difference in validation cost for the selected configurations is $\Delta T_{\text{val}} = T_{\text{val,Ours}} - T_{\text{val,Exact-ID}}$, which may be positive or negative. A simplified condition for a lower end-to-end training cost than the matched method using exact completion is

$$EN(C_{\text{exact}} - C_{\text{frame}}) > T^{\text{inc}}_{\text{label,raw}} + T^{\text{inc}}_{\text{off,pool}} + \Delta T_{\text{val}}. \tag{A.40}$$

When all supervised methods share the same sample labels, the common labeling cost cancels in the cost difference, giving $T^{\text{inc}}_{\text{label,raw}} = 0$. For a comparison starting from raw instances, this cost must still be included in absolute times and speedup ratios. Table 1 instead starts from an existing labeled pool and excludes label generation for every method. If the same data, frozen coefficients, and sample calibration terms are reused across random seeds, hyperparameter settings, or network architectures, the shared offline cost is counted once in an aggregate comparison. Amortization cannot be assumed when only one model is trained.

## A.8 Additional Derivations for Regular Parameter Domains, Approximate Stationarity Transfer, Preconditioned Updates, and End-to-End Guarantees

### A.8.1 Role of the Regular Parameter Domain

The purpose of $\Theta^{\text{tr}}_{\text{reg}}$ is to specify the domain of the exact counterfactual objective, not to assume that the learning model automatically preserves feasibility. For $\boldsymbol{\theta} \notin \Theta^{\text{tr}}_{\text{reg}}$, at least one of the following may occur: the reference solution branch $\boldsymbol{z}_h(\cdot; \boldsymbol{x}_i)$ associated with the labeled point does not exist or is not unique, the segment joining the states leaves the regular tubular neighborhood, the task function or its partial derivatives become unbounded, the sample-calibrated linear surrogate state produces nonfinite constitutive function values, or the uniform inverse bound for $\boldsymbol{J}_{h,z}$ fails. In these cases, $J^{\text{tr}}_{\text{e}}$, $\boldsymbol{e}_t$, or constants in the local results of the main text and this appendix may be undefined. Bounded outputs, trust regions, rejection, or fallback can reduce such events, but theoretical reporting must retain samples outside the domain in the denominator.

An objective over the full parameter space could be defined by assigning a bounded extension to $J^{\text{tr}}_{\text{e}}$, for example a fixed penalty outside the regular domain. Its gradient would generally no longer equal the gradient of the original solution branch, and the boundary could become nonsmooth. We therefore use explicit conditional statements and account for the complementary event through failure penalties in the deployment risk.

### A.8.2 Additional Derivation for Approximate Stationarity Transfer

The triangle inequality gives

$$\|\nabla J^{\text{tr}}_{\text{e}}(\boldsymbol{\theta})\| \le \|\nabla J_{\text{c}}(\boldsymbol{\theta})\| + \|\nabla J_{\text{c}}(\boldsymbol{\theta}) - \nabla J^{\text{tr}}_{\text{e}}(\boldsymbol{\theta})\| \le \delta_{\text{c}} + \varepsilon. \tag{A.40a}$$

Although elementary, this result makes explicit the missing step between convergence of the sample-calibrated linear surrogate objective and first-order quality for the exact counterfactual objective. If $\varepsilon_{\text{grad}}$ does not decrease late in training, an arbitrarily small gradient of this objective can certify only a first-order neighborhood of fixed size.

### A.8.3 Additional Derivation for Preconditioned Updates with History Measurability

Write $f = J^{\text{tr}}_{\text{e}}$, retain the assumptions of Theorem A.2 that $f$ has a finite lower bound and an $L_{J,\text{e}}$-Lipschitz gradient, and consider the preconditioned update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{P}_t \widehat{\boldsymbol{d}}_t$. Let $\mathcal{F}_t$ denote the history before sampling the current minibatch, and let $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_t]$. Assume that the optimizer's preconditioning matrix $\boldsymbol{P}_t$ is $\mathcal{F}_t$-measurable and symmetric, and almost surely satisfies the uniform spectral bounds $\mu \boldsymbol{I} \preceq \boldsymbol{P}_t \preceq M \boldsymbol{I}$, where $0 < \mu \le M < \infty$. Thus, $\boldsymbol{P}_t$ may evolve with the history, but this assumption does not directly cover a preconditioner constructed from the current minibatch gradient.

Write $\boldsymbol{g}_t = \nabla f(\boldsymbol{\theta}_t)$, $\boldsymbol{d}_t = \nabla J_{\text{c}}(\boldsymbol{\theta}_t) = \boldsymbol{g}_t + \boldsymbol{e}_t$, and $\widehat{\boldsymbol{d}}_t = \boldsymbol{d}_t + \boldsymbol{\xi}_t$, and assume $\mathbb{E}_t \boldsymbol{\xi}_t = 0$ and $\mathbb{E}_t \|\boldsymbol{\xi}_t\|^2 \le \sigma^2_{\text{SGD}} / B_{\text{mb}}$. History measurability gives $\mathbb{E}_t[\boldsymbol{P}_t \boldsymbol{\xi}_t] = \boldsymbol{P}_t \mathbb{E}_t[\boldsymbol{\xi}_t] = \boldsymbol{0}$. Combining this with

the smoothness of $f$,

$$\mathbb{E}_t f(\boldsymbol{\theta}_{t+1}) \le f(\boldsymbol{\theta}_t) - \eta \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{d}_t + \frac{L_{J,\mathrm{e}}\eta^2}{2}\mathbb{E}_t\|\boldsymbol{P}_t\widehat{\boldsymbol{d}}_t\|^2. \tag{A.41}$$

The spectral bounds and Young's inequality give

$$\boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{d}_t = \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{g}_t + \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{e}_t \ge \mu\|\boldsymbol{g}_t\|^2 - M\|\boldsymbol{g}_t\|\|\boldsymbol{e}_t\| \ge \frac{\mu}{2}\|\boldsymbol{g}_t\|^2 - \frac{M^2}{2\mu}\|\boldsymbol{e}_t\|^2, \tag{A.42}$$

and

$$\mathbb{E}_t\|\boldsymbol{P}_t\widehat{\boldsymbol{d}}_t\|^2 \le 2M^2\|\boldsymbol{g}_t\|^2 + 2M^2\|\boldsymbol{e}_t\|^2 + \frac{M^2\sigma_{\mathrm{SGD}}^2}{B_{\mathrm{mb}}}. \tag{A.43}$$

Substitute these bounds into (A.41). When $0 < \eta \le \mu/(4L_{J,\mathrm{e}}M^2)$, the net descent coefficient on $\|\boldsymbol{g}_t\|^2$ is at least $\eta\mu/4$, and the coefficient on $\|\boldsymbol{e}_t\|^2$ is at most $\eta\left(\frac{M^2}{2\mu} + \frac{\mu}{4}\right)$. Summing over $t = 0, \dots, T-1$, taking full expectations, applying the lower bound on $f$, and dividing by $\eta\mu T/4$ gives the average gradient bound for preconditioning measurable with respect to the history.

### A.8.4 Additional Derivation Linking Local Error to End-to-End Outcomes

This section connects the local error bounds to empirical objectives and deployment outcomes. Fix the sample set and reference solution branches associated with the labeled points, use the bounded controls $\boldsymbol{u}_{\theta,i}$ from Section 3.1, and let $E_i(\boldsymbol{u})$ denote the state error bound in Eq. (A.3). If its local conditions hold and the task is $L_{\ell,z,i}$-Lipschitz in $\boldsymbol{z}$, then $|L_{\mathrm{c},i}^{\mathrm{task}}(\boldsymbol{u}) - L_{\mathrm{e},i}^{\mathrm{task}}(\boldsymbol{u})| \le L_{\ell,z,i}E_i(\boldsymbol{u})$. All norms use the frozen dimensionless scaling.

If every training sample satisfies the conditions of Appendix A.3, apply Eq. (A.19) to the full training set and define the aggregate gradient error bound

$$\|\nabla J_{\mathrm{c}}(\boldsymbol{\theta}) - \nabla J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta})\| \le \mathcal{B}_\theta(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{J}_{\theta,i}\|\big(\|\boldsymbol{g}_{\mathrm{c},i} - \boldsymbol{g}_{\mathrm{e},i}\| + \|\boldsymbol{q}_{\mathrm{c},i}^{\mathrm{eq}}\|\big). \tag{A.44}$$

Every quantity on the right is evaluated at $\boldsymbol{u}_{\theta,i}$. The standard configuration has $\boldsymbol{q}_{\mathrm{c},i}^{\mathrm{eq}} = \boldsymbol{0}$. Thus, if $\|\nabla J_{\mathrm{c}}(\boldsymbol{\theta})\| \le \delta_{\mathrm{c}}$, the triangle inequality gives $\|\nabla J_{\mathrm{e}}^{\mathrm{tr}}(\boldsymbol{\theta})\| \le \delta_{\mathrm{c}} + \mathcal{B}_\theta(\boldsymbol{\theta})$. The quality of an approximate stationary point of the sample-calibrated linear surrogate objective therefore also depends on gradient bias.

For deployment, define the valid event $S_i(\boldsymbol{\theta})$: the target root exists, the solver successfully returns a finite state $\hat{\boldsymbol{z}}_i$ with equality residual at most $\varepsilon_{\mathrm{solve}}$, and that state lies in the regular ball around the target root required by Appendix A.6. In addition, the sample-calibrated linear surrogate state satisfies the local error bound above, and a common Lipschitz bound holds for the task between the sample-calibrated linear surrogate state, target root, and returned state. This event is stronger than a flag indicating solver success alone. For samples satisfying $S_i(\boldsymbol{\theta})$, Eqs. (A.3) and (A.34), together with the triangle inequality, imply

$$\begin{aligned} \|\hat{\boldsymbol{z}}_i - \tilde{\boldsymbol{z}}_i(\boldsymbol{u}_{\theta,i})\| &\le E_i(\boldsymbol{u}_{\theta,i}) + 2\gamma_{N,i}\varepsilon_{\mathrm{solve}}, \\ |\ell_i^{\mathrm{task}}(\boldsymbol{u}_{\theta,i}, \hat{\boldsymbol{z}}_i) - L_{\mathrm{c},i}^{\mathrm{task}}(\boldsymbol{u}_{\theta,i})| &\le L_{\ell,z,i}\big[E_i(\boldsymbol{u}_{\theta,i}) + 2\gamma_{N,i}\varepsilon_{\mathrm{solve}}\big]. \end{aligned} \tag{A.45}$$

To include the complementary event in the same empirical comparison, fix an interval $[m_i, M_i]$ in advance and apply the same clipping map to the sample-calibrated linear surrogate and deployment task values, denoting the results by $\bar{\ell}_{\mathrm{c},i}$ and $\bar{\ell}_{\mathrm{dep},i}$. Nonfinite sample-calibrated linear surrogate task values are assigned $M_i$. When $S_i$ fails, the deployment loss used in this analysis is also assigned $M_i$. Let $\bar{J}_{\mathrm{c}}^{\mathrm{task}} = N^{-1}\sum_i \bar{\ell}_{\mathrm{c},i}$, $\bar{J}_{\mathrm{dep}} = N^{-1}\sum_i \bar{\ell}_{\mathrm{dep},i}$, and $\Delta_i = M_i - m_i$. Since clipping is 1-Lipschitz, applying Eq. (A.45) samplewise and bounding the complementary event by $\Delta_i$ gives

$$|\bar{J}_{\mathrm{dep}} - \bar{J}_{\mathrm{c}}^{\mathrm{task}}| \le \frac{1}{N}\sum_{i:S_i(\boldsymbol{\theta})} L_{\ell,z,i}\big[E_i(\boldsymbol{u}_{\theta,i}) + 2\gamma_{N,i}\varepsilon_{\mathrm{solve}}\big] + \frac{1}{N}\sum_{i:\neg S_i(\boldsymbol{\theta})}\Delta_i. \tag{A.46}$$

This expression retains every sample in the denominator and separates the contributions of sample-calibrated linear surrogate state error, completion accuracy, and invalid events. Small state error for the sample-calibrated linear surrogate does not rule out failures or incorrect solution branches, and successful completion does not ensure accurate gradients of its loss. Clipping defines only the analytical quantities in this section: it does not change the experimental metrics in the main text or provide a generalization guarantee for the test distribution. Sample calibration is used only as an analytical reference and does not enter deployment.

# B Selection Among Multiple Label-Independent Surrogates

This appendix explains how to select among multiple label-independent surrogates, extending the single model in Eq. (3). Each selected candidate supplies inference warm starts and, after sample calibration, defines a sample-calibrated linear surrogate for training. Let $K$ denote the number of label-independent surrogates in a candidate library. Let $k \in \{1, \dots, K\}$ denote the candidate index. For model $k$, constructed at a chosen anchor or for a known mode, $\boldsymbol{A}^{(k)}$ and $\boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$ denote its frozen matrix and explicit map, respectively. The superscript $(k)$ identifies the model, not a power.

## B.1 Construction, Routing, and Selection of Label-Independent Surrogates

**Details of label-independent surrogate construction.** Applying the construction in Eq. (3) separately to each candidate moves terms independent of $\boldsymbol{z}$ into $\boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$, giving $\boldsymbol{A}^{(k)}\boldsymbol{z} = \boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$. The constant equality residual vanishes if the anchor is strictly feasible. For a parameterized problem, $\boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$ may retain explicit terms or known physical nonlinearities that can be evaluated directly from $(\boldsymbol{x}, \boldsymbol{u})$ and do not involve $\boldsymbol{z}$. All nonlinear dependence on $\boldsymbol{z}$ is frozen or linearized at the anchor. Thus, given $(\boldsymbol{x}, \boldsymbol{u})$, only a reusable linear system must be solved. However, $\boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$ need not be affine in $\boldsymbol{u}$. A single anchor gives one label-independent surrogate, whereas multiple anchors or known modes give a finite candidate library. Appendix C defines the variables and reduced equations for power, water, and gas networks.

Retaining all linear Taylor coefficients as in Eq. (3) gives $\boldsymbol{A}^{(k)} = \boldsymbol{J}^{(k)}_{h,z}$, where $\boldsymbol{J}^{(k)}_{h,z}$ denotes the constraint Jacobian evaluated at the anchor of candidate $k$. This equality need not hold when coefficients are simplified using physical structure. Appendix A.2 analyzes the resulting sensitivity mismatch. Under either construction, the sample-calibrated linear surrogate and label-independent surrogate must share the frozen matrices, $\boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$ maps, and their constant terms. Matching algebraic forms alone does not establish this coefficient consistency.

The sample-calibrated linear surrogate matches the training label at each labeled point. Consequently, the reconstruction error after calibration at those points cannot be used for matrix selection. Moreover, the object being selected is not merely a matrix: it is a deployable candidate rule $a$ comprising the matrix library $\{(\boldsymbol{A}^{(k)}_a, \boldsymbol{b}^{(k)}_{a,x})\}_{k=1}^{K_a}$, a frozen selection rule, scaling, and numerical configuration, where $K_a$ is the library size. Within each candidate, $k_a(\boldsymbol{x}) \in \{1, \dots, K_a\}$ denotes the candidate index selected for instance $\boldsymbol{x}$. The selection rule is frozen before training and uses only instance and mode information available at test time. Each candidate must therefore be evaluated as a whole for its effects on local sensitivities, task gradients, warm starts from the label-independent surrogate, routing and mode coverage, numerical conditioning, and actual solution cost. For sample $i$, this selection gives $k_{a,i} = k_a(\boldsymbol{x}_i)$, with $\boldsymbol{A}_{a,i} = \boldsymbol{A}^{(k_{a,i})}_a$ and $\boldsymbol{b}_{a,i} = \boldsymbol{b}^{(k_{a,i})}_{a,x_i}$. The corresponding sample-calibrated linear surrogate sensitivity at the labeled point is then $\boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}^\star_i) = \boldsymbol{A}^{-1}_{a,i}\boldsymbol{J}_{b_{a,i},u}(\boldsymbol{u}^\star_i)$, while the exact solution branch sensitivity is given by Eq. (A.1b). When labels are solutions computed to finite tolerance, however, diagnostics of exact sensitivities must use roots recomputed to high accuracy on the same solution branch, rather than assuming $\boldsymbol{z}^\star_i = \boldsymbol{z}_h(\boldsymbol{u}^\star_i; \boldsymbol{x}_i)$.

To prevent physical units from dominating the score, use frozen $\boldsymbol{W}_{u,i}, \boldsymbol{W}_{z,i}$ to nondimensionalize the quantities. The subscript $i$ may be omitted when a benchmark uses common scales for the system. We retain it here to accommodate joint scoring across systems. Define

$$\widehat{\mathcal{E}}_{\text{sens}}(a) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left\| \boldsymbol{W}_{z,i} \left[ \boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}^\star_i) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}^\star_i; \boldsymbol{x}_i) \right] \boldsymbol{W}^{-1}_{u,i} \right\|^2_F . \tag{B.1}$$

To target task gradients directly, let $\boldsymbol{v}_i = \nabla_{\boldsymbol{z}} \ell_i^{\text{task}}(\boldsymbol{u}_i^\star, \boldsymbol{z}_h(\boldsymbol{u}_i^\star; \boldsymbol{x}_i))$ and denote the dimensionless control gradient by $\boldsymbol{g}_{\text{e},i}^{\text{nd}} = \boldsymbol{W}_{u,i}^{-\top} \boldsymbol{g}_{\text{e},i}$. Define

$$\widehat{\mathcal{E}}_{\text{task}}(a) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \frac{\left\| \boldsymbol{W}_{u,i}^{-\top} \left[ \boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i) \right]^\top \boldsymbol{v}_i \right\|^2}{\|\boldsymbol{g}_{\text{e},i}^{\text{nd}}\|^2 + \epsilon}. \tag{B.2}$$

The numerator is precisely the error in the task gradient with respect to dimensionless $\boldsymbol{u}^{\text{nd}}$ induced by a mismatch in completion sensitivities. This prevents the units of the controls from arbitrarily changing candidate selection.

At test time, each candidate's label-independent surrogate computes $\boldsymbol{A}_{a,i}^{-1} \boldsymbol{b}_{a,i}(\boldsymbol{u})$. Initialization must therefore be scored using $\boldsymbol{A}_{a,i}^{-1} \boldsymbol{b}_{a,i}(\boldsymbol{u})$, rather than the sample-calibrated linear surrogate $\tilde{\boldsymbol{z}}_{a,i}$, which already interpolates the label:

$$\widehat{\mathcal{E}}_{\text{init}}(a) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left\| \boldsymbol{W}_{z,i} [\boldsymbol{A}_{a,i}^{-1} \boldsymbol{b}_{a,i}(\boldsymbol{u}_i^\star) - \boldsymbol{z}_h(\boldsymbol{u}_i^\star; \boldsymbol{x}_i)] \right\|^2. \tag{B.3}$$

For data with multiple modes or solution branches, a routing or coverage loss must also be specified in advance. For example, if $q_i$ is the target mode label and $q_a(\boldsymbol{x}_i)$ is the mode assigned by the candidate routing, a bounded loss is

$$\widehat{\mathcal{E}}_{\text{route}}(a) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathbf{1}\{q_a(\boldsymbol{x}_i) \neq q_i\}, \tag{B.3a}$$

Alternatively, a preregistered penalty for the worst case can be assigned to uncovered modes, singular matrices, failed linear solves, and failed completions initialized by warm starts. This term may be omitted for a benchmark with a single fixed mode, provided that the reason is stated.

One implementable joint selection objective is

$$a^\star = \arg\min_a \left\{ \widehat{\mathcal{E}}_{\text{sens}}(a) + \lambda_{\text{task}} \widehat{\mathcal{E}}_{\text{task}}(a) + \lambda_0 \widehat{\mathcal{E}}_{\text{init}}(a) + \lambda_q \widehat{\mathcal{E}}_{\text{route}}(a) + \lambda_\kappa \mathcal{R}_\kappa(a) + \lambda_c \mathcal{C}(a) \right\}. \tag{B.4}$$

Here, $\mathcal{R}_\kappa(a)$ aggregates conditioning penalties over every matrix in the candidate library that routing can actually select. Let $\boldsymbol{A}_{a,\text{nd}}^{(k)} = \boldsymbol{W}_h \boldsymbol{A}_a^{(k)} \boldsymbol{W}_z^{-1}$ denote the dimensionless matrix under the corresponding system scales. For example, one may use

$$\mathcal{R}_\kappa(a) = \max_{k \le K_a} \left[ \max\{0, \log \kappa_2(\boldsymbol{A}_{a,\text{nd}}^{(k)}) - \log \kappa_{\max}\} \right]^2, \tag{B.5}$$

The term $\mathcal{C}(a)$ includes library size, new nonzeros from factorization, factor memory, routing time, and measured costs of numerical factorization and batched triangular solves. Equation (B.4) is only one scalarization. A Pareto frontier may also be reported. Candidate rules, scoring weights, scales, clipping thresholds, failure penalties, and rules for resolving ties are frozen before the offline selection scores are computed.

If the exact task gradient at a labeled point is very small, the normalization in (B.2) can be unstable. Directions relevant to the task should then be evaluated at preregistered perturbations around labeled points or early training checkpoints, with the procedure for generating these points included in the candidate definition. The perturbation radius must not be selected retrospectively to favor the observed validation results.

**Anchor Distance and Sensitivity.**

Let $\boldsymbol{x}_{\text{c}}$ denote the continuous instance components and define the complete local dimensionless coordinates under the frozen scaling by $\boldsymbol{\zeta} = (\boldsymbol{W}_x \boldsymbol{x}_{\text{c}}, \boldsymbol{W}_u \boldsymbol{u}, \boldsymbol{W}_z \boldsymbol{z})$. Discrete modes are compared only within the same mode. In this section, $\boldsymbol{J}_{h,z}(\boldsymbol{\zeta})$ and $\boldsymbol{J}_{h,u}(\boldsymbol{\zeta})$ are the Jacobians of the transformed equations with respect to dimensionless $\boldsymbol{z}$ and $\boldsymbol{u}$, namely the representations $\boldsymbol{W}_h \boldsymbol{J}_{h,z} \boldsymbol{W}_z^{-1}$ and $\boldsymbol{W}_h \boldsymbol{J}_{h,u} \boldsymbol{W}_u^{-1}$ of the physical Jacobians. If $\|\boldsymbol{J}_{h,z}(\boldsymbol{\zeta})^{-1}\| \le \gamma$, $\|\boldsymbol{J}_{h,u}(\boldsymbol{\zeta})\| \le \beta$, and $\boldsymbol{J}_{h,z}, \boldsymbol{J}_{h,u}$ are Lipschitz in $\boldsymbol{\zeta}$ with constants $L_z^\zeta, L_u^\zeta$, respectively, then, for two points $\boldsymbol{\zeta}_a, \boldsymbol{\zeta}_i$,

$$\left\| -\boldsymbol{J}_{h,z}(\boldsymbol{\zeta}_a)^{-1} \boldsymbol{J}_{h,u}(\boldsymbol{\zeta}_a) + \boldsymbol{J}_{h,z}(\boldsymbol{\zeta}_i)^{-1} \boldsymbol{J}_{h,u}(\boldsymbol{\zeta}_i) \right\| \le \left( \gamma L_u^\zeta + \gamma^2 \beta L_z^\zeta \right) \|\boldsymbol{\zeta}_a - \boldsymbol{\zeta}_i\|. \tag{B.6}$$

*Proof.* Write $\boldsymbol{J}_a = \boldsymbol{J}_{h,z}(\boldsymbol{\zeta}_a)$, $\boldsymbol{J}_i = \boldsymbol{J}_{h,z}(\boldsymbol{\zeta}_i)$, $\boldsymbol{U}_a = \boldsymbol{J}_{h,u}(\boldsymbol{\zeta}_a)$, and $\boldsymbol{U}_i = \boldsymbol{J}_{h,u}(\boldsymbol{\zeta}_i)$. Add and subtract $-\boldsymbol{J}_a^{-1}\boldsymbol{U}_i$ in $-\boldsymbol{J}_a^{-1}\boldsymbol{U}_a + \boldsymbol{J}_i^{-1}\boldsymbol{U}_i$ to obtain

$$\| - \boldsymbol{J}_a^{-1}\boldsymbol{U}_a + \boldsymbol{J}_i^{-1}\boldsymbol{U}_i\| \leq \|\boldsymbol{J}_a^{-1}\|\|\boldsymbol{U}_a - \boldsymbol{U}_i\| + \|\boldsymbol{J}_a^{-1} - \boldsymbol{J}_i^{-1}\|\|\boldsymbol{U}_i\|. \tag{B.6a}$$

Using the inverse identity $\boldsymbol{J}_a^{-1} - \boldsymbol{J}_i^{-1} = \boldsymbol{J}_a^{-1}(\boldsymbol{J}_i - \boldsymbol{J}_a)\boldsymbol{J}_i^{-1}$, together with the uniform inverse bounds and Lipschitz conditions, gives (B.6). □

This result explains why feasible anchors near the center of the data often offer smaller sensitivity bias, but it does not establish optimality of the Euclidean mean. An unprojected mean can be infeasible, cross solution branches, or have worse conditioning. Candidates must still be evaluated directly using (B.4) or the corresponding Pareto criteria.

**Tradeoffs in Candidate Libraries and Routing Strategies.**

Table B1: Tradeoffs among candidate matrices and routing strategies

| Candidate strategy | Sensitivity and initialization fit | Offline work and storage | Batching support | Main risks |
|---|---|---|---|---|
| Single nominal matrix with constant routing | Potentially poor | One factorization | Strongest | Derivative and initialization bias outside the operating region |
| Matrix at a feasible anchor near the center of the data | Typically closer to the training distribution | One factorization | Strongest | No direct task gradient optimization. May cross solution branches |
| Shared matrix selected for sensitivity | Directly targets Eq. (B.1) | A few offline probes | Strong | Overfitting to selection data. Suboptimal initialization |
| $K_a$ clustered matrices Fixed routing from $\boldsymbol{x}$ | More accurate locally | $K_a$ factorizations | Grouped by route | Routing errors, discontinuous boundaries, or unseen clusters |
| Library indexed by known discrete modes | Interpretable within modes | Several factors per mode | Strong within modes | Incorrect mode identification or unseen modes |
| Matrix adapted to each instance | Highly adaptive | May require a factorization per instance | Moderate to weak | Less factor reuse. Added construction cost at inference |
| True Jacobian for every sample | May be exact to first order at labeled points | About $N$ factorizations | Weakest | Implicit computation and storage costs move offline. No natural routing rule at inference |
| Approximation using physical structure | Varies with structure | Sparse and reusable | Strong | Systematic derivative bias or incorrect modes |

The primary method favors one or a few matrices with frozen routing based only on $\boldsymbol{x}$ and known modes. Hard routing can be discontinuous across partitions of $\boldsymbol{x}$, but the method selects one matrix for a given sample, so its training derivative with respect to $\boldsymbol{u}$ contains no routing term. An extension using a soft mixture $\sum_k \pi_k(\boldsymbol{x}, \boldsymbol{u})\boldsymbol{A}_k^{-1}\boldsymbol{b}_k(\boldsymbol{u}; \boldsymbol{x})$ that depends on the outputs must include $\boldsymbol{J}_{\pi_k,u}$ in the sensitivity. Hard routing based on outputs instead gives a piecewise discontinuous map. Neither is part of the default setting of the main theorems.

The Euclidean mean or median of training labeled points is only a candidate starting point. It may be infeasible, cross solution branches, or have poor conditioning. Even after projection onto the feasible set, the selected solution branch must be recorded. Continuation anchors along a continuous operating path can improve solution branch consistency. If anchors, matrices, or routing continue to be learned during formal training, each change may require updating the factors and sample calibration terms. Both the reuse of frozen linear systems and the sensitivity analysis for a fixed map must then be justified anew.

**An Implementable Selection Procedure.**

1. Use training samples $\mathcal{S}$ for candidate construction, with separate sets for checkpoint validation and final testing.
2. Use training data and physical units to freeze $\boldsymbol{W}_x, \boldsymbol{W}_u, \boldsymbol{W}_z, \boldsymbol{W}_h, \boldsymbol{W}_g$, thresholds for successful label generation, and mode and solution branch definitions. Specify how failed labels and labels with unknown solution branches are handled before constructing candidates.
3. Construct complete candidate rules: the matrix library, the $\boldsymbol{b}^{(k)}(\boldsymbol{u}; \boldsymbol{x})$ map for each matrix, routing based only on information available at test time, numerical precision, solver backend, and fallback for unseen modes.
4. Perform symbolic and numerical factorizations for all candidate matrices, recording condition estimates, nonnormality diagnostics, new nonzeros from factorization, factor memory, routing cost, and time for batched triangular solves.
5. On $\mathcal{S}$, compute reference roots to high accuracy on the same solution branch, sensitivity probes, or the inexpensive approximations described below. Freeze scoring weights, clipping thresholds, failure penalties, number of random probes, candidate count, and rule for resolving ties.
6. Evaluate the prespecified bounded scores on $\mathcal{S}$: dimensionless sensitivity or task gradient error, initialization error of the label-independent surrogate, routing and mode errors, completion success with warm starts, and cost.
7. Freeze the scalar score and rule for resolving ties before selecting a candidate. Do not retune weights, cluster counts, routing features, or Pareto choices after scoring.
8. Only after fixing the final candidate rule and numerical factors should sample calibration terms be computed for the full training set. Matrices must not be reselected using subsequent training or test performance while still being described as frozen in advance.
9. Final testing uses only the frozen routing and warm starts from the label-independent surrogate. Any rebuilding, rejection, or fallback must be timed and counted under preregistered rules.

The candidate scores above are offline diagnostics for selecting an implementation rule. The primary method uses a single frozen candidate, with sample calibration for training and no calibration at inference.

**Inexpensive Sensitivity Probes and Tangent Residuals.**

The full exact sensitivity matrix need not be formed. For each probed sample, obtain an accurate root $\boldsymbol{z}_i^{\mathrm{ref}} = \boldsymbol{z}_h(\boldsymbol{u}_i^\star; \boldsymbol{x}_i)$ on the same solution branch. Set $\boldsymbol{z}_i^{\mathrm{ref}} = \boldsymbol{z}_i^\star$ if the label meets the diagnostic tolerance. Otherwise, recomplete and record the solution branch. For a Rademacher random vector $\boldsymbol{\nu}$, solve

$$\boldsymbol{J}_{h,z}(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^{\mathrm{ref}}; \boldsymbol{x}_i)\boldsymbol{w} = -\boldsymbol{J}_{h,u}(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^{\mathrm{ref}}; \boldsymbol{x}_i)\boldsymbol{W}_{u,i}^{-1}\boldsymbol{\nu}, \tag{B.14}$$

to obtain the exact directional sensitivity $\boldsymbol{w} = \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i)\boldsymbol{W}_{u,i}^{-1}\boldsymbol{\nu}$. The corresponding sample-calibrated linear surrogate response in the same direction is $\boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star)\boldsymbol{W}_{u,i}^{-1}\boldsymbol{\nu}$. When $\mathbb{E}[\boldsymbol{\nu}\boldsymbol{\nu}^\top] = \boldsymbol{I}$, the Hutchinson identity gives

$$\mathbb{E}_{\boldsymbol{\nu}} \left\| \boldsymbol{W}_{z,i} \left[ \boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i) \right] \boldsymbol{W}_{u,i}^{-1}\boldsymbol{\nu} \right\|^2 = \left\| \boldsymbol{W}_{z,i} \left[ \boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i) \right] \boldsymbol{W}_{u,i}^{-1} \right\|_F^2. \tag{B.15}$$

A few random directions therefore provide an unbiased estimate of the weighted Frobenius objective in (B.1). The associated linear systems depend on the state and are solved only on a small offline selection subset, not during formal training.

Tangent residuals can be used to avoid even exact directional sensitivities. The operator norms below are induced by the frozen scales for $\boldsymbol{u}$, $\boldsymbol{z}$, and equality residuals:

$$\boldsymbol{R}_{a,i} = \boldsymbol{J}_{h,u}(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^{\mathrm{ref}}; \boldsymbol{x}_i) + \boldsymbol{J}_{h,z}(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^{\mathrm{ref}}; \boldsymbol{x}_i)\boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star). \tag{B.16}$$

The exact tangent identity $\boldsymbol{J}_{h,u} + \boldsymbol{J}_{h,z}\boldsymbol{J}_{z_h,u} = 0$ gives

$$\boldsymbol{R}_{a,i} = \boldsymbol{J}_{h,z}(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^{\mathrm{ref}}; \boldsymbol{x}_i) \left[ \boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i) \right], \tag{B.16a}$$

Therefore,

$$\|\boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star) - \boldsymbol{J}_{z_h,u}(\boldsymbol{u}_i^\star;\boldsymbol{x}_i)\| \le \|\boldsymbol{J}_{h,z}(\boldsymbol{u}_i^\star,\boldsymbol{z}_i^{\text{ref}};\boldsymbol{x}_i)^{-1}\|\|\boldsymbol{R}_{a,i}\|. \tag{B.17}$$

Directional tangent residuals can also be approximated using only nonlinear function evaluations. For the direction $\boldsymbol{\eta}_i = \boldsymbol{W}_{u,i}^{-1}\boldsymbol{\nu}$ in physical coordinates,

$$\frac{\boldsymbol{h}(\boldsymbol{u}_i^\star + \varepsilon\boldsymbol{\eta}_i, \boldsymbol{z}_i^{\text{ref}} + \varepsilon\boldsymbol{J}_{\tilde{z}_{a,i},u}(\boldsymbol{u}_i^\star)\boldsymbol{\eta}_i;\boldsymbol{x}_i) - \boldsymbol{h}(\boldsymbol{u}_i^\star,\boldsymbol{z}_i^{\text{ref}};\boldsymbol{x}_i)}{\varepsilon} = \boldsymbol{R}_{a,i}\boldsymbol{\eta}_i + O(\varepsilon). \tag{B.18}$$

This approximation requires no additional nonlinear completion, but equation scaling and amplification by $\boldsymbol{J}_{h,z}^{-1}$ must both be considered. These probes are used only as offline diagnostics and do not enter formal training. Step sizes for finite differences must balance truncation and rounding errors. Samples with failed probe solves, unknown reference solution branches associated with the labeled points, or switches in the active mode must receive the preregistered failure penalty in the score.

# C Problem Formulations and Reduced Equations

This appendix presents the underlying problems, variable partitions, reduced equations, and label-independent surrogates $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})$ for AC optimal power flow, optimization of water and gas networks in steady state, and the controlled CSCP construction. Each label-independent surrogate begins with a first-order expansion. The original objectives and constraints remain in use for training and deployment evaluation. For each system and fixed mode, the sample-calibrated linear surrogate and label-independent surrogate share frozen coefficients: $(\boldsymbol{A}_{\text{PF}}, \boldsymbol{b}_{x,\text{PF}})$ for power, $(\boldsymbol{A}_{\text{W}}, \boldsymbol{b}_{x,\text{W}})$ for water, and $(\boldsymbol{A}_{\text{G}}, \boldsymbol{b}_{x,\text{G}})$ for gas. Sharing means identical coefficients, anchors, constant terms in the $\boldsymbol{b}_x$ maps, variable ordering, and scaling, not merely identical matrix structure. All are frozen before training and retained at inference. The matrix $\boldsymbol{A}$ is factorized once before training. Its cached factors support forward and transposed gradient solves with the sample-calibrated linear surrogate during training, and warm starts from the label-independent surrogate at inference. The training form adds a sample calibration term, which is omitted at inference. The resulting initial state is then used to solve the original nonlinear equalities, without rebuilding the frozen coefficients. Appendix D gives the configurations and experimental diagnostics.

## C.1 AC Optimal Power Flow

Given network parameters and nodal active and reactive loads, AC optimal power flow jointly selects generator active outputs $\boldsymbol{P}^g$, reactive outputs $\boldsymbol{Q}^g$, voltage magnitudes $\boldsymbol{V}$, and phase angles $\boldsymbol{\vartheta}$ to minimize generation cost subject to power balance and operating limits:

$$\begin{aligned}
\min_{\boldsymbol{P}^g,\boldsymbol{Q}^g,\boldsymbol{V},\boldsymbol{\vartheta}} \quad & \sum_{j\in\mathcal{G}} c_j(P_j^g) \\
\text{s.t.} \quad & P_i^{g,\text{bus}} - P_i^d = S_{\text{base}} V_i \sum_j V_j(G_{ij}\cos\vartheta_{ij} + B_{ij}\sin\vartheta_{ij}), \quad \forall i, \\
& Q_i^{g,\text{bus}} - Q_i^d = S_{\text{base}} V_i \sum_j V_j(G_{ij}\sin\vartheta_{ij} - B_{ij}\cos\vartheta_{ij}), \quad \forall i, \\
& \boldsymbol{P}_{\min}^g \le \boldsymbol{P}^g \le \boldsymbol{P}_{\max}^g, \qquad \boldsymbol{Q}_{\min}^g \le \boldsymbol{Q}^g \le \boldsymbol{Q}_{\max}^g, \\
& \boldsymbol{V}_{\min} \le \boldsymbol{V} \le \boldsymbol{V}_{\max}, \qquad \vartheta_{\text{ref}} = 0, \\
& |S_e^{\text{f}}(\boldsymbol{V},\boldsymbol{\vartheta})|^2 \le (S_e^{\max})^2, \quad |S_e^{\text{t}}(\boldsymbol{V},\boldsymbol{\vartheta})|^2 \le (S_e^{\max})^2, \quad e \in \mathcal{E}_S, \\
& \vartheta_e^{\min} \le \vartheta_{f(e)} - \vartheta_{t(e)} \le \vartheta_e^{\max}, \quad e \in \mathcal{E}_\vartheta.
\end{aligned} \tag{C.1}$$

Here, $\mathcal{G}$ is the generator set, $c_j$ is the prescribed generator cost function, and nodal generation aggregates generators at the same bus. The per-unit bus admittance matrix is $\boldsymbol{Y} = \boldsymbol{G} + \text{j}\boldsymbol{B}$, $\vartheta_{ij} = \vartheta_i - \vartheta_j$, and $S_{\text{base}}$ is the power base. The sets $\mathcal{E}_S$ and $\mathcal{E}_\vartheta$ contain branches with enabled capacity and limits on phase angle differences, respectively. $f(e), t(e)$ are the endpoint buses. All bounds come from the corresponding case. Writing the complex voltage as $v_i^{\text{ac}} = V_i \exp(\text{j}\vartheta_i)$, the complex powers at the two ends follow directly from the branch admittances:

$$\begin{aligned}
S_e^{\text{f}} &= S_{\text{base}} v_{f(e)}^{\text{ac}} \overline{Y_e^{\text{ff}} v_{f(e)}^{\text{ac}} + Y_e^{\text{ft}} v_{t(e)}^{\text{ac}}}, \\
S_e^{\text{t}} &= S_{\text{base}} v_{t(e)}^{\text{ac}} \overline{Y_e^{\text{tf}} v_{f(e)}^{\text{ac}} + Y_e^{\text{tt}} v_{t(e)}^{\text{ac}}}.
\end{aligned}$$

The fixed branch admittance coefficients $Y_e^{\mathrm{ff}}, Y_e^{\mathrm{ft}}, Y_e^{\mathrm{tf}}, Y_e^{\mathrm{tt}}$ include line charging, transformer tap ratios, and phase shifts. An overbar denotes complex conjugation. Coupling products of voltage magnitudes with trigonometric functions of angle differences makes the power balance equalities nonlinear and is a primary source of nonconvexity in this continuous optimization problem.

**Variable selection and reduced equalities.** The input $\boldsymbol{x}$ contains active and reactive demands at loaded buses and is expanded into load vectors covering all buses $(\boldsymbol{P}^d, \boldsymbol{Q}^d)$. Let $\mathcal{N}_{\mathrm{R}}$ be the nonreference buses, $\mathcal{N}_{\mathrm{PQ}}$ the PQ buses, $\mathcal{N}_{\mathrm{V}}$ the reference and PV buses, and $\mathcal{G}_{\mathcal{N}_{\mathrm{R}}}$ the generators at nonreference buses. Variables are ordered as follows:

$$\boldsymbol{u} = (\boldsymbol{P}^g_{\mathcal{G}_{\mathcal{N}_{\mathrm{R}}}}, \boldsymbol{V}_{\mathcal{N}_{\mathrm{V}}}), \qquad \boldsymbol{z} = (\boldsymbol{\vartheta}_{\mathcal{N}_{\mathrm{R}}}, \boldsymbol{V}_{\mathcal{N}_{\mathrm{PQ}}}). \tag{C.1a}$$

The network predicts active outputs of nonreference generators and voltage magnitudes at controlled buses. Power balance recovers the remaining bus angles and voltage magnitudes at PQ buses. The voltage magnitude at the reference bus is a control, while its angle is fixed to zero. Label angles are first shifted by the reference angle, removing the global angle shift degree of freedom. Case14, Case57, and Case118 have 9, 13, and 107 controls and 22, 106, and 181 state variables, respectively, giving $(p, m) = (9, 22), (13, 106), (107, 181)$ in Table D1. Generator active power is measured in MW, angles in radians, and voltages in per unit.

Divide the active and reactive power balance expressions in Eq. (C.1) by $S_{\mathrm{base}}$, and denote them by $P_i(\boldsymbol{V}, \boldsymbol{\vartheta})$ and $Q_i(\boldsymbol{V}, \boldsymbol{\vartheta})$, respectively. The corresponding per-unit net injections are $\boldsymbol{p}^{\mathrm{sp}} = (\boldsymbol{P}^{g,\mathrm{bus}} - \boldsymbol{P}^d)/S_{\mathrm{base}}$ and $\boldsymbol{q}^{\mathrm{sp}} = (\boldsymbol{Q}^{g,\mathrm{bus}} - \boldsymbol{Q}^d)/S_{\mathrm{base}}$. The completion residual, with signs matching the implementation, is

$$\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}) = \begin{bmatrix} \boldsymbol{P}_{\mathcal{N}_{\mathrm{R}}}(\boldsymbol{V}, \boldsymbol{\vartheta}) - \boldsymbol{p}^{\mathrm{sp}}_{\mathcal{N}_{\mathrm{R}}} \\ \boldsymbol{Q}_{\mathcal{N}_{\mathrm{PQ}}}(\boldsymbol{V}, \boldsymbol{\vartheta}) - \boldsymbol{q}^{\mathrm{sp}}_{\mathcal{N}_{\mathrm{PQ}}} \end{bmatrix} = \boldsymbol{0}. \tag{C.1b}$$

Here, $\boldsymbol{P}^{g,\mathrm{bus}}$ aggregates active generation at each bus, and $\boldsymbol{Q}^{g,\mathrm{bus}}_{\mathcal{N}_{\mathrm{PQ}}}$ is the prescribed reactive generation at PQ buses. The system has $|\mathcal{N}_{\mathrm{R}}| + |\mathcal{N}_{\mathrm{PQ}}| = m$ equations. After completion, total active generation at the reference bus and total reactive generation at controlled buses are obtained by adding loads to nodal injections and are not included again in $\boldsymbol{z}$. Active generation at the reference bus is allocated among generators using fixed proportions based on their capacity intervals. Other active outputs come directly from $\boldsymbol{u}$.

**Label-independent surrogate from a Taylor expansion.** Following the first-order construction in Eq. (3), expand the injection equations at the flat reference state $(\boldsymbol{V}, \boldsymbol{\vartheta}) = (\boldsymbol{1}, \boldsymbol{0})$, with unit per-unit voltage magnitudes and zero phase angles, and then simplify the coefficients as described below. A subscript 0 denotes evaluation of a partial derivative at that state. Define $\boldsymbol{D}_G = \mathrm{diag}(\boldsymbol{G1})$, $\boldsymbol{D}_B = \mathrm{diag}(\boldsymbol{B1})$, and $\boldsymbol{B}_P = -\boldsymbol{B} + \boldsymbol{D}_B$. Then

$$\begin{aligned} \boldsymbol{P}(\boldsymbol{V}, \boldsymbol{\vartheta}) &\approx \boldsymbol{P}(\boldsymbol{1}, \boldsymbol{0}) + (\partial_{\boldsymbol{V}} \boldsymbol{P})_0(\boldsymbol{V} - \boldsymbol{1}) + (\partial_{\boldsymbol{\vartheta}} \boldsymbol{P})_0 \boldsymbol{\vartheta} \\ &= \boldsymbol{B}_P \boldsymbol{\vartheta} + \boldsymbol{G}\boldsymbol{V} + \boldsymbol{D}_G(\boldsymbol{V} - \boldsymbol{1}), \\ \boldsymbol{Q}(\boldsymbol{V}, \boldsymbol{\vartheta}) &\approx \boldsymbol{Q}(\boldsymbol{1}, \boldsymbol{0}) + (\partial_{\boldsymbol{V}} \boldsymbol{Q})_0(\boldsymbol{V} - \boldsymbol{1}) + (\partial_{\boldsymbol{\vartheta}} \boldsymbol{Q})_0 \boldsymbol{\vartheta} \\ &= -\boldsymbol{G}\boldsymbol{\vartheta} - \boldsymbol{B}\boldsymbol{V} + \boldsymbol{D}_G \boldsymbol{\vartheta} - \boldsymbol{D}_B(\boldsymbol{V} - \boldsymbol{1}). \end{aligned} \tag{C.1c}$$

The label-independent surrogate omits the corrections based on admittance row sums $\boldsymbol{D}_G(\boldsymbol{V} - \boldsymbol{1})$ and $\boldsymbol{D}_G \boldsymbol{\vartheta} - \boldsymbol{D}_B(\boldsymbol{V} - \boldsymbol{1})$, yielding the Taylor model with simplified coefficients $\boldsymbol{P} \approx \boldsymbol{B}_P \boldsymbol{\vartheta} + \boldsymbol{G}\boldsymbol{V}$ and $\boldsymbol{Q} \approx -\boldsymbol{G}\boldsymbol{\vartheta} - \boldsymbol{B}\boldsymbol{V}$. Partitioning variables as in Eq. (C.1a) and moving known controls to the right gives the label-independent surrogate below, with coefficients frozen for both forms and subscripts denoting the corresponding row and column blocks:

$$\underbrace{\begin{bmatrix} (\boldsymbol{B}_P)_{\mathcal{N}_{\mathrm{R}}\mathcal{N}_{\mathrm{R}}} & \boldsymbol{G}_{\mathcal{N}_{\mathrm{R}}\mathcal{N}_{\mathrm{PQ}}} \\ -\boldsymbol{G}_{\mathcal{N}_{\mathrm{PQ}}\mathcal{N}_{\mathrm{R}}} & -\boldsymbol{B}_{\mathcal{N}_{\mathrm{PQ}}\mathcal{N}_{\mathrm{PQ}}} \end{bmatrix}}_{\boldsymbol{A}_{\mathrm{PF}}} \begin{bmatrix} \boldsymbol{\vartheta}_{\mathcal{N}_{\mathrm{R}}} \\ \boldsymbol{V}_{\mathcal{N}_{\mathrm{PQ}}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{p}^{\mathrm{sp}}_{\mathcal{N}_{\mathrm{R}}} - \boldsymbol{G}_{\mathcal{N}_{\mathrm{R}}\mathcal{N}_{\mathrm{V}}} \boldsymbol{V}_{\mathcal{N}_{\mathrm{V}}} \\ \boldsymbol{q}^{\mathrm{sp}}_{\mathcal{N}_{\mathrm{PQ}}} + \boldsymbol{B}_{\mathcal{N}_{\mathrm{PQ}}\mathcal{N}_{\mathrm{V}}} \boldsymbol{V}_{\mathcal{N}_{\mathrm{V}}} \end{bmatrix}}_{\boldsymbol{b}_{x,\mathrm{PF}}(\boldsymbol{u})}. \tag{C.1d}$$

The matrix $\boldsymbol{A}_{\mathrm{PF}}$ depends only on fixed network parameters and variable sets and retains coupling between active and reactive powers, angles, and magnitudes. Its simplified coefficients generally differ from those of the exact Jacobian of the original AC residual. Let $\boldsymbol{E}_g$ aggregate nonreference generator active outputs at their corresponding buses in $\mathcal{N}_{\mathrm{R}}$. The control sensitivity and recovered state of the sample-calibrated linear surrogate are then

$$\boldsymbol{K}_{\mathrm{PF}} = \boldsymbol{A}_{\mathrm{PF}}^{-1} \begin{bmatrix} \boldsymbol{E}_g / S_{\mathrm{base}} & -\boldsymbol{G}_{\mathcal{N}_{\mathrm{R}}\mathcal{N}_{\mathrm{V}}} \\ \boldsymbol{0} & \boldsymbol{B}_{\mathcal{N}_{\mathrm{PQ}}\mathcal{N}_{\mathrm{V}}} \end{bmatrix}, \qquad \tilde{\boldsymbol{z}}_i = \boldsymbol{z}_i^\star + \boldsymbol{K}_{\mathrm{PF}}(\boldsymbol{u}_i - \boldsymbol{u}_i^\star). \tag{C.1e}$$

This is exactly the sample calibration form defined by $\boldsymbol{r}_i = \boldsymbol{A}_{\text{PF}}\boldsymbol{z}_i^\star - \boldsymbol{b}_{x_i,\text{PF}}(\boldsymbol{u}_i^\star)$, with $\boldsymbol{J}_{\tilde{z}_i,u} = \boldsymbol{K}_{\text{PF}} = \boldsymbol{A}_{\text{PF}}^{-1}\boldsymbol{J}_{b_{x_i,\text{PF}},u}(\boldsymbol{u})$. Calibration adds only a constant offset to the label-independent surrogate, leaving its control sensitivity unchanged.

**Warm-started inference with the label-independent surrogate.** Inference reuses $\boldsymbol{A}_{\text{PF}}$, $\boldsymbol{b}_{x,\text{PF}}$, and the factorization from Eq. (C.1d), removing only $\boldsymbol{r}_i$ to obtain

$$\boldsymbol{z}^{\text{init}} = \boldsymbol{A}_{\text{PF}}^{-1}\boldsymbol{b}_{x,\text{PF}}(\hat{\boldsymbol{u}}), \qquad \hat{\boldsymbol{z}} = \text{Root}_h(\hat{\boldsymbol{u}}; \boldsymbol{x}, \boldsymbol{z}^{\text{init}}). \tag{C.1f}$$

The initialization depends only on the instance and predicted controls. It uses no test labels and does not introduce a new Taylor expansion at inference. It then initializes Newton completion of the original AC equalities. The Jacobians evaluated at the current states during those iterations belong to the nonlinear completion solver. They neither replace nor redefine $\boldsymbol{A}_{\text{PF}}$.

### C.2 Water Distribution Networks in Steady State

Given nodal demands, boundary heads, and equipment modes, optimization of water distribution in steady state adjusts pump speeds to reduce operating cost while determining nodal total heads and pipe flows. With pump speed controls $\boldsymbol{u}$, unknown nodal total heads $\boldsymbol{p}$, and link flows $\boldsymbol{q}$, the problem is

$$\begin{aligned}
\min_{\boldsymbol{u},\boldsymbol{p},\boldsymbol{q}} \quad & F^{\text{W}}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q};\boldsymbol{x}) = W_{\text{pump}}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q})/W_{\text{ref}} + J_{\text{op}}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q}) \\
\text{s.t.} \quad & \boldsymbol{x} + \boldsymbol{C}_r\boldsymbol{q} = \boldsymbol{0}, \qquad (\boldsymbol{u},\boldsymbol{p},\boldsymbol{q}) \in \mathcal{Y}_{\text{W}}, \\
& \Delta h_e - r_e q_e |q_e|^{0.852} - r_e^{\text{minor}} q_e |q_e| = 0, \quad e \in \mathcal{E}_{\text{res}}, \\
& \Delta h_e + a_e s_e^2 - b_e s_e^{2-\nu_e} q_e^{\nu_e} = 0, \quad q_e > 0, \quad e \in \mathcal{E}_{\text{sp}}, \\
& \Delta h_e + W_e/(\rho_{\text{w}} g_{\text{grav}} q_e) = 0, \quad q_e > 0, \quad e \in \mathcal{E}_{\text{pw}}, \\
& q_e = 0, \quad e \in \mathcal{E}_{\text{cl}}, \\
& p_{\text{down}(e)} - e_{\text{down}(e)} - \bar{P}_e = 0, \quad e \in \mathcal{E}_{\text{prv}}.
\end{aligned} \tag{C.2}$$

The sets $\mathcal{E}_{\text{res}}$, $\mathcal{E}_{\text{sp}}$, $\mathcal{E}_{\text{pw}}$, $\mathcal{E}_{\text{cl}}$, and $\mathcal{E}_{\text{prv}}$ denote open pipes and valves governed by resistance relations, operating pumps with variable speed, operating pumps at constant power, closed equipment, and valves that reduce pressure and operate at constant pressure, respectively. These sets are determined by the prescribed equipment modes. Here, $s_e = s_e(\boldsymbol{u})$ is pump speed and $\Delta h_e$ the head difference along the reference orientation. The equations show pipe resistance and pump curves in their ordinary flow ranges. Smoothing or continuation near zero flow and the remaining coefficients are described below. The term $W_{\text{pump}}$ is pump power, $W_{\text{ref}} > 0$ a fixed normalization base, and $J_{\text{op}}$ collects additional operating costs specific to each case: a penalty for pressure deficits for Net1, tracking of tank levels at the next step for C-Town, and pressure deficits, tank net inflow tracking, and pump speed deviation penalties for Net6. The set $\mathcal{Y}_{\text{W}}$ includes pump speed bounds, equipment restrictions on flow direction, and mode consistency. C-Town and Net6 also explicitly constrain service pressure and predicted tank levels, whereas Net1 expresses its pressure requirements through the objective penalty. Tank levels at the next step are estimated from current net inflows in steady state, without introducing multiperiod scheduling variables. The problem couples linear flow conservation with pipe resistance governed by fractional powers and pump head gains that depend on speed. We next organize these physical equalities into a completion system.

**Variable selection and equipment controls.** The input consists of nodal demand features. Here, $\boldsymbol{x}$ denotes the reconstructed full nodal demand vector, with zeros inserted at nodes whose demands are fixed at zero. Controls $\boldsymbol{u}$ are pump speed ratios relative to rated speed. WNTR-Net1 controls one pump. C-Town uses five controls: PU1 and PU2 share one speed, while PU4, PU7, PU8, and PU10 each have their own. Net6 groups operating pumps that have variable speed and share the same start and end nodes into twenty controls for pump stations. Settings of pumps operating at constant power, valve settings, equipment activation states, and tank and reservoir boundary heads are not learned controls in these experiments. If $s_e(\boldsymbol{u})$ is the speed assigned to pump $e$ by this grouping, each pump's speed is extracted from $\boldsymbol{u}$ by a fixed selection matrix.

The state variables are $\boldsymbol{z} = (\boldsymbol{p},\boldsymbol{q})$, ordered with unknown nodal total heads first and flows of all retained links second. Boundaries with fixed heads are not included again in $\boldsymbol{p}$. Flows through closed equipment remain variables constrained by equations enforcing zero flow. Net1, C-Town, and Net6 have 22, 832, and 7,215 state variables, respectively, giving $(p, m) = (1, 22), (5, 832), (20, 7{,}215)$.

This partition lets the network predict only a small number of equipment control variables, while hydraulic equations recover heads and network transport states, including reverse flow. Heads and flows are measured in m and $m^3/s$, respectively.

**Hydraulic equations under fixed modes.** Let $\boldsymbol{C}_r \in \mathbb{R}^{n_h \times n_e}$ be the node–edge incidence matrix with rows for boundaries with fixed heads removed, using $+1$ at an edge's start and $-1$ at its end. For an ordinary open link, $\Delta h_e = (\boldsymbol{C}_r^\top \boldsymbol{p})_e + \beta_e$, where $\beta_e$ includes fixed boundary heads. To derive the Jacobian and label-independent surrogate, collect the equipment equalities from Eq. (C.2) rowwise as $\boldsymbol{L}\boldsymbol{p} + \boldsymbol{c}(\boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{0}$, where $\boldsymbol{c}$ contains the remaining terms independent of unknown heads. The complete system is

$$\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}) = \begin{bmatrix} \boldsymbol{x} + \boldsymbol{C}_r \boldsymbol{q} \\ \boldsymbol{L}\boldsymbol{p} + \boldsymbol{c}(\boldsymbol{q}, \boldsymbol{u}) \end{bmatrix} = \boldsymbol{0}, \qquad m = n_h + n_e. \tag{C.2a}$$

The matrix $\boldsymbol{L}$ is determined by equipment modes: ordinary open links use their rows for head differences, closed equipment uses zero rows, and valves that maintain constant downstream pressure use rows that select downstream heads. The vector $\boldsymbol{c}$ includes boundary constants and the following constitutive terms. An open pipe satisfies $\Delta h_e - \ell_e(q_e) = 0$, with loss in the ordinary flow range $\ell_e(q_e) = r_e q_e |q_e|^{0.852} + r_e^{\text{minor}} q_e |q_e|$. Net1 uses the unsmoothed Hazen–Williams relation. C-Town and Net6 use a linear segment and cubic transition near zero flow and recover the $1.852$-power relation at larger flows. An operating pump with variable speed satisfies

$$\Delta h_e + \gamma_e(q_e, s_e) = 0, \qquad \gamma_e(q_e, s_e) = a_e s_e^2 - b_e s_e^{2-\nu_e} q_e^{\nu_e}, \qquad q_e > 0, \tag{C.2b}$$

Here, $a_e, b_e, \nu_e$ are fixed pump curve parameters, and $\rho_{\text{w}}$ and $g_{\text{grav}}$ are water density and gravitational acceleration. At very low flows, C-Town and Net6 use a continuation consistent with speed scaling. Net6's operating pumps at constant power satisfy $\Delta h_e + W_e/(\rho_{\text{w}} g_{\text{grav}} q_e) = 0$. Closed pipes, pumps, and valves all satisfy $q_e = 0$. A valve that reduces pressure and operates in constant pressure mode satisfies $p_{\text{down}(e)} - e_{\text{down}(e)} - \bar{P}_e = 0$, where $e_{\text{down}(e)}$ is downstream elevation and $\bar{P}_e$ the fixed pressure head setting. C-Town's open valves for flow control are modeled using local resistance, without forcing flow to equal the setpoint. Thus, valves and closed equipment cannot all be represented by one common equation for pipe head loss.

**The frozen label-independent surrogate.** At the selected anchor $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{z}})$, write $\bar{\boldsymbol{c}} = \boldsymbol{c}(\bar{\boldsymbol{q}}, \bar{\boldsymbol{u}})$, $\bar{\boldsymbol{D}} = \partial_{\boldsymbol{q}} \boldsymbol{c}$, and $\bar{\boldsymbol{M}} = \partial_{\boldsymbol{u}} \boldsymbol{c}$. Each link's constitutive relation depends only on its own flow, so $\bar{\boldsymbol{D}}$ is diagonal. Nonzero entries of $\bar{\boldsymbol{M}}$ occur only in rows for controlled operating pumps. Pumps sharing a speed use the same control column. Freezing these derivatives while retaining exact linear conservation gives

$$\underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{C}_r \\ \boldsymbol{L} & \bar{\boldsymbol{D}} \end{bmatrix}}_{\boldsymbol{A}_{\text{W}}} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix} = \underbrace{\begin{bmatrix} -\boldsymbol{x} \\ \bar{\boldsymbol{D}}\bar{\boldsymbol{q}} - \bar{\boldsymbol{c}} - \bar{\boldsymbol{M}}(\boldsymbol{u} - \bar{\boldsymbol{u}}) \end{bmatrix}}_{\boldsymbol{b}_{x,\text{W}}(\boldsymbol{u})}. \tag{C.2c}$$

The diagonal entries are $-\ell_e'(\bar{q}_e)$ for open pipes, $\partial_{q_e} \gamma_e$ for operating pumps with variable speed, $-W_e/(\rho_{\text{w}} g_{\text{grav}} \bar{q}_e^2)$ for operating pumps at constant power, $1$ for closed equipment, and $0$ for valves that maintain constant downstream pressure. In the ordinary flow range of Eq. (C.2b), $\partial_{s_e} \gamma_e = 2a_e s_e - b_e(2 - \nu_e) s_e^{1-\nu_e} q_e^{\nu_e}$. Multiplication by the fixed pump speed grouping matrix constructs $\bar{\boldsymbol{M}}$. This block structure gives both $\boldsymbol{A}_{\text{W}} = \bar{\boldsymbol{J}}_{h,z}$ and $\bar{\boldsymbol{J}}_{h,u} = (\boldsymbol{0}^\top, \bar{\boldsymbol{M}}^\top)^\top$, rather than approximating every device by a nonzero pipe resistance.

The sample-calibrated linear surrogate and label-independent surrogate share $(\boldsymbol{A}_{\text{W}}, \boldsymbol{b}_{x,\text{W}})$ from Eq. (C.2c). With the training calibration term $\boldsymbol{r}_i = \boldsymbol{A}_{\text{W}} \boldsymbol{z}_i^\star - \boldsymbol{b}_{x_i,\text{W}}(\boldsymbol{u}_i^\star)$, the common control sensitivity, sample-calibrated linear surrogate state, and label-independent surrogate initialization are

$$\boldsymbol{K}_{\text{W}} = -\boldsymbol{A}_{\text{W}}^{-1} \begin{bmatrix} \boldsymbol{0} \\ \bar{\boldsymbol{M}} \end{bmatrix}, \qquad \tilde{\boldsymbol{z}}_i = \boldsymbol{z}_i^\star + \boldsymbol{K}_{\text{W}}(\boldsymbol{u}_i - \boldsymbol{u}_i^\star), \qquad \boldsymbol{z}^{\text{init}} = \boldsymbol{A}_{\text{W}}^{-1} \boldsymbol{b}_{x,\text{W}}(\hat{\boldsymbol{u}}). \tag{C.2d}$$

Calibration removes the state offset at the labeled point without changing $\boldsymbol{K}_{\text{W}} = \boldsymbol{J}_{\tilde{z}_i,u} = \boldsymbol{A}_{\text{W}}^{-1} \boldsymbol{J}_{b_{x_i,\text{W}},u}(\boldsymbol{u})$. Both forms share the frozen anchor, equipment modes, matrix coefficients, and constant terms in the $\boldsymbol{b}_x$ maps. Omitting only $\boldsymbol{r}_i$ gives the label-independent surrogate, which initializes original nonlinear completion without changing the anchor or frozen coefficients. Open pipes permit reverse flow, whereas operating pumps require forward flow. The unsmoothed $1.852$-power pipe relation does not have a Lipschitz Jacobian at zero flow, so local results for quadratic error bounds must be restricted to regions with the required regularity.

### C.3 Gas Networks in Steady State

Given gas demands, network parameters, and equipment modes, optimization of gas networks in steady state jointly selects source supplies $\boldsymbol{q}^{\text{src}}$, link flows $\boldsymbol{q}$, and squared nodal pressures $\boldsymbol{\pi}$, where $\pi_v = P_v^2$, to satisfy demand while reducing supply and transport costs:

$$\begin{aligned}
\min_{\boldsymbol{q}^{\text{src}},\boldsymbol{q},\boldsymbol{\pi}} \quad & F^{\text{G}}(\boldsymbol{q}^{\text{src}},\boldsymbol{q},\boldsymbol{\pi};\boldsymbol{x}) = \sum_{j=1}^{n_s} \alpha_j (q_j^{\text{src}})^2 + J_{\text{G}}(\boldsymbol{q},\boldsymbol{\pi}) \\
\text{s.t.} \quad & \boldsymbol{C}\boldsymbol{q} = \boldsymbol{E}_{\text{src}}\boldsymbol{q}^{\text{src}} - \boldsymbol{E}_{\text{d}}\boldsymbol{x}, \\
& (\boldsymbol{C}^\top \boldsymbol{\pi})_e = k_e q_e \sqrt{q_e^2 + \varepsilon_e^2}, \quad e \in \mathcal{E}^{\text{G}}_{\text{res}}, \\
& (\boldsymbol{C}^\top \boldsymbol{\pi})_e = 0, \quad e \in \mathcal{E}^{\text{G}}_{\text{eq}}, \\
& \boldsymbol{q}^{\text{src}}_{\min} \le \boldsymbol{q}^{\text{src}} \le \boldsymbol{q}^{\text{src}}_{\max}, \\
& \boldsymbol{P}^2_{\min} \le \boldsymbol{\pi} \le \boldsymbol{P}^2_{\max}, \qquad \boldsymbol{q}_{\min} \le \boldsymbol{q} \le \boldsymbol{q}_{\max}.
\end{aligned} \tag{C.3}$$

Here, $\boldsymbol{C}$ is the node–edge incidence matrix, with $+1$ at an edge's start and $-1$ at its end. $\boldsymbol{E}_{\text{src}}$ and $\boldsymbol{E}_{\text{d}}$ map source supplies and demands to their corresponding nodes. The sets $\mathcal{E}^{\text{G}}_{\text{res}}$ and $\mathcal{E}^{\text{G}}_{\text{eq}}$ contain resistance links and links with zero pressure drop, such as open valves and bypassed compressors, respectively. Resistance links use $\varepsilon_e = 0$ in GasLib-11 and GasLib-135, recovering $k_e q_e |q_e|$, and $\varepsilon_e = 0.1$ in GasLib-4197. Node contraction, equivalent resistance coefficients, and fixed equipment modes are detailed below. Supply, pressure, and flow limits follow the corresponding cases. Pressure bounds are positive, and squares and inequalities are interpreted componentwise.

The number of sources is $n_s$, and $\alpha_j > 0$ is a supply cost coefficient incorporating fixed normalization. For GasLib-11 and GasLib-135, $J_{\text{G}}$ is a weighted mean of squared pressures that discourages unnecessarily high pressures. GasLib-4197 instead uses the pipeline transport cost $\sum_e \beta_e |q_e|^3$, where $\beta_e \ge 0$ incorporates resistance coefficients, weights, and normalization and is zero for non-pipe edges. Equipment activation decisions and active compression ratios are not optimized, so this objective must not be interpreted as compressor energy minimization. Flow signs indicate transport direction relative to a prescribed orientation. Coupling differences in squared pressure with signed nonlinear flow terms gives the main nonlinear equalities. We first parameterize supply decisions and then derive the reduced completion equations.

**Variables and fixed equipment modes.** All three cases use continuous models in steady state with fixed topology and equipment modes. Compressor activation and valve switching are not optimized. The input $\boldsymbol{x}$ contains demands at nodes that consume gas. Controls are $\boldsymbol{u} = (a_1, \dots, a_{n_s-1}, P_{\text{ref}})$, where $n_s$ is the source count, $a_j \in [0,1]$ controls supply allocation, and $P_{\text{ref}}$ is the absolute reference node pressure. Set $w_j = a_j \prod_{k<j}(1-a_k)$ for $j < n_s$ and $w_{n_s} = \prod_{k<n_s}(1-a_k)$. Source supplies are then

$$\boldsymbol{q}^{\text{src}} = \boldsymbol{q}_0^{\text{src}} + \big(\mathbf{1}^\top \boldsymbol{x} - \mathbf{1}^\top \boldsymbol{q}_0^{\text{src}}\big)\boldsymbol{w}, \qquad \mathbf{1}^\top \boldsymbol{w} = 1. \tag{C.3a}$$

Here, $\boldsymbol{q}_0^{\text{src}}$ is a fixed supply baseline: $(50, 100, 0)^\top$ for GasLib-11 and zero for GasLib-135 and GasLib-4197. This parameterization makes total supply identically equal to total demand. Flow is measured in $10^3\,\text{Nm}^3/\text{h}$ and pressure in bar. GasLib-11 has an open valve and two bypassed compressors. All GasLib-135 compressors are bypassed. These devices impose equal pressures at their endpoints. In GasLib-4197, nodes connected by open valves, bypassed compressors, and other links with zero pressure drop are contracted, and resistance self-loops whose flow is necessarily zero after contraction are removed. This gives 3,302 groups of nodes with equal pressure and 3,539 retained edges. After completion, group pressures are mapped to the original nodes. Within each group, flows on edges with zero pressure drop are recovered from nodal conservation along a fixed spanning forest, while remaining such edge flows are set to zero to eliminate nonuniqueness of internal circulations.

**Conservation and pressure drop equations.** On this model graph, let $\boldsymbol{C} \in \mathbb{R}^{n_v \times n_e}$ be the node–edge incidence matrix, with $+1$ at each edge's start and $-1$ at its end, and let $\boldsymbol{C}_r$ omit the reference node row. The edge flows $\boldsymbol{q}$ use this orientation, so negative values indicate reverse transport. Let $\pi_v = P_v^2$ be squared nodal pressure and $\boldsymbol{s}_x(\boldsymbol{u})$ the nodal injection defined as supply minus demand, aggregating injections within contracted groups. Equation (C.3a) gives $\mathbf{1}^\top \boldsymbol{s}_x = 0$, so reference node conservation is redundant. The completion system is

$$\boldsymbol{C}_r \boldsymbol{q} = \boldsymbol{s}_{x,r}(\boldsymbol{u}), \qquad \boldsymbol{C}^\top \boldsymbol{\pi} = \boldsymbol{\ell}(\boldsymbol{q}), \qquad \pi_{\text{ref}} = P_{\text{ref}}^2. \tag{C.3b}$$

Pipes in GasLib-11 and GasLib-135 use $\ell_e(q_e) = k_e q_e|q_e|$, where $k_e > 0$ is determined by length, diameter, and frozen isothermal gas parameters. Devices with equal pressure at both endpoints use $k_e = 0$. All retained GasLib-4197 edges use $\ell_e(q_e) = k_e q_e \sqrt{q_e^2 + \varepsilon^2}$, with $\varepsilon = 0.1$, to smooth resistance near zero flow. Pipe coefficients still follow the isothermal relation for pipe flow. A local resistance $\Delta P = c_e q_e|q_e|$ is converted to a model in squared pressures using $k_e = 2P_{\text{nom}} c_e$. Six resistance elements with fixed pressure loss use $k_e = 2P_{\text{nom}} \Delta P_e / Q_{\text{nom}}^2$, where $P_{\text{nom}} = 51.01325$ and $Q_{\text{nom}} = 1{,}000$. These are frozen bidirectional smooth resistance approximations for this case, not the original device equations for fixed pressure loss. One additional resistance element is fixed to zero pressure drop. This large case also uses common pressure bounds across the network. It therefore tests a continuous problem with fixed modes constructed from the GasLib network, rather than the original mixed-integer equipment scheduling problem.

**Normalized states and Jacobians.** Write $\widehat{\boldsymbol{q}} = \boldsymbol{q}/Q_0$ and $\widehat{\boldsymbol{\pi}} = \boldsymbol{\pi}/P_0^2$. State variables are ordered with flows first and nonreference squared pressures second: $\boldsymbol{z} = (\widehat{\boldsymbol{q}}, \widehat{\boldsymbol{\pi}}_r)$. The three systems use $(Q_0, P_0) = (300, 50)$, $(4{,}000, 80)$, and $(15{,}000, 101.01325)$, respectively. Let $\widehat{\ell}_e(t) = \ell_e(Q_0 t)/P_0^2$ and $\boldsymbol{c}_{\text{ref}} = \boldsymbol{C}^\top \boldsymbol{e}_{\text{ref}}$, where $\boldsymbol{e}_{\text{ref}}$ is the reference node unit vector. The implementation uses the dimensionless residual

$$\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}) = \begin{bmatrix} \boldsymbol{C}_r \widehat{\boldsymbol{q}} - \boldsymbol{s}_{x,r}(\boldsymbol{u})/Q_0 \\ \boldsymbol{C}_r^\top \widehat{\boldsymbol{\pi}}_r + \boldsymbol{c}_{\text{ref}} P_{\text{ref}}^2/P_0^2 - \widehat{\boldsymbol{\ell}}(\widehat{\boldsymbol{q}}) \end{bmatrix} = \boldsymbol{0}. \tag{C.3c}$$

The three cases have dimensions of the demand inputs 3, 99, and 1,009, control dimensions 3, 6, and 11, and state dimensions 21, 304, and 6,840, respectively. Thus, $m = n_e + n_v - 1$ is both the equation count and state variable dimension, giving $(p, m) = (3, 21), (6, 304), (11, 6{,}840)$, consistent with Table D1. A small number of control variables first determine supply allocation and the pressure reference. Completion then recovers flows and pressures throughout the network. Let $\boldsymbol{D} = \operatorname{diag}(\widehat{\ell}_e'(\widehat{q}_e))$, and let $\boldsymbol{e}_p$ be the unit vector corresponding to the reference pressure control. Then

$$\boldsymbol{J}_{h,z} = \begin{bmatrix} \boldsymbol{C}_r & \boldsymbol{0} \\ -\boldsymbol{D} & \boldsymbol{C}_r^\top \end{bmatrix}, \qquad \boldsymbol{J}_{h,u} = \begin{bmatrix} -\partial_{\boldsymbol{u}} \boldsymbol{s}_{x,r}(\boldsymbol{u})/Q_0 \\ (2P_{\text{ref}}/P_0^2)\boldsymbol{c}_{\text{ref}} \boldsymbol{e}_p^\top \end{bmatrix}. \tag{C.3d}$$

The unsmoothed pipe relation has derivative $\ell_e'(q) = 2k_e|q|$, the smoothed relation has derivative $\ell_e'(q) = k_e(2q^2 + \varepsilon^2)/\sqrt{q^2 + \varepsilon^2}$, and $\widehat{\ell}_e'(t) = (Q_0/P_0^2)\ell_e'(Q_0 t)$. The former remains continuously differentiable at zero flow with a Lipschitz derivative, but zero flows and edges with zero pressure drop can affect nonsingularity of the full system. Differentiability of individual edge equations alone does not imply that $\boldsymbol{J}_{h,z}$ is invertible. Fixed modes permit flow reversal without switching equations according to flow signs.

**The frozen label-independent surrogate.** Select an anchor $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{z}})$ before training and freeze the Jacobian above. Also expand supply allocation and squared reference pressure to first order: write $\boldsymbol{s}_x^{\text{lin}}(\boldsymbol{u}) = \bar{\boldsymbol{s}} + (\partial_{\boldsymbol{x}} \boldsymbol{s})_{\text{anchor}}(\boldsymbol{x} - \bar{\boldsymbol{x}}) + (\partial_{\boldsymbol{u}} \boldsymbol{s})_{\text{anchor}}(\boldsymbol{u} - \bar{\boldsymbol{u}})$ and $\pi_{\text{ref}}^{\text{lin}} = \bar{P}_{\text{ref}}^2 + 2\bar{P}_{\text{ref}}(P_{\text{ref}} - \bar{P}_{\text{ref}})$. The label-independent surrogate uses an affine map through this anchor, with coefficients retained in the sample-calibrated linear surrogate. With anchor residual $\bar{\boldsymbol{h}} = \boldsymbol{h}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{z}}; \bar{\boldsymbol{x}})$, the corresponding system is

$$\underbrace{\begin{bmatrix} \boldsymbol{C}_r & \boldsymbol{0} \\ -\bar{\boldsymbol{D}} & \boldsymbol{C}_r^\top \end{bmatrix}}_{\boldsymbol{A}_{\text{G}}} \begin{bmatrix} \widehat{\boldsymbol{q}} \\ \widehat{\boldsymbol{\pi}}_r \end{bmatrix} = \underbrace{\left( \begin{bmatrix} \boldsymbol{s}_{x,r}^{\text{lin}}(\boldsymbol{u})/Q_0 \\ \widehat{\boldsymbol{\ell}}(\bar{\widehat{\boldsymbol{q}}}) - \bar{\boldsymbol{D}}\bar{\widehat{\boldsymbol{q}}} - \boldsymbol{c}_{\text{ref}} \pi_{\text{ref}}^{\text{lin}}/P_0^2 \end{bmatrix} + \bar{\boldsymbol{h}} \right)}_{\boldsymbol{b}_{x,\text{G}}(\boldsymbol{u})}. \tag{C.3e}$$

Here, $\bar{\boldsymbol{D}}$ is evaluated at the anchor flows, and $\boldsymbol{A}_{\text{G}} = \bar{\boldsymbol{J}}_{h,z}$. Equation (C.3e) is equivalent to $\boldsymbol{b}_{x,\text{G}} = \boldsymbol{A}_{\text{G}} \bar{\boldsymbol{z}} - \bar{\boldsymbol{J}}_{h,x}(\boldsymbol{x} - \bar{\boldsymbol{x}}) - \bar{\boldsymbol{J}}_{h,u}(\boldsymbol{u} - \bar{\boldsymbol{u}})$. A strictly feasible anchor has $\bar{\boldsymbol{h}} = \boldsymbol{0}$, recovering the first-order expansion of the original equalities. At finite solver tolerance, adding back the fixed anchor residual makes the label-independent surrogate pass through the anchor. This fixed term remains in the $\boldsymbol{b}_{x,\text{G}}(\boldsymbol{u})$ map shared by both forms, unlike the sample calibration term used only in training. The expansion coefficients for supply allocation and squared reference pressure are also shared and are not recomputed from predictions. Setting $\boldsymbol{r}_i = \boldsymbol{A}_{\text{G}} \boldsymbol{z}_i^\star - \boldsymbol{b}_{x_i,\text{G}}(\boldsymbol{u}_i^\star)$ gives the sample-calibrated linear surrogate state and label-independent surrogate initialization, respectively:

$$\boldsymbol{K}_{\text{G}} = -\boldsymbol{A}_{\text{G}}^{-1} \bar{\boldsymbol{J}}_{h,u}, \qquad \tilde{\boldsymbol{z}}_i = \boldsymbol{z}_i^\star + \boldsymbol{K}_{\text{G}}(\boldsymbol{u}_i - \boldsymbol{u}_i^\star), \qquad \boldsymbol{z}^{\text{init}} = \boldsymbol{A}_{\text{G}}^{-1} \boldsymbol{b}_{x,\text{G}}(\hat{\boldsymbol{u}}). \tag{C.3f}$$

Both forms share $(\boldsymbol{A}_{\text{G}}, \boldsymbol{b}_{x,\text{G}})$, with $\boldsymbol{J}_{\tilde{z}_i,u} = \boldsymbol{K}_{\text{G}} = \boldsymbol{A}_{\text{G}}^{-1} \boldsymbol{J}_{b_{x_i,\text{G}},u}(\boldsymbol{u})$. The label-independent surrogate omits only $\boldsymbol{r}_i$, retaining the anchor, matrix, and $\boldsymbol{b}_{x,\text{G}}(\boldsymbol{u})$ map. GasLib-11 uses this initialization

for damped Newton iterations. Larger systems first eliminate nodal conservation, perform Newton iterations in the space of cycle flows, and then recover squared nodal pressures. These are solution steps for the original nonlinear completion, not additional label-independent surrogates. The completed state is checked using the residual in Eq. (C.3c). Physical flows and pressures are then restored to check the original constraints. Successful equality completion alone does not guarantee that all pressure, supply, and flow bounds hold.

### C.4 The Controlled CSCP Construction

CSCP varies nonlinearity, matrix conditioning, and the structure of solution branches within one problem family, allowing their effects on training with the sample-calibrated linear surrogate and warm starts from the label-independent surrogate to be distinguished. Take fixed orthogonal matrices $\boldsymbol{O}, \boldsymbol{T} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Lambda} = \operatorname{diag}(1, \kappa, \ldots, \kappa)$, where $\kappa > 1$. The parameter vector $\boldsymbol{x} = (\boldsymbol{c}_0, \boldsymbol{c}_u, \boldsymbol{c}_z)$ collects the fixed instance coefficients in the parametric NLP

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{z}} \quad & \tfrac{1}{2}\|\boldsymbol{u}\|_2^2 + \boldsymbol{c}_u^\top \boldsymbol{u} + \tfrac{\rho}{2}\|\boldsymbol{z}\|_2^2 + \boldsymbol{c}_z^\top \boldsymbol{z}, \\ \text{s.t.} \quad & \boldsymbol{O}^\top \big[\boldsymbol{\Lambda}\boldsymbol{O}\boldsymbol{z} + \tfrac{\alpha}{\omega}\sin(\omega \boldsymbol{O}\boldsymbol{z})\big] - \boldsymbol{T}\boldsymbol{u} - \boldsymbol{c}_0 = \boldsymbol{0}, \qquad -\boldsymbol{1} \le \boldsymbol{u} \le \boldsymbol{1}. \end{aligned} \tag{C.4}$$

Here, $\rho > 0$ weights the state in the objective, $\alpha \ge 0$ controls nonlinearity strength, $\omega > 0$ controls oscillation frequency, and $\kappa$ determines the spectral scale of the linear component. The sine acts coordinatewise. These parameters define the problem family. Their values for the standard performance comparisons and boundary diagnostics are specified separately in the experimental settings.

**Variable partition and reduced completion equations.** The network predicts controls $\boldsymbol{u} \in [-1, 1]^m$, and the $m$ nonlinear equalities above recover state variables $\boldsymbol{z} \in \mathbb{R}^m$. Thus, $p = m$ in the notation of Eq. (1). Denote the nonlinear map acting on $\boldsymbol{z}$ by $G(\boldsymbol{z})$, so that $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{z}; \boldsymbol{x}) = G(\boldsymbol{z}) - \boldsymbol{T}\boldsymbol{u} - \boldsymbol{c}_0$. With $\boldsymbol{w} = \boldsymbol{O}\boldsymbol{z}$ and $\boldsymbol{t} = \boldsymbol{O}(\boldsymbol{T}\boldsymbol{u} + \boldsymbol{c}_0)$, the orthogonal transformation reduces completion to coordinatewise equations

$$F_j(w_j) = d_j w_j + \frac{\alpha}{\omega}\sin(\omega w_j) = t_j, \qquad F_j'(w_j) = d_j + \alpha\cos(\omega w_j), \tag{C.4a}$$

where $d_j = \Lambda_{jj}$. The corresponding Jacobians are

$$\boldsymbol{J}_{h,z}(\boldsymbol{z}) = \boldsymbol{O}^\top \operatorname{diag}\big(F_j'(w_j)\big)\boldsymbol{O}, \qquad \boldsymbol{J}_{h,u} = -\boldsymbol{T}. \tag{C.4b}$$

Thus, the eigenvalues of $\boldsymbol{J}_{h,z}$ are given directly by the coordinatewise derivatives. The orthogonal transformation couples variables in the original coordinates without changing these eigenvalues or the number of completion roots.

**Label-independent surrogate from a Taylor expansion.** Before training, fix an anchor $\bar{\boldsymbol{z}} = \boldsymbol{O}^\top \bar{\boldsymbol{w}}$, expand $G$ to first order, and retain the exact linear dependence on $\boldsymbol{u}$ and the instance offset $\boldsymbol{c}_0$. This gives

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{J}_{h,z}(\bar{\boldsymbol{z}}) = \boldsymbol{O}^\top \operatorname{diag}\big(F_j'(\bar{w}_j)\big)\boldsymbol{O}, \\ \boldsymbol{A}\boldsymbol{z} &= \boldsymbol{b}(\boldsymbol{u}; \boldsymbol{x}) = \boldsymbol{T}\boldsymbol{u} + \boldsymbol{c}_0 - G(\bar{\boldsymbol{z}}) + \boldsymbol{A}\bar{\boldsymbol{z}}. \end{aligned} \tag{C.4c}$$

The matrix $\boldsymbol{A}$ is nonsingular if every $F_j'(\bar{w}_j) \neq 0$. For label $(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star)$, use $\boldsymbol{r}_i = \boldsymbol{A}\boldsymbol{z}_i^\star - \boldsymbol{b}(\boldsymbol{u}_i^\star; \boldsymbol{x}_i)$. The sample-calibrated linear surrogate state, its explicit sensitivity, and the label-independent surrogate initialization are, respectively,

$$\tilde{\boldsymbol{z}}_i = \boldsymbol{z}_i^\star + \boldsymbol{A}^{-1}\boldsymbol{T}(\boldsymbol{u}_i - \boldsymbol{u}_i^\star), \qquad \boldsymbol{J}_{\tilde{z}_i,u} = \boldsymbol{A}^{-1}\boldsymbol{T}, \qquad \boldsymbol{z}^{\text{init}} = \boldsymbol{A}^{-1}\boldsymbol{b}(\hat{\boldsymbol{u}}; \boldsymbol{x}). \tag{C.4d}$$

The label-independent surrogate initializes the original nonlinear equality solver without test labels. This construction explicitly separates the true sensitivity $\boldsymbol{J}_{h,z}^{-1}\boldsymbol{T}$ at a regular completion point from the control sensitivity shared by both forms, allowing their conditioning to be controlled independently.

**Constructing curvature and two types of ill conditioning.** Nonlinear curvature is governed by $F_j''(w_j) = -\alpha\omega\sin(\omega w_j)$. Holding the phase distribution fixed while increasing $\omega$ strengthens curvature while preserving the corresponding Jacobian distribution at reference states. Taking $\alpha < 1$ close to 1 and moving the reference state to $w_1^\star \approx \pi/\omega$ makes $\boldsymbol{J}_{h,z}$ nearly singular while completion still has a unique root. Conversely, keeping the Jacobian well conditioned near the true reference state $w_1^\star = 0$ and moving only the linearization anchor $\bar{w}_1$ makes $\boldsymbol{A}$ nearly singular. These constructions

alter the exact completion sensitivity and the frozen sensitivity shared by both forms separately and must not be conflated.

**Multiple roots and solution branch structure.** When $\alpha < 1$, every $F_j'$ is positive, so each coordinatewise completion relation is strictly monotone. For $1 < \alpha < \kappa$, the first coordinate's derivative can change sign while the remaining coordinates stay strictly monotone. Let $\vartheta_{\text{fold}} = \arccos(-1/\alpha)$. The first coordinate has adjacent local maximum and minimum points at $w_{\text{L}} = \vartheta_{\text{fold}}/\omega$ and $w_{\text{R}} = (2\pi - \vartheta_{\text{fold}})/\omega$, respectively. When $F_1(w_{\text{R}}) < t_1 < F_1(w_{\text{L}})$, distinct left, middle, and right completion states can occur. Perturbing the controls changes $\boldsymbol{t}$, potentially eliminating the prescribed reference solution branch associated with the labeled point at a fold while other roots remain. Multiple roots here mean multiple equality completion states for the same controls, not multiple equivalent optimal solutions. Appendix D.3 specifies the reference solution branches associated with the labeled points and distance parameters.

**Constructing reference solutions and problem coefficients.** First select an interior control point $\boldsymbol{u}^\star$, a reference state $\boldsymbol{w}^\star$, and a multiplier $\bar{\boldsymbol{\lambda}}$ in rotated coordinates. Set $\boldsymbol{z}^\star = \boldsymbol{O}^\top \boldsymbol{w}^\star$ and $\boldsymbol{\lambda}^\star = \boldsymbol{O}^\top \bar{\boldsymbol{\lambda}}$, and then choose

$$\boldsymbol{c}_0 = G(\boldsymbol{z}^\star) - \boldsymbol{T}\boldsymbol{u}^\star, \quad \boldsymbol{c}_u = -\boldsymbol{u}^\star + \boldsymbol{T}^\top \boldsymbol{\lambda}^\star, \quad \boldsymbol{c}_z = -\rho \boldsymbol{z}^\star - \boldsymbol{J}_{h,z}(\boldsymbol{z}^\star)^\top \boldsymbol{\lambda}^\star. \tag{C.4e}$$

The reference point consequently satisfies the original equalities and the KKT stationarity conditions for an interior control point. Optimality still requires checking the second-order conditions of the Lagrangian. The reference solution branch used to generate the labeled points is fixed before sample generation and is not reassigned using method outputs. Appendix D.3 gives the sampling ranges, parameter values, optimality checks, and diagnostic results.

# D Supplementary Experiments and Reproducibility

This appendix documents the configurations and data splits, baseline definitions, evaluation protocols, and supplementary evidence for Section 4. The speedup and sample calibration ablations, CSCP diagnostics, and repeatability analyses clarify the sources and scope of the reported gains. Problem formulations and derivations of both forms are given in Appendix C. Here we focus on experimental settings and interpretation.

## D.1 System Configurations and Data Splits

Table D1 summarizes system sizes, variable dimensions, and scalar constraint counts. Table D2 summarizes the data allocation across four categories.

Table D1: System sizes, variable dimensions, and scalar constraint counts

| | Power | | | Gas | | | Water | | | Constructed |
|---|---|---|---|---|---|---|---|---|---|---|
| Metric | 14 | 57 | 118 | 11 | 135 | 4197 | Net1 | C-Town | Net6 | CSCP |
| Nodes | 14 | 57 | 118 | 11 | 135 | 4,197 | 9 | 388 | 3,323 | Synthetic |
| Branches | 20 | 80 | 186 | 11 | 170 | 4,466 | 13 | 444 | 3,892 | |
| Control variables | 9 | 13 | 107 | 3 | 6 | 11 | 1 | 5 | 20 | 32 |
| State variables | 22 | 106 | 181 | 21 | 304 | 6,840 | 22 | 832 | 7,215 | 32 |
| Equality constraints | 22 | 106 | 181 | 21 | 304 | 6,840 | 22 | 832 | 7,215 | 32 |
| Inequality constraints | 48 | 462 | 1,196 | 56 | 636 | 15,516 | 3 | 379 | 1,793 | 64 |

Table D2: Data splits for the ten test cases

| | Power | | | Gas | | | Water | | | Constructed |
|---|---|---|---|---|---|---|---|---|---|---|
| Subset | 14 | 57 | 118 | 11 | 135 | 4197 | Net1 | C-Town | Net6 | CSCP |
| Training | 9,500 | 9,500 | 9,500 | 8,500 | 8,500 | 8,500 | 10,000 | 9,500 | 9,500 | 9,500 |
| Frozen coefficient construction | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 |
| Validation | 1,500 | 1,500 | 1,500 | 2,500 | 2,500 | 2,500 | 1,000 | 1,500 | 1,500 | 1,500 |
| Test | 3,000 | 3,000 | 3,000 | 3,000 | 3,000 | 3,000 | 3,000 | 3,000 | 3,000 | 3,000 |

### D.2 Baselines, Gradient Verification, Comparison Protocols, and Speedup Ablations

Section 4 compares EG-Opt with Direct-NN, Partial-NN, Exact-ID, DC3, and ZO. The baselines drawn from the literature retain the core prediction, constraint handling, or gradient estimation mechanisms of the original methods and adapt them to our problems from different domains and evaluation interface. They do not reproduce every component of the systems in the source papers.

**Direct-NN and Partial-NN.** Following the prediction approach based on label supervision in Sun et al. (2018), Direct-NN supervises both control and state variables, using the same control supervision as Partial-NN and an additional state output head with a mean squared error loss. The control loss and state normalization follow the domain-specific definitions below. At deployment, it directly predicts all decision variables without invoking a completion or constraint correction procedure. Partial-NN follows the approach of predicting a subset of variables and then performing equality completion in Zamzam & Baker (2020). It is trained only on control variable labels, with no nonlinear completion during parameter updates. At deployment, it recovers the state variables using the same control variable partition and original equality solver as EG-Opt.

**Exact-ID.** During the training forward pass, Exact-ID recovers the state variables by solving the original nonlinear equalities and propagates task loss gradients to the control variables and network parameters through implicit differentiation. This baseline does not add iterative inequality correction, allowing us to isolate the effect of replacing nonlinear completion and its gradient computation with the sample-calibrated linear surrogate.

**DC3.** DC3 follows the framework combining partial prediction, equality completion, and inequality correction in Donti et al. (2021). The learning model predicts only the control variables, recovers the state variables through nonlinear equality completion, and then iteratively corrects inequality violations. Gradients are propagated through the completion map by implicit differentiation and through the actual correction iterations by backpropagation. Relative to Exact-ID, DC3 additionally incurs the cost of correction to improve inequality feasibility and its derivatives.

**ZO.** ZO uses the two-point zeroth-order estimator of DeepOPF (Pan et al., 2023), with $K = 1$ throughout. Each update samples one independent random unit direction per sample and estimates the gradient from two nonlinear completions and loss evaluations at opposite perturbations, without differentiating through the solver. The estimator is applied to the full scalar training loss, without adding analytical gradients of individual loss terms. This direction budget does not increase with problem dimension. The radius for finite differences follows the common tuning protocol. Gradient fidelity diagnostics use an initial radius of $10^{-4}$.

**Fair comparison protocol.** All methods share the original problem, data splits, objective and constraints, frozen scaling, evaluation samples, numerical precision, hardware, and metric definitions. Network backbones are matched while preserving the required output dimensions and core computational graphs.

1. Within each case, EG-Opt, Exact-ID, DC3, and ZO share reference labels, control supervision, normalization, and all loss weights. Their domain-specific losses are specified below. Equation (5) is a basic example, not the exact loss used in every experiment. The methods differ in state computation, gradient evaluation, and DC3's inequality correction, not in available supervision or separately tuned loss weights. Direct-NN and Partial-NN use only their prescribed supervised losses. The boundary diagnostics in Appendix D.3 specify their loss separately and match it across the compared methods.
2. Each domain uses a fixed training configuration. Within each case, methods share the control network backbone, optimizer, learning rate schedule, batch size, and prescribed training budget, with the output dimensions and computation required by each method retained. All methods and cases use no pretraining. Main-table evaluation uses the final model after the prescribed budget, not a checkpoint selected by validation or test performance. Test feasibility is not used to select seeds, checkpoints, or samples, and methods that miss the target receive no additional training. Normalization statistics are computed from the training set only.
3. Table 1 reports arithmetic means over three independently trained models per case and method, using distinct random seeds rather than selecting the best run. Paired comparisons share data and, where compatible, initialization and batch order. Solver failures remain in feasibility denominators and total runtime. Objective gaps and feasibility are evaluated separately on the un-

changed evaluation set. Each independent run starts from a reinitialized model and completes the training procedure. Repeated evaluations or timings of the same model are not additional training runs. The first qualifying checkpoint used for Time to Target does not replace the final model reported in Table 1. Appendix D.4 distinguishes these sources of variation.

**Domain-specific training losses.** The following settings are fixed across EG-Opt, Exact-ID, DC3, and ZO within each case. They instantiate the supervision, objective, and constraint terms, including auxiliary equality penalties where used. The chain rule in Eq. (6) uses derivatives of the selected loss. For batch-dependent tail penalties, these derivatives are taken from the full batch loss.

**Power.** Control supervision, objective, equality, and inequality weights are $1, 10^{-3}, 1, 20$, respectively. Control supervision uses Bernoulli-KL balanced across generator active-power and voltage-magnitude groups. The equality penalty is the mean squared residual. The inequality penalty uses softplus with $\beta = 25$ for Case57 and $\beta = 50$ for Case14 and Case118. Direct-NN uses the same control supervision plus state MSE normalized by physical units. Partial-NN uses only control supervision.

**Water.** Control supervision, objective, equality, and inequality weights are $1, 10^{-3}, 1, 20$, with no state supervision in the four physics-based methods. Control MSE is normalized by the control range, and equality residuals use fixed scales. The inequality penalty uses softplus with temperature $10^{-3}$ and zero threshold, taking the worst member within each constraint family and summing over families. Direct-NN uses control MSE and standardized state MSE, both with unit weight. The state standard deviation is floored at the larger of $10^{-6}$ times the absolute mean and $10^{-6}$. Partial-NN uses control MSE only.

**Gas.** The weights for control supervision, objective, inequalities, and the worst $1\%$ tail penalty are $1, 10^{3}, 10^{4}, 10^{4}$, respectively, with safety margin $10^{-3}$. The number of tail samples is rounded up from the actual batch size. Control scaling uses the larger of the training standard deviation and $10^{-3}$ times the control range. Direct-NN uses control MSE and standardized state MSE with unit weights, and Partial-NN uses control MSE only. State normalization statistics are computed from the training set. Neither supervised baseline uses physical losses or nonlinear completion during training.

**Standard CSCP.** Control supervision, objective, and equality weights are $1, 10^{-3}, 1$, respectively. Control supervision is normalized by the control range. The signed objective difference is scaled by the mean absolute reference objective on the training set, and equality residuals are normalized by the training standard deviation of the right-hand sides. The shared output mapping enforces control box bounds, so no additional inequality penalty is used. Direct-NN uses control MSE and standardized state MSE with unit weights, whereas Partial-NN supervises controls only. This training objective is distinct from the absolute relative objective gap used for evaluation.

**Numerical gradient verification.** Exact-ID serves as both a training baseline and a reference for gradient diagnostics. We record its forward completion residual distribution and adjoint linear system residuals. At finite tolerances, its gradient is an approximation with controlled residuals to the derivative of the implicit root map, rather than the exact derivative of the algorithmic map of the solver after finitely many steps. Sampled checks use double precision and stricter completion and adjoint tolerances. At the same $(\boldsymbol{x}, \boldsymbol{u})$, reference solution branch associated with the labeled point, and task loss, we recompute a completion to high accuracy at each perturbed point and compare Exact-ID against central finite differences, reporting absolute and relative errors, angular discrepancies, and sensitivity to the step size used for finite differences. Full Unrolled-to-Tolerance is instead checked against central finite differences of the algorithmic map, using the same initialization, iteration limit, and stopping rule. At the verification points, we fix the actual iteration count or avoid stopping boundaries, and examine sensitivity to the iteration count and stopping threshold.

**Common objective gap definition.** Table 1, Figure D3, and Time to Target use the mean absolute relative objective gap, expressed as a percentage. Let $(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{z}}_i)$ be the method output, $(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star)$ the reference decision, and $\mathcal{I}_{\text{obj}}$ the set of evaluation samples for which both objective values are finite. Then

$$\text{Gap}(\%) = \frac{100}{|\mathcal{I}_{\text{obj}}|} \sum_{i \in \mathcal{I}_{\text{obj}}} \frac{|F(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{z}}_i; \boldsymbol{x}_i) - F(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star; \boldsymbol{x}_i)|}{\max\{|F(\boldsymbol{u}_i^\star, \boldsymbol{z}_i^\star; \boldsymbol{x}_i)|, 10^{-12}\}}. \tag{D.1}$$

This metric takes the absolute relative gap for each sample before averaging. It does not filter samples by constraint feasibility, and it is neither a difference of means nor a mean signed gap. Objective

value coverage is $|\mathcal{I}_{\text{obj}}|/N$. With full coverage, the metric is the mean over the entire evaluation set. Nonfinite objective values are not assigned a zero gap. TTT additionally requires $100\%$ objective value coverage and simultaneous satisfaction of the thresholds for equality and inequality feasibility in the main text. A small objective gap alone implies neither constraint feasibility nor certified global optimality of the reference solution.

**Algorithmic gains and implementation optimizations.** EG-Opt replaces nonlinear completion and implicit differentiation within training with state computation and explicit gradients from the sample-calibrated linear surrogate. The shared matrix $\boldsymbol{A}$ and its factorization are fixed and reused. If the explicit $\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})$ map is nonlinear in $\boldsymbol{u}$, then $\boldsymbol{J}_{b,u}(\boldsymbol{u};\boldsymbol{x})$ and the sensitivity $\boldsymbol{A}^{-1}\boldsymbol{J}_{b,u}(\boldsymbol{u};\boldsymbol{x})$ may still vary with the prediction. They are evaluated through ordinary automatic differentiation and linear solves using the cached factorization. Only when $\boldsymbol{b}(\boldsymbol{u};\boldsymbol{x})$ is affine for a fixed instance and route is the sample-calibrated linear surrogate sensitivity independent of $\boldsymbol{u}$, allowing further precomputation and reuse. The network Jacobian and loss derivatives with respect to the states still require ordinary backpropagation. The full loss gradient is not frozen. Keeping data on the GPU, batching, and reusing fixed operators can further reduce assembly, synchronization, and control flow costs. The device cache ablation within EG-Opt (Table D3) examines one implementation optimization, while the CSCP comparison without a GPU (Table D4) evaluates training gains without GPU acceleration. Their results are not combined into a single speedup. All methods may use standard implementation optimizations, with actual computational costs measured under consistent timing boundaries.

**Operation counts and timing scope.** Under the common $K = 1$ protocol, ZO requires one unperturbed and two perturbed nonlinear completions per sample used for physical computation (Table 2). Its implicit gradient count is zero, but runtime includes ordinary backpropagation. EG-Opt's linear row counts samples processed by the sample-calibrated linear surrogate, not matrix factorizations. Figure 3 uses shared states and batches at three training phases, with nine timings per phase. It reports the median of the phase medians and their range. Water network batches contain $256$ samples, of which $64$ enter physical computation. End-to-end training, timing of individual updates, and time to target follow Appendix A.7, with EG-Opt as the speedup denominator under the same timing scope. Validation and deployment completion calls are counted separately from training updates. Preparation of both surrogate forms enters end-to-end training time, whereas data and label generation are excluded from all reported training times. No method uses pretraining.

**Device cache ablation for fixed sensitivity operators.** This experiment keeps EG-Opt's computational method fixed and changes only whether device copies of the precomputable sensitivity operator of the sample-calibrated linear surrogate $\boldsymbol{S}_{\text{lin}} = \boldsymbol{A}^{-1}\boldsymbol{J}_{b,u}(\boldsymbol{u};\boldsymbol{x})$, including its batched tensors, are reused across batches. Cache ON loads the operator onto the GPU before training and reuses it for each batch. Cache OFF creates and transfers a new GPU tensor that does not require gradients from the same immutable host copy at every update. Both settings construct the same operator only once, neither refactor $\boldsymbol{A}$ nor recompute the sensitivity, and leave the trainability of network parameters unchanged. The network, initial weights, training data and batch order, loss, optimizer, learning rate schedule, and update budget are identical. General optimizations, including data residency, batching, caching of static tensors, and synchronization policies, remain enabled in both settings. The ablation therefore isolates the benefit of caching the operator on the device, rather than differences between the CPU and GPU or the combined benefit of all implementation optimizations.

The ablation covers the nine engineering cases and Water-L-Town, excluding CSCP, on an RTX 5060 Ti (16 GB) with PyTorch 2.9.0 and CUDA 12.9. Each case uses one fixed training seed and four paired timing runs, alternating execution order. Both settings start from the same initial state and train for eight epochs. Synchronized timing includes batch indexing, forward and backward computation, loss evaluation, optimizer updates, scheduling, and loop checks. Initialization, operator construction, warmup, validation, testing, and checkpoint writes are excluded. All forty pairs produce identical final losses and model states, update counts, and solver workloads. Caching changes only the operator transfers from host to device, from one per update to zero.

In Table D3, the median paired speedup for each case ranges from $1.012$–$1.102\times$. The median of these case medians is $1.054\times$, corresponding to approximately $5\%$ time savings in the training loop. Three of the forty pairs show no speedup. All repetitions are retained in the statistics. The results show that caching fixed operators on the device provides an additional benefit without changing parameter updates, but cannot by itself explain the large speedups in the main experiments.

Table D3: Isolated device cache ablation for the fixed sensitivity operator in EG-Opt

| Case | Updates per run | Cache OFF (s) | Cache ON (s) | Paired speedup median [min, max] |
|---|---|---|---|---|
| Power-Case14 | 304 | 6.142 | 5.578 | 1.102 [1.013, 1.163] |
| Power-Case57 | 304 | 6.021 | 5.519 | 1.084 [1.036, 1.161] |
| Power-Case118 | 304 | 6.095 | 5.725 | 1.059 [1.042, 1.075] |
| GasLib-11 | 536 | 6.661 | 6.136 | 1.037 [1.008, 1.118] |
| GasLib-135 | 536 | 9.519 | 8.850 | 1.083 [0.949, 1.174] |
| GasLib-4197 | 536 | 20.668 | 18.667 | 1.101 [1.081, 1.218] |
| Water-Net1 | 320 | 2.347 | 2.206 | 1.049 [1.025, 1.129] |
| Water-C-Town | 304 | 6.415 | 6.244 | 1.012 [0.992, 1.050] |
| Water-L-Town | 304 | 3.959 | 3.803 | 1.043 [1.034, 1.051] |
| Water-Net6 | 304 | 20.582 | 20.393 | 1.016 [0.965, 1.060] |

**CSCP time to target without a GPU.** We compare all six methods on a single CPU worker, with one thread per computational library and no solver process pool. The experiment uses double precision, $9500$ training samples, batches of $256$, and a SiLU MLP with three layers and width $256$ with a linear skip connection. Methods share matched initial states and the same full training budget, using AdamW with cosine learning rate decay from $10^{-3}$ to $10^{-5}$. ZO uses one direction per sample with perturbations on both sides ($K = 1$). We evaluate the same $1500$ validation samples after every epoch. The first qualifying checkpoint must have finite objective values throughout, a mean absolute relative objective gap of at most $1\%$ by Eq. (D.1), $100\%$ equality feasibility (residual measured in the infinity norm at most $10^{-8}$), and at least $99\%$ inequality feasibility. The only inequalities are control box constraints.

Following the timing scope for the update stage in Appendix A.7, we measure time to target by accumulating the durations of the update stage through the first qualifying epoch: $T^{(a)}_{\text{update}} = \sum_{e=1}^{E^{(a)}_{\text{hit}}} t^{(a)}_{\text{update},e}$. Here, $E^{(a)}_{\text{hit}}$ is the epoch of the first hit and $t^{(a)}_{\text{update},e}$ is the measured update time of epoch $e$. The total includes batch indexing, forward and backward passes, completion, gradient clipping, optimizer steps, and scheduling. It excludes offline preparation, initialization, validation, disk writes, and test inference, and therefore differs from the end-to-end training time in Table 1.

Table D4: Measured CSCP time to target with a single CPU worker

| Method | First hit epoch | Measured TTT (s) | Time relative to EG-Opt | Mean abs. rel. objective gap (%) |
|---|---|---|---|---|
| **EG-Opt** | **11** | **3.808** | **1.000** | **0.886** |
| Exact-ID | 11 | 4.714 | 1.238 | 0.886 |
| DC3 | 11 | 4.725 | 1.241 | 0.886 |
| ZO | 112 | 68.632 | 18.023 | 0.970 |
| Partial-NN | 11 | 3.434 | 0.902 | 0.886 |
| Direct-NN | Not reached | Not reached | — | 0.096 (final) |

In Table D4, EG-Opt, Exact-ID, and DC3 reach the target at epoch $11$, with EG-Opt respectively $1.238\times$ and $1.241\times$ faster than the latter two. DC3 requires no additional inequality correction in this setting, so its quality trajectory matches Exact-ID's. ZO reaches the target at epoch $112$, taking $18.023$ times EG-Opt's update time: cheaper updates alone do not guarantee faster convergence to the target. Partial-NN takes $0.902$ times EG-Opt's time, but this update-only comparison excludes preparation and validation and does not establish a lower end-to-end cost from an existing labeled pool. These results demonstrate training gains without GPU acceleration for this CSCP configuration and single training seed, not a universal speedup. The CPU comparison and device cache ablation test different effects. Their speedups cannot be combined.

### D.2.1 Preparation and Training Time Shares Across Methods

We compare the preparation and full training stages of all six methods on Power-Case14, Power-Case57, and Power-Case118. For each run, each stage's time is divided by that run's end-to-end training time, and the resulting percentages are averaged over three independent training seeds. These are means of per-run percentages, not ratios of mean times. These totals follow Table 1: they start from existing labeled data and exclude data and label generation. Preparation is an aggregate stage rather than an isolated measurement of linear surrogate construction. Figure D1 shows the full composition alongside a separate view of the small preparation shares.

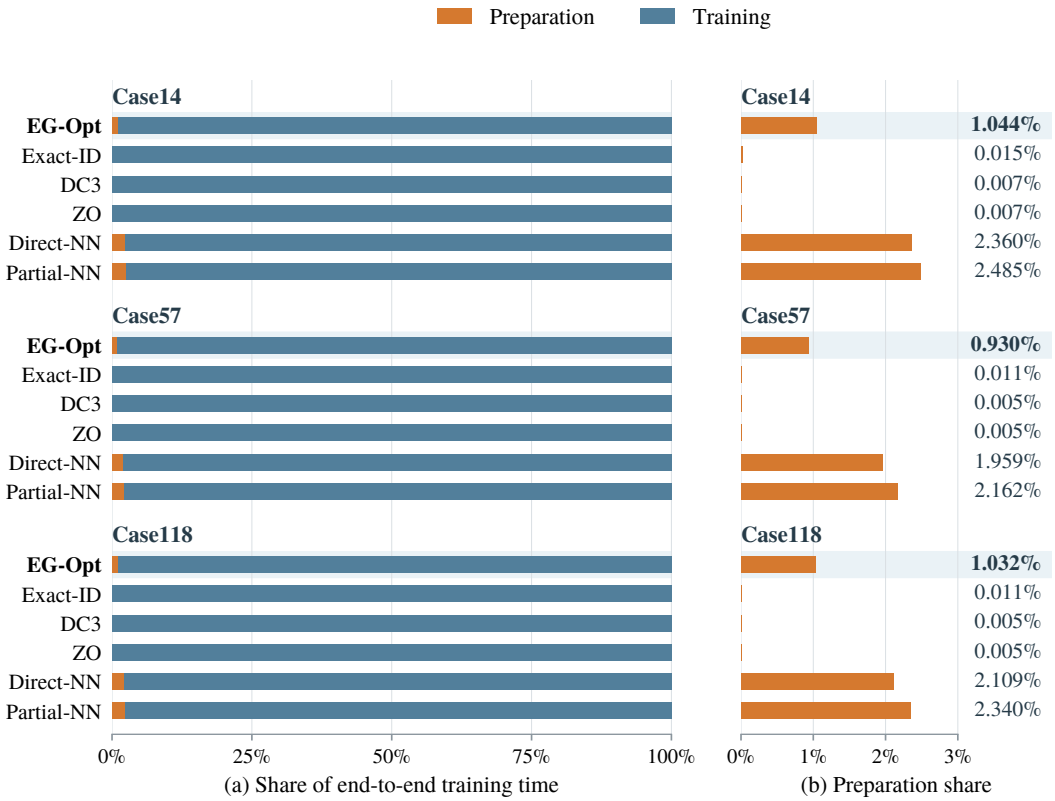



Figure D1: Preparation and training time shares across six methods on three power cases

Preparation accounts for less than 2.5% of total time across all six methods, leaving training as the dominant cost. For EG-Opt, preparation occupies only 0.93–1.04%, while full training accounts for 98.96–99.07%. Thus, EG-Opt's training acceleration is not offset by a large preparation burden in these three power cases.

The smaller preparation shares of Exact-ID, DC3, and ZO (below 0.02%) do not by themselves establish lower absolute preparation costs. Conversely, the 1.96–2.48% shares of Direct-NN and Partial-NN are larger primarily because their shorter training runs yield smaller total-time denominators. Stage proportions should therefore be interpreted together with the absolute end-to-end times in Table 1, rather than as another ranking of computational efficiency.

### D.2.2 Nonlinear Completion with Approximate Surrogate Gradients

To distinguish the contributions of explicit surrogate gradients and linear state computation, we compare EG-Opt and Exact-ID with a hybrid variant on Power-Case14, Power-Case57, and Power-Case118. The hybrid retains nonlinear completion for forward state recovery but replaces implicit differentiation with approximate gradients based on the explicit sensitivity of the sample-calibrated linear surrogate. Each case is evaluated on 3,000 test instances using the objective gap and feasibility metrics defined in Appendix D.2. Table D5 compares solution quality and end-to-end training time.

Hybrid times use the same measured end-to-end scope as Table 1, including preparation and full training from existing labels but excluding label generation.

Table D5: Effect of nonlinear completion with approximate surrogate gradients

| Case | Method | Objective gap (%) | Eq. feasibility (%) | Ineq. feasibility (%) | E2E training time (s) |
|---|---|---|---|---|---|
| Power-Case14 | **EG-Opt** | **0.0983** | **100.0** | **100.0** | **227.7** |
| | Hybrid | 0.0998 | 100.0 | 100.0 | 10,138.4 |
| | Exact-ID | 0.0986 | 100.0 | 100.0 | 16,587 |
| Power-Case57 | **EG-Opt** | **0.0706** | **100.0** | **100.0** | **236.1** |
| | Hybrid | 0.0700 | 100.0 | 100.0 | 16,659.3 |
| | Exact-ID | 0.1057 | 100.0 | 99.9 | 26,538 |
| Power-Case118 | **EG-Opt** | **0.0145** | **100.0** | **100.0** | **238.8** |
| | Hybrid | 0.0147 | 100.0 | 100.0 | 18,778.4 |
| | Exact-ID | 0.0146 | 100.0 | 100.0 | 30,594 |

EG-Opt and Exact-ID values are from Table 1.

The hybrid achieves 100% equality and inequality feasibility on all three cases, with objective gaps close to those of EG-Opt. Its end-to-end training time is 61.12–62.78% of Exact-ID's, or approximately 60%, corresponding to time savings of 37.22–38.88%. However, it still takes 44.53–78.64 times as long as EG-Opt. Thus, approximate gradient computation alone does not recover the full training benefit. These power case results support the complementary roles of explicit surrogate gradients and linear state computation, with the latter avoiding repeated nonlinear completion while retaining comparable solution quality.

### D.3 CSCP Gradient and Deployment Diagnostics

Using diagnostics at frozen checkpoints and controlled experiments in control space, this section distinguishes gradient accuracy, descent capability, and deployment solution branch recovery. Gradient diagnostics evaluate the sample-calibrated linear surrogate, whereas deployment uses the label-independent surrogate for warm starts. The model is given in Appendix C.4.

#### D.3.1 Experimental Settings and Sample Construction

We fix $m = 32, \rho = 2, \kappa = 10$ and report these results separately from the standard CSCP main experiments. Reference points are constructed using Eq. (C.4e), with $\|\bar{\boldsymbol{\lambda}}\|_\infty \le 0.05$. Positive definiteness of the Lagrangian Hessian is verified through $\rho - 0.05\alpha\omega > 0$.

**Frozen models.** The five scenarios are defined as follows. The regular scenario uses $(\alpha, \omega) = (0.5, 2)$. The scenario with high curvature uses $(0.5, 16)$. Nearly singular completion is obtained with $(0.99, 2)$ and $w_1^\star \in [\pi/\omega - 0.02, \pi/\omega + 0.02]$. The scenario with a nearly singular shared matrix uses $(1 - 10^{-6}, 2)$, $w_1^\star = 0$, and $\mu_A = 10^{-3}$. The scenario with multiple roots uses $(1.05, 2)$ and distance $s = 0.5$ along the left reference solution branch associated with the labeled point. We retain the orthogonal transformations and normalization used in training. For each scenario, EG-Opt and Exact-ID each contribute five independently trained models. Their final checkpoints after full training are evaluated on the same 512 diagnostic samples.

**Mechanism sweeps.** We set $\boldsymbol{O} = \boldsymbol{T} = \boldsymbol{I}$ and do not train a network. Three Sobol sample sets generated with distinct seeds each contain 1024 samples with $\boldsymbol{u}^\star \in [-0.3, 0.3]^{32}$ and ordinary phases $\omega \boldsymbol{w}^\star \in [-0.6, 0.6]^{32}$. Four normalized Gaussian directions are used per sample. Samples and directions are shared across settings, with the first coordinate set according to the factor being swept. The shared matrix conditioning sweep fixes $\alpha = 1 - 10^{-7}, \omega = 2, w_1^\star = 0$ and moves only the anchor, jointly examining $\mu_A \in [10^{-6}, 2 - 10^{-7}]$ and $\delta \in [10^{-4}, 0.6]$. The curvature sweep uses $\alpha = 0.5, \omega \in \{1, 2, 4, 8, 16\}$. The sweep for nearly singular completion fixes $\omega = 2, w_1^\star = \pi/\omega$ and varies $1 - \alpha \in [10^{-5}, 0.5]$. Both use $\delta \in \{0.001, 0.03, 0.3\}$. The multiple root sweep uses $\alpha = 1.05, \omega = 2, s \in [0.001, 0.5]$ and $\delta \in \{10^{-4}, 10^{-3}, 10^{-2}\}$. Figure 5(h) uses $\delta = 10^{-3}$.

The left reference solution branch associated with the labeled point is prescribed by $w_1^\star = [\vartheta_{\text{fold}} - s(\pi - \vartheta_{\text{fold}})]/\omega$, with smaller $s$ placing the reference point closer to the fold. The anchor shared by both forms is $\bar{\boldsymbol{w}} = \mathbf{0}$.

#### D.3.2 Gradient and Deployment Evaluation Protocols

**Parameter gradients.** The frozen models use a training loss for boundary diagnostics comprising control supervision, a normalized objective term, and an auxiliary equality term, with weights $1, 10^{-3}, 1$. Like the standard CSCP training configuration, this diagnostic loss includes an auxiliary equality term absent from the basic example in Eq. (5). The boundary scenarios and models remain separate from the main-table experiment. We also evaluate the original objective alone. For each checkpoint, four batches of $64$ samples are used to compare gradients of the sample-calibrated linear surrogate and exact completion losses at the same parameters, computing $R_{\boldsymbol{\theta}}$ by Eq. (9). Updates along unit directions have length $h = c\max(1, \|\boldsymbol{\theta}\|_2)$, with $c \in \{10^{-5}, 10^{-4}, 10^{-3}\}$. The main results use $10^{-4}$. Descent is always assessed using the corresponding loss after exact completion at the updated parameters.

**Control gradients.** The mechanism experiments apply the sample-calibrated linear surrogate without network training and evaluate only the control gradient error $R_{\boldsymbol{u}}$ for the original objective and descent after a single step with equal step lengths. These results are not pooled with checkpoint diagnostics. Relative error is undefined when the exact gradient norm is at most $10^{-10}$. The figures report medians over valid observations. The short error bars in Figure 5 are $95\%$ descriptive intervals obtained from $1000$ cluster bootstrap resamples, keeping all directions for a given sample together.

**Deployment.** The warm start from the label-independent surrogate and the diagonal initializer $\boldsymbol{w}_0 = \boldsymbol{\Lambda}^{-1}\boldsymbol{O}(\boldsymbol{T}\boldsymbol{u} + \boldsymbol{c}_0)$ use the same damped Newton solver, with tolerance $10^{-10}$, at most $40$ iterations, at most $12$ backtracking trials per iteration, and a cap on step length of $\pi/\omega$. An independent reference solve uses tolerance $10^{-12}$ and at most $120$ iterations. Solution branch recovery is recorded if the returned solution's distance measured in the infinity norm to the target root is at most $10^{-6}$. Reference roots are not used at deployment. Completion and recovery of the reference solution branch associated with the labeled point are both evaluated over all samples, including solver failures and samples for which the reference solution branch associated with the labeled point does not exist.

#### D.3.3 Supplementary Results and Scope

In the regular scenario and the scenario with high curvature, parameter gradients of the sample-calibrated linear surrogate loss at both types of checkpoints are accurate and yield descent (Table D6). When the state phases are fixed, increasing the frequency also does not substantially worsen control gradient error (Figure D2). These results support the effectiveness of the sample-calibrated linear surrogate in the regular settings tested, but do not extend to all nonlinear settings.

When the completion Jacobian is nearly singular, the full loss can still decrease even though the gradient error for the original objective alone is close to one. Thus, descent of the full loss does not establish the accuracy of the gradient of the original objective. Severe ill conditioning of the shared matrix $\boldsymbol{A}$ substantially amplifies error and impairs both descent at EG-Opt checkpoints and completion initialized by the label-independent surrogate (Tables D6 and D7), making it an important failure factor. Descent remains possible at some checkpoints or with smaller steps, further demonstrating that descent capability does not imply an accurate gradient approximation. Moreover, as the exact gradient becomes small near a stationary point, residual sensitivity bias becomes more prominent in relative error. A larger relative error under smaller perturbations therefore does not imply a larger absolute error and should be interpreted together with descent behavior.

With multiple roots, checkpoint gradient diagnostics select the regular root closest to the initialization from the label-independent surrogate, which need not lie on the left reference solution branch associated with the labeled point. Root switching is observed during updates, so these diagnostics assess actual updates but cannot verify the local descent theorem for a fixed solution branch. With predictions and solver settings fixed, warm starts from the label-independent surrogate recover the left reference solution branch associated with the labeled point more often than diagonal initialization (Table D7). When the shared matrix is nearly singular, switching to the diagonal initializer restores completion, showing that deployment failures also require analysis separate from gradient

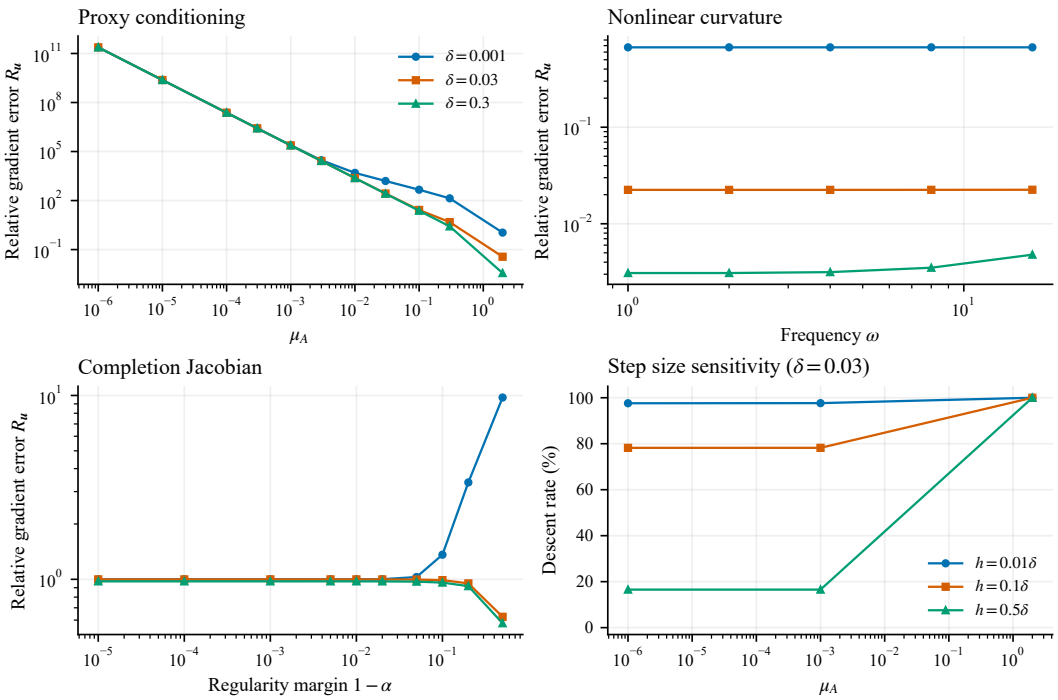



Figure D2: Additional CSCP stress scans and step sensitivity for sample-calibrated linear surrogate gradients

Table D6: Parameter gradient diagnostics by checkpoint training method and loss

| Scenario | Checkpoint training method | Full loss for boundary training | | | Original objective only | | |
|---|---|---|---|---|---|---|---|
| | | $R_{\boldsymbol{\theta}}$ | Calibrated direction descent (%) | Exact direction descent (%) | $R_{\boldsymbol{\theta}}$ | Calibrated direction descent (%) | Exact direction descent (%) |
| Regular | EG-Opt | 0.0011 | 100 | 100 | 0.095 | 100 | 100 |
| | Exact-ID | 0.0011 | 100 | 100 | 0.095 | 100 | 100 |
| High curvature | EG-Opt | 0.046 | 100 | 100 | 0.072 | 100 | 100 |
| | Exact-ID | 0.049 | 100 | 100 | 0.076 | 100 | 100 |
| Nearly singular completion | EG-Opt | 0.45 | 100 | 100 | 1.0 | 85 | 100 |
| | Exact-ID | 0.43 | 100 | 100 | 1.0 | 90 | 100 |
| Nearly singular matrix $\boldsymbol{A}$ | EG-Opt | 1300 | 5 | 100 | 6300 | 0 | 100 |
| | Exact-ID | 70000 | 100 | 100 | 330000 | 100 | 100 |
| Multiple roots | EG-Opt | 0.51 | 100 | 100 | 1.0 | 50 | 100 |
| | Exact-ID | 0.44 | 100 | 95 | 1.0 | 10 | 95 |

$R_{\boldsymbol{\theta}}$ is the median over 20 batches, rounded to two significant figures. Each descent rate uses the same 20 batches. Calibrated directions use the sample-calibrated linear surrogate. Exact-ID rows apply this training form to checkpoints trained by Exact-ID, measuring counterfactual approximation errors rather than errors in Exact-ID's own gradients.

diagnostics. Figure 5(h) further shows that successful completion does not imply recovery of the reference solution branch associated with the labeled point. Near a fold, disappearance of the reference solution branch associated with the labeled point must also be distinguished from numerical solver failure.

Table D7: Completion and recovery of the reference solution branch associated with the labeled point using different initializations (%)

| Scenario | Checkpoint training method | Completion rate | | Reference solution branch recovery rate | |
|---|---|---|---|---|---|
| | | Label-independent surrogate | Diagonal init. | Label-independent surrogate | Diagonal init. |
| Nearly singular matrix $\boldsymbol{A}$ | EG-Opt | 0 | 100 | 0 | 100 |
| | Exact-ID | 0 | 100 | 0 | 100 |
| Multiple roots | EG-Opt | $61.3 \pm 2.1$ | $34.8 \pm 1.4$ | $54.6 \pm 1.6$ | $5.4 \pm 0.5$ |
| | Exact-ID | $72.0 \pm 0.8$ | $33.0 \pm 2.1$ | $67.2 \pm 1.7$ | $7.3 \pm 1.3$ |

Mean $\pm$ standard deviation across five models per training method, evaluated on the same 512 samples. Zero standard deviations are omitted for entries that are uniformly 0 or 100. In the other three scenarios, both initializers achieve 100% completion and solution branch recovery.

### D.4 Independent Training Runs and Timing Variation

Table 1 reports arithmetic means of objective gaps, feasibility rates, and consistently timed runtimes over three independent training runs. Table D8 separately reports variation across five CSCP training runs, while Table D9 measures timing variation at fixed states. These repetition counts are not pooled: test samples, repeated timings, and training phases are not independent training runs.

**Solution quality across independent training runs.** Table D8 uses the regular CSCP boundary configuration in Appendix D.3, not the standard configuration of the main experiment. Each method completes five independent training runs with distinct seeds and fixed data and hyperparameters. Initialization and batch order are matched between methods within each paired run. NoCal omits the calibration term as defined in Appendix D.5. Final models are evaluated on the same 3000 test samples using Eq. (D.1) and the feasibility criteria. We report the mean and sample standard deviation across runs, without pooling test predictions. Objective value coverage is 100%.

Table D8: Variability across independent training runs on the normal CSCP boundary setting

| Method | Training runs | Mean abs. rel. objective gap (%) | Equality feasibility (%) | Inequality feasibility (%) |
|---|---|---|---|---|
| EG-Opt | 5 | $0.0481 \pm 0.0002$ | $100.00 \pm 0.00$ | $100.00 \pm 0.00$ |
| Exact-ID | 5 | $0.0481 \pm 0.0002$ | $100.00 \pm 0.00$ | $100.00 \pm 0.00$ |
| NoCal | 5 | $0.0482 \pm 0.0002$ | $100.00 \pm 0.00$ | $100.00 \pm 0.00$ |

EG-Opt and Exact-ID show similar objective gaps and variability in this regular scenario. The small difference between EG-Opt and NoCal does not establish a generally significant calibration benefit. All five runs achieve 100% deployment feasibility, although zero observed variation does not imply zero population variance. Concurrent training makes these wall clock records unsuitable for runtime comparisons or estimates of end-to-end speedup.

**Repeated timing at fixed states.** Table D9 supplements Figure 3 with timing variation for EG-Opt and Exact-ID. For each engineering case, we fix the state and batch at early, middle, and late training phases and repeat timing nine times per phase. Runtime is the median of the three phase medians. The coefficient of variation (CV) is the sample standard deviation of the nine timings within a phase divided by their mean, reported as a range across phases. The 27 timings are not pooled to estimate a single variance.

EG-Opt's CV within each phase is below 6% throughout. Exact-ID's is below 5% except in C-Town's early and middle phases. This supports stable comparisons of update time at fixed states, not low variability across independent training runs. These timings also exclude the preparation costs in Appendix A.7 and therefore do not characterize end-to-end runtime variation.

Table D9: Timing of individual updates at fixed states and variability within each phase

| Case | EG-Opt | | Exact-ID | |
|---|---|---|---|---|
| | Time ms per update | CV (%) | Time ms per update | CV (%) |
| Power-Case14 | 16.55 | 1.71–2.93 | 1106.51 | 0.54–1.68 |
| Power-Case57 | 16.42 | 1.63–4.00 | 1275.39 | 0.45–1.62 |
| Power-Case118 | 16.69 | 0.89–1.42 | 1521.75 | 0.36–0.56 |
| GasLib-11 | 10.16 | 3.16–5.99 | 60.34 | 3.96–4.51 |
| GasLib-135 | 14.38 | 1.08–3.43 | 148.75 | 1.43–2.22 |
| GasLib-4197 | 30.54 | 0.94–1.08 | 1343.70 | 0.43–1.21 |
| Water-Net1 | 5.86 | 1.24–2.84 | 41.77 | 3.56–4.85 |
| Water-C-Town | 16.30 | 1.13–2.63 | 488.54 | 0.64–10.81 |
| Water-Net6 | 66.39 | 1.35–2.26 | 4236.27 | 0.55–2.23 |

### D.5 The Role of Sample Calibration

We compare full EG-Opt (Cal) with NoCal on the nine engineering cases. NoCal sets the calibration term $r_i$ to zero while keeping all other training settings unchanged. This ablation removes only the zeroth-order state correction at labeled points, leaving the sensitivity to control variables unchanged. Both use the same label-independent surrogate for warm starts and the same original nonlinear completion solver at deployment.

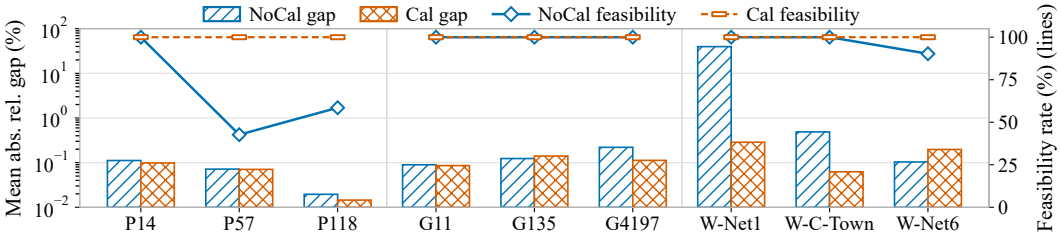



Figure D3: Effects of sample calibration on objective quality and overall feasibility across nine engineering cases

Figure D3 shows that Cal reduces objective gaps in several test cases and raises the overall feasibility rate to 100% on Power-Case57/118 and Water-Net6. On GasLib-135, NoCal achieves a slightly lower objective gap with the same feasibility rate. On Water-Net6, however, NoCal's smaller objective gap is accompanied by inequality violations, with overall feasibility of 90.27% compared with 100% for Cal. Because the objective gap is evaluated over the full evaluation set, outputs outside the feasible region also contribute to this lower value. Thus, the smaller gap does not establish better feasible solution quality. Cal maintains full feasibility while retaining a small objective gap, highlighting the value of sample calibration for reliable deployment. Overall, sample calibration improves objective quality or deployment robustness in most paired comparisons.

### D.6 Code Availability and Reproducibility

The code for EG-Opt is available for anonymous review at `https://anonymous.4open.science/r/EG-Opt-C46C`. The problem formulations and linear surrogate constructions are described in Appendix C, while Appendix D.2 specifies the baseline definitions, domain-specific training losses, and comparison protocols.

## E Use of AI in Manuscript Preparation

Generative AI tools were used to assist with writing and language polishing, including improvements to grammar, clarity, and organization. The authors reviewed and revised the AI-assisted text and take full responsibility for the scientific content, experimental results, and conclusions.