FROM NESTED SOLVING TO PREDICTIVE TRACKING: LEARNING LOWER-LEVEL DYNAMICS FOR BI-LEVEL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bi-level optimization couples an upper-level objective with a parameter-dependent lower-level problem, requiring repeated lower-level solves as the upper-level variable evolves. Repeatedly solving the lower-level problem is costly, but inaccurate responses distort the upper-level update. We introduce SUPRO, a first-order method that tracks a persistent lower-level response across outer iterations. SUPRO models the joint state of the lower-level iterate and an auxiliary value response to predict response motion, and adapts the predictor online using feedback from subsequent corrections. Fixed-budget gradient corrections and objective-based candidate selection control prediction errors. We establish a non-asymptotic stationarity guarantee for the associated fixed value-function penalty, together with a response-tracking bound that improves as predictions become more accurate. SUPRO uses only single-loop computation and requires neither Hessian-vector products nor implicit linear solves. Experiments on nonconvex problems, representative learning tasks, and large-model post-training show that SUPRO realize remarkable performances while maintaining stable optimization.

1 INTRODUCTION

Modern machine learning increasingly involves hierarchical optimization, where decisions at one level depend on optimization outcomes at another. Bi-Level Optimization (BLO) provides a natural framework for such hierarchical dependence and has been widely used in hyperparameter optimization, data reweighting, meta-learning, neural architecture search, and robust learning (Franceschi et al., 2018; Ghadimi & Wang, 2018; Ren et al., 2018; Liu et al., 2019), in which hyperparameters, network architectures, or training-data weights are optimized through an underlying model-training process. Its computational importance is growing further with Large Language Models (LLMs), where data selection and post-training decisions directly influence model utility, safety, and alignment (Yu et al., 2025; Shen et al., 2025; Pan et al., 2025). The BLO problem is formulated as:

$$\min_{x,y} \; F(x,y) \qquad \text{s.t.} \qquad y \in S(x) := \arg\min_{y} f(x,y), \tag{1}$$

where the upper-level variable x represents the learning decision and the lower-level solution y represents the corresponding trained model or response.

Existing BLO algorithms can be broadly divided into three categories. Explicit differentiation methods unroll a finite lower-level trajectory and differentiate through the optimization dynamics (Franceschi et al., 2018; Shaban et al., 2019). Implicit differentiation methods differentiate the lower-level optimality condition and approximate an inverse-Hessian-vector product (Ghadimi & Wang, 2018; Lorraine et al., 2020; Ji et al., 2021). Value-function methods replace the lower-level solution constraint with a condition involving the optimal lower-level value, leading to penalty-based or constrained reformulations (Liu et al., 2021; 2022). Recent work has significantly improved the scalability of these approaches by developing fully first-order, Hessian-free, and single-loop algorithms, while also relaxing lower-level strong convexity (Kwon et al., 2023; Liu et al., 2023; 2022; Shen & Chen, 2023; Liu et al., 2024; Li et al., 2022). For example, MEHA introduces a Moreau-envelope reformulation and tracks its associated proximal response using a single-loop first-order procedure (Liu et al., 2024).

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.

Despite these advances, existing approaches still rely critically on repeatedly approximating the response of the current lower-level problem. Explicit differentiation requires enough unrolled steps to obtain a useful approximation of the lower-level dynamics. Implicit differentiation requires a sufficiently accurate lower-level point before solving the sensitivity system. Value-function approaches require an accurate approximation of the lower-level value or of a response attaining that value. Hessian-free single-loop schemes also maintain auxiliary responses whose approximation errors enter their descent analyses. As a result, reducing the number of lower-level steps generally increases the bias in the upper-level update, whereas improving the response accuracy introduces additional gradient evaluations. This tension becomes particularly severe when the lower-level variable is high-dimensional, for which every additional correction may require an expensive forward-backward pass. It also complicates theoretical analysis: lower-level errors accumulate along the bi-level trajectory, while nonunique lower-level solutions do not define a unique response trajectory. This motivates a more specific question: can learned temporal predictions improve response tracking while retaining a prediction-independent convergence guarantee? Our focus is on learning explicit response displacements and controlling their errors within a value-function penalty method.

To partially address these issues, we propose Safe Upper-guided Predictive Response Optimization (SUPRO). SUPRO maintains an auxiliary value response across outer iterations and updates a predictive model online using the coupled states of the lower-level iterate and the auxiliary response. It combines predicted response motion with a fixed number of gradient corrections. A candidate-selection rule compares reactive and predictor-corrected responses, while a subsequent learned proposal is accepted only if it satisfies lower-level and upper-level objective criteria. We establish a prediction-adaptive bound on cumulative response-tracking error and a non-asymptotic stationarity guarantee for the associated fixed value-function penalty. Experiments on nonconvex problems, typical learning, and large-model adaptation show earlier attainment of high accuracy and competitive final performance. Our contributions are as follows:

- We introduce SUPRO, which learns the temporal dynamics of lower-level responses across bi-level iterations. By predicting the responses and refining them through certified corrections, SUPRO turns successive lower-level solves into the response-tracking process.
- We develop a first-order implementation that replaces repeated lower-level solving with fixed-budget response tracking. By amortizing lower-level computation across bi-level iterations, SUPRO reduces correction overhead while maintaining certified optimization progress, making it suitable for high-dimensional lower-level variables.
- We establish the non-asymptotic convergence to stationarity of the value-function penalty for nonconvex lower-level problems. Our tracking analysis further quantifies how accurate predictions tighten the error bound.
- We evaluate SUPRO on nonconvex numerical cases, representative learning problems, and large-scale model tasks, measuring lower-level computation and bi-level optimization performance against existing first-order methods.

2 Related Works

Bi-Level Optimization Methods. A central challenge in BLO is controlling the error introduced by approximate lower-level solutions while limiting their computational cost. Unrolling-based methods expose a trade-off between differentiation accuracy and the time and memory required to retain the lower-level trajectory (Franceschi et al., 2018; Shaban et al., 2019). Implicit methods avoid storing this trajectory, but must approximate the sensitivity of the lower-level solution through Hessian-related computations (Ghadimi & Wang, 2018; Lorraine et al., 2020; Ji et al., 2021). Value-function and penalty formulations provide alternative ways to handle lower-level optimality, making the accuracy of auxiliary value or response estimates central to their convergence analysis (Liu et al., 2021; 2022; Shen & Chen, 2023). Subsequent developments pursue fully first-order computation, single-loop updates, and guarantees beyond lower-level strong convexity (Kwon et al., 2023; Liu et al., 2023; 2024; Huang, 2024; Lu & Mei, 2024). Our work complements these efforts by exploiting temporal information across successive lower-level problems. SUPRO learns to predict response motion and compares the corrected prediction against a reactive candidate, yielding a tracking guarantee that benefits from accurate predictions while retaining a prediction-independent fallback.

Learning to Optimize. Learning-to-optimize methods parameterize optimization rules with learned models that map optimization histories, such as gradients, parameter states, and recurrent statistics, to update directions (Andrychowicz et al., 2016; Li & Malik, 2017; Wichrowska et al., 2017; Metz et al., 2022). Early work showed that recurrent optimizers can learn update rules from gradient trajectories, while subsequent studies improved their scalability and generalization across optimization problems. These methods primarily learn how to update variables within a given optimization process. In contrast, our method focuses on the temporal coupling across a sequence of lower-level problems induced by changes in the upper variable.

Advanced Applications of BLO. BLO has recently emerged as a useful abstraction for adapting foundation models at the levels of data, parameters, and agentic workflows. For data selection and utilization, dynamic bi-level optimization learns batch-wise data weights according to the evolving training state of an LLM, SEAL learns a ranker that prioritizes safe and high-quality instruction-tuning data (Shen et al., 2025), and ScaleBiO develops a scalable first-order implementation for data reweighting in large language models (Pan et al., 2025). For parameter-efficient adaptation, BLO-SAM places the SAM parameters and a learnable prompt embedding at different optimization levels and optimizes them on separate data splits to alleviate few-shot overfitting (Zhang et al., 2024). Similarly, BiLoRA separates the optimization of pseudo singular vectors and singular values (Qiang et al., 2024), while BiDoRA asynchronously optimizes the directional and magnitude components of weight-decomposed low-rank updates using training and validation data, respectively (Qin et al., 2025). However, applying BLO to foundation models introduces a fundamental challenge: the lower-level response corresponds to a high-dimensional model state, and repeatedly correcting this response across outer iterations becomes increasingly expensive.

3 The Proposed Method

3.1 From Lower-Level Nested Solving to Value-Response Tracking

Consider the bi-level problem

$$\min_{x\in\mathbb{R}^{d_x},\, y\in\mathbb{R}^{d_y}} F(x,y) \quad \text{s.t.} \quad y \in \mathcal{S}(x) := \arg\min_{y\in\mathbb{R}^{d_y}} f(x,y), \tag{2}$$

where F and f denote the upper- and lower-level objectives, respectively. Let

$$v(x) := \min_{z} f(x,z) \tag{3}$$

be the lower-level value function. Its constraint can equivalently be written as $f(x,y) - v(x) \le 0$. A value-function penalty therefore takes the form

$$P_\lambda(x,y) := F(x,y) + \lambda\big[f(x,y) - v(x)\big], \qquad \lambda > 0. \tag{4}$$

In the first-order oracle setting considered here, we estimate $v(x)$ and its gradient using a lower-level response at the current x. At each upper-level iteration, approximating a response for the current x_t typically requires multiple lower-level optimization steps. Repeating this computation as x_t changes makes lower-level response estimation a major computational cost. Warm-starting reduces initialization error, but a purely reactive update still spends its budget correcting the entire response displacement after it occurs.

To reduce the lower-level computation required after each upper-level update, we introduce a persistent response variable ξ_t and approximate the value term at iteration t by

$$\widehat{v}_t := f(x_t,\xi_t), \qquad \widetilde{P}_\lambda(x_t,y_t;\xi_t) := F(x_t,y_t) + \lambda\big[f(x_t,y_t) - f(x_t,\xi_t)\big]. \tag{5}$$

The variables y_t and ξ_t play different roles. The former is the explicit lower-level variable appearing in the penalized bi-level problem; the latter is an auxiliary value response used to estimate $v(x_t)$ and its x-gradient. Rather than solving the lower-level problem anew at each x_t, SUPRO uses the previous response ξ_{t-1} to construct reactive and predictor-corrected candidates, then selects the accepted response ξ_t.

3.2 JOINT-STATE PREDICTIVE RESPONSE MODELING

SUPRO maintains two lower-level states with complementary roles. The variable y_t is the actual lower-level iterate produced by the task optimizer, whereas ξ_{t-1} is an auxiliary response used to approximate the lower-level value function. Their discrepancy provides information that is unavailable from the response history alone: differences in their parameters and lower-level gradients indicate how the tracked response relates to the current optimization trajectory. We therefore jointly encode y_t, ξ_{t-1}, and their coordinate-wise relations. Specifically, o_t^y and o_t^ξ summarize the position, gradient, gradient statistics, upper-variable displacement, and optimization time of the two states, while $r_t^{y\xi}$ describes their position and gradient differences. The joint state is then

$$H_t = \mathcal{E}_{\psi_t}\big(o_t^y, o_t^\xi, r_t^{y\xi}, H_{t-1}\big). \tag{6}$$

This representation conditions the prediction, correction, and response-selection updates on both the actual lower-level trajectory and the auxiliary trajectory. Details are deferred to Appendix A.3.2.

The learned model does not replace a first-order response step. Let $g_t^0 := \nabla_z f(x_t, \xi_{t-1})$ and $p_t^{\mathrm{GD}} := -\eta_p g_t^0$. The prediction head generates a bounded residual $\delta p_t = \mathcal{D}_{\psi_t}^{\mathrm{P}}(H_t)$, scaled blockwise relative to p_t^{GD}. The complete prediction displacement is

$$p_t = \begin{cases} p_t^{\mathrm{GD}} + \delta p_t, & \langle g_t^0, p_t^{\mathrm{GD}} + \delta p_t\rangle \le -c\eta_p \|g_t^0\|^2, \\ p_t^{\mathrm{GD}}, & \text{otherwise}, \end{cases} \qquad \widehat{\xi}_t = \xi_{t-1} + p_t. \tag{7}$$

The displacement p_t consists of a gradient step and, when it passes the direction test, a bounded learned residual. If the residual fails this test, p_t reverts to the gradient step.

3.3 PREDICTION, CORRECTION, AND SAFEGUARDED SELECTION

The predicted response may still have a large lower-level error at the current upper iterate. Accepting predictions directly would therefore couple correctness to the learned optimizer. Conversely, always discarding a prediction after a fixed reactive step would remove the very advantage of learning response motion. We resolve this tension with two independently verifiable correction paths. To provide a prediction-independent route to progress while retaining the benefit of accurate forecasts, let

$$\mathsf{GD}_{x,\eta}^K(u) := u^K, \qquad u^{k+1} = u^k - \eta \nabla_z f(x, u^k), \quad u^0 = u, \tag{8}$$

denote K lower-gradient steps. We construct two certified candidates:

$$\xi_t^{\mathrm{R}} := \mathsf{GD}_{x_t,\eta_b}^{K_b}(\xi_{t-1}), \qquad \xi_t^{\mathrm{P}} := \mathsf{GD}_{x_t,\eta_c}^{K_c}(\widehat{\xi}_t). \tag{9}$$

The reactive candidate ξ_t^{R} ignores the predictor and updates the previous accepted response on the current lower objective. The predictor-corrected candidate ξ_t^{P} instead optimizes only the error left after prediction. In the default implementation, $K_b = K_c = 1$; larger but iteration-independent values can be used when a stronger contraction is required. Importantly, ξ_t^{P} contains the prediction and its analytic gradient correction, but not the learned residual c_t introduced below.

The two candidates yield complementary guarantees. The reactive branch is controlled by how far the target response moves, whereas the predictive branch is controlled by how much of that motion the predictor fails to explain. We retain the better certified point,

$$\xi_t^{\mathrm{safe}} \in \operatorname*{arg\,min}_{\zeta \in \{\xi_t^{\mathrm{R}}, \xi_t^{\mathrm{P}}\}} f(x_t, \zeta). \tag{10}$$

Using both candidates is essential. The reactive candidate provides a prediction-independent fallback: its error depends on the displacement between successive target responses. The predictor-corrected candidate starts closer to the new target whenever p_t explains that displacement, so its correction error depends on the prediction residual instead. Selecting the lower-valued candidate retains whichever explanation is better at the current iteration. When the lower solution is nonunique, minimizing $f(x_t, \cdot)$ alone does not distinguish among lower-near-optimal auxiliary responses. Identifying a particular exact minimizer is not required for the value-function update; we use the upper gradient only after lower-level correction as an upper-aware tie-breaking signal. Define

$$g_t^{\mathrm{L}} := \nabla_z f(x_t, \xi_t^{\mathrm{P}}), \qquad g_t^{\mathrm{U}} := \nabla_z F(x_t, \xi_t^{\mathrm{P}}), \tag{11}$$

and define a regularized upper-gradient feature,

$$\widetilde{g}_t^{\mathrm{U}} := g_t^{\mathrm{U}} - \frac{\langle g_t^{\mathrm{U}}, g_t^{\mathrm{L}} \rangle}{\|g_t^{\mathrm{L}}\|^2 + \varepsilon} g_t^{\mathrm{L}}. \tag{12}$$

The regularized projection suppresses the component of the upper gradient aligned with the lower gradient while remaining well defined near lower-level stationarity.

At a lower minimizer, it equals the upper gradient and need not lie in a tangent space of the minimizer set. A lower-stationarity gate

$$a_t^{\mathrm{U}} := \frac{\tau_u}{\|g_t^{\mathrm{L}}\| + \tau_u} \tag{13}$$

suppresses upper guidance when the lower residual is large. The correction and selection heads reuse H_t and receive detached phase-specific gradient features: $c_t = \mathcal{D}_{\psi_t}^{\mathrm{C}}(H_t)$ and $u_t = \mathcal{D}_{\psi_t}^{\mathrm{U}}(H_t)$. The full learned proposal is

$$\xi_t^{\mathrm{learn}} := \xi_t^{\mathrm{P}} + c_t + a_t^{\mathrm{U}} u_t. \tag{14}$$

The residual head is intended to correct prediction errors, while the upper head proposes auxiliary responses with lower upper-objective values. The learned proposal is accepted only if it preserves certified lower-level quality and is no worse for response selection:

$$\xi_t = \begin{cases} \xi_t^{\mathrm{learn}}, & f(x_t, \xi_t^{\mathrm{learn}}) \le f(x_t, \xi_t^{\mathrm{safe}}) + \epsilon_t, \\ & F(x_t, \xi_t^{\mathrm{learn}}) \le F(x_t, \xi_t^{\mathrm{safe}}), \qquad \epsilon_t \ge 0. \\ \xi_t^{\mathrm{safe}}, & \text{otherwise}, \end{cases} \tag{15}$$

Irrespective of the predictor architecture, corrections, or online training dynamics, this rule implies

$$f(x_t, \xi_t) \le \min\{ f(x_t, \xi_t^{\mathrm{R}}), f(x_t, \xi_t^{\mathrm{P}}) \} + \epsilon_t. \tag{16}$$

Equation (16) is the algorithmic invariant used by our analysis. It cleanly separates a potentially aggressive learned proposal from a deterministic lower-level certificate. The upper comparison can bias the auxiliary response toward an upper-favorable candidate, but it is not an optimistic lower-level solver; the lower comparison ensures that this bias cannot bypass the certified correction paths. Thus upper guidance may improve response selection, while the lower-level safeguard carries the worst-case guarantee.

3.4 First-Order Online Adaptation and Bi-level Updates

The mapping from first-order history to response motion depends on the current objective, optimization stage, and realized response trajectory. A predictor trained only offline may therefore face a substantial deployment shift. At the same time, the correction and response selection already reveal whether the prediction was useful, providing supervision along the trajectory itself.

We therefore adapt the response predictor on the current problem rather than assuming that a fixed optimizer generalizes across tasks. Specifically, we optimize a composite objective

$$\begin{aligned} \mathcal{L}_t^{\mathrm{opt}} &= \omega_{\mathrm{val}} \mathcal{L}_t^{\mathrm{val}} + \omega_{\mathrm{dom}} \mathcal{L}_t^{\mathrm{dom}} + \omega_{\mathrm{sel}} \mathcal{L}_t^{\mathrm{sel}} \\ &\quad + \omega_{\mathrm{pred}} \mathcal{L}_t^{\mathrm{pred}} + \omega_{\mathrm{stab}} \mathcal{L}_t^{\mathrm{stab}} + \omega_{\mathrm{gain}} \mathcal{L}_t^{\mathrm{gain}}, \qquad \psi_{t+1} = \psi_t - \beta_t \nabla_\psi \mathcal{L}_t^{\mathrm{opt}}. \end{aligned} \tag{17}$$

Here, $\mathcal{L}_t^{\mathrm{opt}}$ is the total online objective used to train the response optimizer. The term $\mathcal{L}_t^{\mathrm{val}}$ is the lower-level objective of the learned proposal; $\mathcal{L}_t^{\mathrm{dom}}$ is the certified-domination loss that compares the proposal with ξ_t^{safe}; $\mathcal{L}_t^{\mathrm{sel}}$ is the upper-level response-selection loss; $\mathcal{L}_t^{\mathrm{pred}}$ is the prediction-distillation loss that transfers the subsequently observed correction into the predictor; $\mathcal{L}_t^{\mathrm{stab}}$ suppresses unnecessary response motion near stationarity; and $\mathcal{L}_t^{\mathrm{gain}}$ trains the predictor-corrected branch to improve over the reactive candidate. The nonnegative coefficients $\omega_{\mathrm{val}}, \omega_{\mathrm{dom}}, \omega_{\mathrm{sel}}, \omega_{\mathrm{pred}}, \omega_{\mathrm{stab}}, \omega_{\mathrm{gain}}$ weight the six objectives. The symbol ψ_t denotes the learned optimizer parameters at iteration t, ∇_ψ denotes differentiation with respect to these parameters, and $\beta_t > 0$ is their online learning rate. Exact loss definitions and the sparse large-model update schedule are given in Appendix A.3.3.

Online adaptation uses a first-order surrogate computation graph. The forward correction remains gradient descent, while the backward map omits derivatives of the lower gradient. Thus the optimizer update requires first derivatives of the objectives and Jacobian products through the learned

map. After accepting ξ_t, we update the explicit bi-level variables using the tracked penalty. The response ξ_t is held fixed when computing the outer partial derivatives; differentiating through its construction would introduce terms outside this analysis:

$$\bar{y}_{t+1} := y_t - \eta_y \Big[\nabla_y F(x_t, y_t) + \lambda \nabla_y f(x_t, y_t)\Big], \tag{18}$$

$$x_{t+1} := x_t - \eta_x \Big[\nabla_x F(x_t, y_t) + \lambda \nabla_x f(x_t, y_t) - \lambda \nabla_x f(x_t, \xi_t)\Big]. \tag{19}$$

SUPRO does not replace the task optimizer in (18). After its base update, we transfer a relatively bounded amount of information from the certified response:

$$a_t^{\mathrm{mix}} := \min\left\{1, \frac{\nu_t \|\bar{y}_{t+1} - y_t\|}{\max\{\|\xi_t - \bar{y}_{t+1}\|, \varepsilon\}}\right\}, \quad m_t := a_t^{\mathrm{mix}}(\xi_t - \bar{y}_{t+1}), \quad y_{t+1} := \bar{y}_{t+1} + m_t, \tag{20}$$

where $0 \le \nu_t \le \nu_{\max} < 1$. By construction, $\|m_t\| \le \nu_t \|\bar{y}_{t+1} - y_t\|$; hence response transfer cannot dominate the base lower-variable update. **Algorithm 1** summarizes the procedure. Here, single-loop means that K_b and K_c remain fixed across outer iterations, rather than that candidate construction uses only $K_b + K_c$ model-gradient evaluations.

4 Non-asymptotic Theory of Predictive Response Tracking

Assumption 4.1 (Uniform lower-level geometry)**.** On an open convex set $\mathcal{X}$ containing all upper iterates, f is continuously differentiable. For every $x \in \mathcal{X}$, $\mathcal{S}(x)$ is nonempty, $f(x, \cdot)$ is L_z-smooth on $\mathbb{R}^{d_y}$, and, for a uniform $0 < \mu \le L_z$,

$$\tfrac{1}{2}\|\nabla_z f(x, z)\|^2 \ge \mu G(x, z), \qquad G(x, z) := f(x, z) - v(x), \quad z \in \mathbb{R}^{d_y}. \tag{21}$$

In particular, $f(x, \cdot)$ need not be convex, and the solution set $\mathcal{S}(x)$ need not be a singleton. Its global lower-variable scope ensures validity at every predicted point and along the correction trajectories. A local version additionally requires these trajectories to remain in a region with the same properties. It implies $\mathrm{dist}(z, \mathcal{S}(x))^2 \le 2G(x, z)/\mu$ (Appendix A.5.1). Set $\kappa = 2/\mu$, choose any nearest comparator $z_t^\star \in \mathrm{proj}_{\mathcal{S}(x_t)}(\xi_t)$, and define

$$\Delta_t = G(x_t, \xi_t), \quad d_t^\star = z_t^\star - z_{t-1}^\star, \quad V_T = \sum_{t=1}^{T} \|d_t^\star\|^2, \quad E_T = \sum_{t=1}^{T} \|p_t - d_t^\star\|^2. \tag{22}$$

These comparators are analytical objects tied to the accepted responses and need not form the differentiable branch used for motivation.

Prediction-adaptive tracking. We next quantify how the candidate-selection rule adapts to the accuracy of the predicted response displacement. At iteration t, the reactive branch, indexed by b, incurs an error determined by the true response displacement $d_t^\star$, whereas the predictor-corrected branch, indexed by c, depends on the prediction residual $p_t - d_t^\star$, where p_t is the complete gradient-plus-residual displacement in Eq. (7). Thus, the two branches provide complementary controls of the tracking error.

For each branch $j \in \{b, c\}$, let $\eta_j \in (0, 1/L_z]$ and K_j denote the step size and the number of lower-level gradient corrections, respectively. For any $\theta_j > 0$, define

$$q_j = \big[1 - 2\mu\eta_j(1 - L_z\eta_j/2)\big]^{K_j}, \qquad \rho_j = \frac{q_j L_z \kappa}{2}(1 + \theta_j), \qquad c_j = \frac{q_j L_z}{2}(1 + \theta_j^{-1}). \tag{23}$$

Here, L_z is the lower-level smoothness constant, μ is the Polyak–Łojasiewicz constant, and κ is the response error-bound constant. The quantity q_j is the contraction factor after K_j gradient corrections, while ρ_j and c_j account for the propagation of the previous tracking error and the response displacement, respectively. The parameters θ_j are auxiliary constants used to balance these terms.

Theorem 4.2 (Prediction-adaptive response tracking)**.** *Suppose Assumption 4.1 and the certified candidate-selection condition (16) hold. Let* $\Delta_t := f(x_t, \xi_t) - v(x_t)$ *and* $\bar{\rho} := \max\{\rho_b, \rho_c\} <$

1. *Define the cumulative response-path variation* $V_T := \sum_{t=1}^{T} \|d_t^\star\|^2$, *prediction error* $E_T := \sum_{t=1}^{T} \|p_t - d_t^\star\|^2$, *and safeguard error* $\mathcal{E}_T := \sum_{t=1}^{T} \epsilon_t$. *Then, for every* $T \geq 1$,

$$\sum_{t=1}^{T} \Delta_t \leq \frac{\bar{\rho}\Delta_0 + \sum_{t=1}^{T} \min\{c_b\|d_t^\star\|^2,\, c_c\|p_t - d_t^\star\|^2\} + \mathcal{E}_T}{1-\bar{\rho}} \leq \frac{\bar{\rho}\Delta_0 + \min\{c_b V_T, c_c E_T\} + \mathcal{E}_T}{1-\bar{\rho}} =: \mathcal{B}_T. \tag{24}$$

Moreover,

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla_z f(x_t, \xi_t)\|^2 \leq \frac{2L_z \mathcal{B}_T}{T}. \tag{25}$$

Hence, the tracking bound adapts to the smaller of the response-path variation and the cumulative prediction error.

The tracking bound adaptively exploits the tighter control between the reactive and predictor-corrected branches. Specifically, it is governed by the response-path variation when predictions are inaccurate, and by the cumulative prediction error when they are informative. If $\min\{c_b V_T, c_c E_T\} + \mathcal{E}_T$ remains bounded, Theorem 4.2 yields an $\mathcal{O}(T^{-1})$ rate for the averaged squared lower-level stationarity residual. The contraction condition $\bar{\rho} < 1$ can be ensured with a sufficiently large but fixed correction budget; an explicit sufficient choice is given in Appendix A.5.1.

Assumption 4.3 (Outer regularity)**.** On $\mathcal{X} \times \mathbb{R}^{d_y}$, $\nabla_x f$ is L_{xx}-Lipschitz in x and L_{xz}-Lipschitz in z, and $\nabla_z f$ is L_{zx}-Lipschitz in x. For fixed $\lambda > 0$, F is differentiable and P_λ is L_P-smooth and bounded below by $P_{\inf}$ on this domain.

These assumptions ensure that the value function remains differentiable even when the lower-level minimizer is nonunique, with $\nabla v(x) = \nabla_x f(x, z^\star)$ for any $z^\star \in \mathcal{S}(x)$; see Appendix A.5.2. We next couple the outer stepsize with the contraction of the lower-level tracking dynamics. Let $\eta_x = \eta_y = \alpha > 0$ and define $A_\lambda = 2\lambda^2 L_{xz}^2/\mu$ and $C_\lambda = A_\lambda + 2L_z$. Choose $\vartheta > 0$ such that the reactive correction is contractive, $r := q_b(1+\vartheta) < 1$, and set

$$B := \frac{q_b(1+\vartheta^{-1})L_{xz}^2}{2\mu}, \qquad \omega := \frac{\alpha C_\lambda}{1-r}, \qquad \alpha \leq \min\left\{\frac{1}{2L_P}, \sqrt{\frac{1-r}{4C_\lambda B}}\right\}. \tag{26}$$

For the response-guided mixing step, assume $0 \leq \nu_t \leq \nu_{\max} < 1$ and define

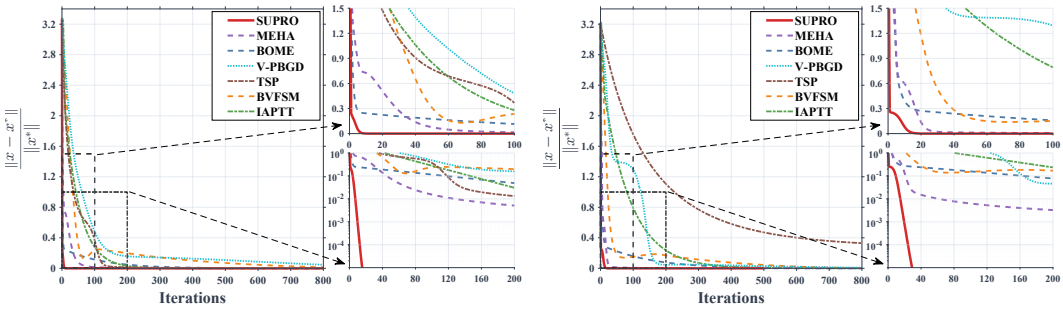
$$\phi_{\min} := (1-\nu_{\max})\left[1 - \frac{L_P\alpha}{2}(1-\nu_{\max})\right] \geq \frac{1}{2}. \tag{27}$$

Here, the stepsize restriction ensures that the descent of the penalty objective dominates the tracking error induced by the changing upper-level variable. The second restriction is omitted when $B = 0$.

Theorem 4.4 (Joint convergence with response-guided mixing)**.** *Under Assumptions 4.1 and 4.3, the certified selection condition, and Eqs. (26)– (27), let* $\Psi_1 = P_\lambda(x_1, y_1) + \omega\Delta_1$. *Then*

$$\frac{1}{T}\sum_{t=1}^{T}\left[\|\nabla P_\lambda(x_t, y_t)\|^2 + \|\nabla_z f(x_t, \xi_t)\|^2\right] \leq \frac{2(\Psi_1 - P_{\inf}) + 2\omega\mathcal{E}_T}{\alpha T}. \tag{28}$$

Condition (27) follows directly from the relative displacement bound in (20); it does not require the mixing direction itself to be a descent direction. In particular, $\alpha \leq 1/(2L_P)$ and $\nu_{\max} \leq \sqrt{2}-1$ are sufficient, so the practical cap $\nu_{\max} = 0.3$ lies within this range. No boundedness assumption on V_T or E_T is needed here. Any fixed $K_b \geq 1$ admits an $r < 1$ and a sufficiently small positive α; the prediction-dependent bound additionally requires $\bar{\rho} < 1$ and can be uninformative when comparator motion is large. Summable safeguard tolerances give an $O(T^{-1})$ joint squared-residual rate. This concerns the fixed penalty, not exact stationarity of the original bi-level problem. If $\|\nabla_y F(x_t, y_t)\| \leq M$, then $G(x_t, y_t) \leq (M + \|\nabla P_\lambda(x_t, y_t)\|)^2/(2\mu\lambda^2)$. Appendices A.5.2–A.5.3 provide the proofs, the penalty interpretation, and extensions to unequal outer steps and quantified evaluation errors.



(a) $d_y = 10$ (b) $d_y = 1000$

Figure 1: Convergence comparison on the nonconvex numerical problem under different lower-level dimensions. We compare SUPRO with existing bi-level optimization methods in terms of the relative solution error $\|x - x^*\|/\|x^*\|$. The right panels provide enlarged views of the early iterations.

Table 1: Comparison of convergence speed for few-shot learning (10-way and 20-way) and accuracy for data hyper-cleaning tasks.

Methods	10-Way		20-Way		FashionMNIST		MNIST	
	Acc. (%)	Time (S)	Acc. (%)	Time (S)	Acc. (%)	F1 score	Acc. (%)	F1 score
BOME	90.27	56.17	90.44	72.50	82.62	92.38	87.28	93.23
TSP	90.08	1015.13	90.07	5442.20	81.93	91.56	88.55	93.86
MEHA	90.27	46.92	90.06	56.19	82.94	93.21	88.17	94.04
IAPTT	90.03	463.45	90.30	438.82	83.75	94.04	87.46	93.49
BVFSM	90.23	343.17	**90.63**	317.15	84.03	**94.14**	90.11	**96.20**
V-PBGD	90.64	59.29	90.14	149.09	82.06	88.06	90.07	87.08
SUPRO	**90.73**	**32.72**	90.51	**35.71**	**84.35**	94.12	**90.14**	95.42

5 Experiments

We evaluate SUPRO on high-dimensional nonconvex, data hyper-cleaning, meta-learning, and large-model post-training tasks against MEHA, BOME, V-PBGD, TSP (Lu, 2025), BVFSM, and IAPTT (Liu et al., 2026). Appendix A.2 provides more experimental results to demonstrate the effectiveness.

Lower-level non-convex numerical case. We first evaluate SUPRO on a high-dimensional bi-level problem with a nonconvex and nonunique lower-level solution set. Following the widely-used nonconvex case (Liu et al., 2021), we consider

$$\min_{x\in\mathbb{R},\, y\in\mathbb{R}^n} \|x-a\|^2 + \|y - a\mathbf{e} - c\|^2 \quad \text{s.t.} \quad y_i \in \arg\min_{u\in\mathbb{R}} \sin(x+u-c_i),\ \forall i, \tag{29}$$

where $a \in \mathbb{R}$, $c \in \mathbb{R}^n$, and $\mathbf{e}$ is the all-ones vector. The periodic lower-level objective induces multiple lower-level minimizers, while the closed-form optimum allows us to measure the relative solution error $\|x - x^\star\|/\|x^\star\|$ exactly. We vary the lower-level dimension over $d_y \in \{10, 100, 500, 1000\}$. Figure 1 shows representative convergence trajectories for $d_y = 10$ and $d_y = 1000$. SUPRO rapidly reduces the relative solution error in both regimes and, unlike several competing methods, remains stable after reaching a low-error region. This behavior is particularly pronounced at $d_y = 1000$, where SUPRO reaches an error below 10^{-3} within approximately 20 iterations and continues to decrease to the 10^{-6} scale. Table 2 quantifies this behavior across all dimensions. Specifically, at iteration 500, SUPRO attains the smallest relative errors, consistently lower than those of the competing methods. Overall, these results show that SUPRO is with stable high-accuracy convergence as the lower-level dimension increases.

Few-shot Learning. We evaluate SUPRO on few-shot classification with 10-way and 20-way tasks (Wang et al., 2020). Let $\mathcal{D}_{\text{sup}}$ and $\mathcal{D}_{\text{qry}}$ denote the support and query sets, respectively. The model adaptation is formulated as

$$\min_x \ \ell\big(x, y^*(x); \mathcal{D}_{\text{qry}}\big), \qquad \text{s.t.} \quad y^*(x) \in \arg\min_y \ell\big(x, y; \mathcal{D}_{\text{sup}}\big).$$

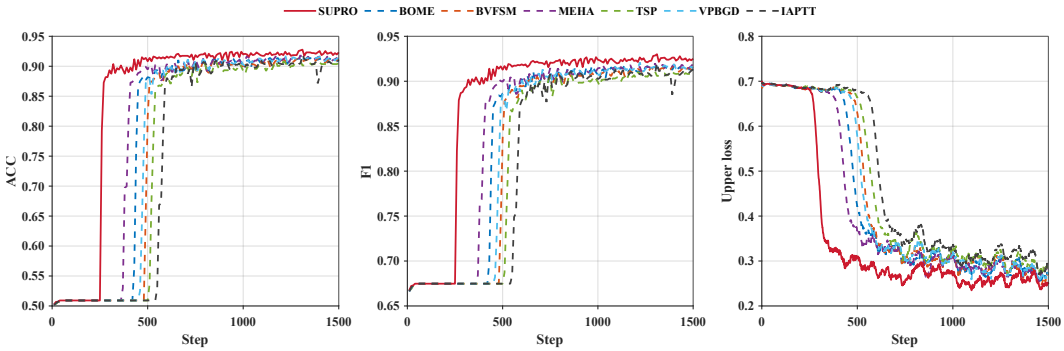


Figure 2: Performance comparison of different bi-level optimization methods on the BiDoRA task. From left to right, the panels show validation accuracy (ACC), F1 score, and upper-level loss.

As reported in Table 1, SUPRO achieves the highest accuracy on the 10-way task with the shortest runtime, and remains competitive in the more challenging 20-way setting while substantially reducing optimization time. These results indicate that predictive response tracking can accelerate task adaptation without sacrificing few-shot generalization.

Data hyper-cleaning. We further consider data hyper-cleaning (Liu et al., 2024), where the upper-level variable x assigns learnable weights to potentially corrupted training samples and the lower-level variable y denotes the classifier parameters. The problem is formulated as

$$\min_x \ \ell\big(y^*(x); \mathcal{D}_{\text{val}}\big), \qquad \text{s.t.} \quad y^*(x) \in \arg\min_y \frac{1}{N} \sum_{i=1}^{N} \sigma(x_i)\, \ell\big(y; z_i\big),$$

where $\sigma(x_i)$ maps the hyperparameter to a nonnegative sample weight. On FashionMNIST and MNIST, SUPRO achieves the highest accuracy while maintaining competitive F1 scores in Table 1, showing that its response-tracking mechanism remains effective for sample-reweighting problems with coupled upper- and lower-level updates.

Large model post-training. We further evaluate SUPRO on Parameter-Efficient Fine-Tuning (PEFT) of pretrained language models (Hu et al., 2021; Chen et al., 2024). Following BiDoRA (Qin et al., 2025), the upper-level variable x parameterizes the magnitudes of the Query and Value projections, while the lower-level variable y contains their rank-4 low-rank directional parameters and the classification head. The resulting problem is formulated as

$$\min_x \ell(x, y^*(x); \mathcal{D}_{\text{val}}), \qquad \text{s.t.} \quad y^*(x) \in \arg\min_y \ell(x, y; \mathcal{D}_{\text{tr}}).$$

This task provides a representative large-scale bi-level setting, since the lower-level response evolves continuously with the outer variable and repeated correction can incur substantial model-evaluation cost. Figure 2 compares SUPRO with representative bi-level optimization baselines. SUPRO enters the high-accuracy and high-F1 regime substantially earlier than the competing methods and maintains competitive final predictive performance. Meanwhile, its upper-level loss decreases earlier and remains generally lower over the optimization trajectory. The improvement is consistent with the motivation of predictive response tracking: SUPRO anticipates its motion and allocates the correction budget to the remaining prediction error. The results therefore indicate that SUPRO can reduce the optimization delay induced by expensive lower-level adaptation.

6 Conclusion

We presented SUPRO, a predictive response-tracking approach for scalable bi-level optimization that amortizes lower-level computation across outer iterations. SUPRO combines learned response prediction with certified first-order corrections, allowing accurate predictions to improve tracking while retaining a prediction-independent safeguard. Under a uniform PL condition, we established non-asymptotic convergence to stationarity of a fixed value-function penalty for nonconvex lower-level problems. The method is fully first-order, single-loop, and requires neither Hessian-vector products nor implicit linear solves. Across numerical problems, typical learning tasks, and large-model post-training settings, SUPRO exhibits stable optimization and computational efficiency.

REFERENCES

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. In *International Conference on Learning Representations*, volume 2024, pp. 8220–8238, 2024.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1568–1577. PMLR, 2018.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Feihu Huang. Optimal Hessian/Jacobian-free nonconvex-PL bilevel optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19598–19621. PMLR, 2024.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4882–4892. PMLR, 2021.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D. Nowak. A fully first-order method for stochastic bilevel optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18083–18113. PMLR, 2023.

Junyi Li, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7426–7434, 2022.

Ke Li and Jitendra Malik. Learning to optimize. In *International Conference on Learning Representations*, 2017.

Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. BOME! bilevel optimization made easy: A simple first-order approach. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17248–17262, 2022.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6882–6892. PMLR, 2021.

Risheng Liu, Yaohua Liu, Wei Yao, Shangzhi Zeng, and Jin Zhang. Averaged method of multipliers for bi-level optimization without lower-level strong convexity. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21839–21866. PMLR, 2023.

Risheng Liu, Zhu Liu, Wei Yao, Shangzhi Zeng, and Jin Zhang. Moreau envelope for nonconvex bi-level optimization: A single-loop and hessian-free solution strategy. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31566–31596. PMLR, 2024.

Risheng Liu, Yaohua Liu, Shangzhi Zeng, and Jin Zhang. Augmenting iterative trajectory for bilevel optimization: Methodology, analysis and extensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4239–4256, 2026.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1540–1552. PMLR, 2020.

Songtao Lu. TSP: A two-sided smoothed primal-dual method for nonconvex bilevel optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 40665–40708, 2025.

Zhaosong Lu and Sanyou Mei. First-order penalty methods for bilevel optimization. *SIAM Journal on Optimization*, 34(2):1937–1969, 2024.

Luke Metz, James Harrison, C. Daniel Freeman, Amil Merchant, Lucas Beyer, James Bradbury, Naman Agrawal, Ben Poole, Igor Mordatch, Adam Roberts, and Jascha Sohl-Dickstein. VeLO: Training versatile learned optimizers by scaling up. *arXiv preprint arXiv:2211.09760*, 2022.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. ScaleBiO: Scalable bilevel optimization for LLM data reweighting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31959–31982, Vienna, Austria, 2025. Association for Computational Linguistics.

Rushi Qiang, Ruiyi Zhang, and Pengtao Xie. BiLoRA: A bi-level optimization framework for overfitting-resilient low-rank adaptation of large pre-trained models. *arXiv preprint arXiv:2403.13037*, 2024. URL `https://arxiv.org/abs/2403.13037`.

Peijia Qin, Ruiyi Zhang, and Pengtao Xie. BiDoRA: Bi-level optimization-based weight-decomposed low-rank adaptation. *Transactions on Machine Learning Research*, 2025.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4334–4343. PMLR, 2018.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1723–1732. PMLR, 2019.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 30992–31015, 2023.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *International Conference on Learning Representations*, 2025.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Olga Wichrowska, Niru Maheswaranathan, Matthew W. Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3751–3760. PMLR, 2017.

Yang Yu, Kai Han, Hang Zhou, Yehui Tang, Kaiqi Huang, Yunhe Wang, and Dacheng Tao. LLM data selection and utilization via dynamic bi-level optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72995–73008. PMLR, 2025.

Li Zhang, Youwei Liang, Ruiyi Zhang, Amirhosein Javadi, and Pengtao Xie. BLO-SAM: Bi-level optimization based finetuning of the segment anything model for overfitting-preventing semantic segmentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59289–59309. PMLR, 2024.

A APPENDIX

A.1 MORE EXPERIMENTAL RESULTS

Table 2: Computational efficiency and convergence accuracy across different problem dimensions. Lower values are better for all metrics.

n	SUPRO	IAPTT	MEHA	BOME	V-PBGD	TSP	BVFSM
(a) Time to first reach relative error 10^{-3} *(s)* $\downarrow$							
10	**0.374**	4.308	2.023	22.574	4.953	1.199	$15.864^{\dagger}$
100	**1.209**	–	–	–	6.254	–	13.868
500	**1.167**	–	2.768	33.721	4.153	–	$14.226^{\dagger}$
1000	**0.923**	6.567	1.683	25.271	2.728	1.076	$14.081^{\dagger}$
(b) Iterations to first reach relative error 10^{-3} $\downarrow$							
10	**10**	355	756	648	1682	1909	$926^{\dagger}$
100	**25**	–	–	–	2066	–	786
500	**21**	–	1431	1627	1355	–	$805^{\dagger}$
1000	**20**	654	689	805	1200	1965	$808^{\dagger}$
(c) Relative error at iteration 500 $\downarrow$							
10	$\mathbf{2.81 \times 10^{-6}}$	3.93×10^{-5}	1.61×10^{-3}	3.61×10^{-3}	1.08×10^{-1}	4.01×10^{-3}	7.02×10^{-2}
100	$\mathbf{1.69 \times 10^{-5}}$	4.57×10^{0}	1.42×10^{-2}	1.05×10^{0}	1.57×10^{-1}	4.69×10^{1}	4.43×10^{-2}
500	$\mathbf{3.46 \times 10^{-6}}$	1.91×10^{26}	2.72×10^{-3}	4.88×10^{-2}	5.48×10^{-2}	4.707×10^{-1}	4.45×10^{-2}
1000	$\mathbf{1.85 \times 10^{-6}}$	6.34×10^{-3}	1.36×10^{-5}	8.49×10^{-3}	2.95×10^{-2}	4.78×10^{-1}	4.45×10^{-2}

Notes. Bold values indicate the best result for each problem dimension. "–" indicates that the method does not reach the target relative error within the 3000-iteration evaluation horizon. † The method reaches the target threshold transiently, but its error subsequently rises above 10^{-3}. For $d_y = 100$ and $d_y = 500$, all methods use the same hyperparameter configuration as in the $d_y = 1000$ setting.

Table 2 quantifies this behavior across all dimensions. SUPRO reaches the 10^{-3} threshold in only 10, 25, 21, and 20 iterations for $d_y = 10, 100, 500$, and 1000, respectively, showing that its iteration requirement remains nearly insensitive to the increase in lower-level dimension. At iteration 500, SUPRO attains relative errors between 1.85×10^{-6} and 1.69×10^{-5} across all settings, consistently lower than those of the competing methods. Overall, these results show that SUPRO is with stable high-accuracy convergence as the lower-level dimension increases.

A.2 ABLATION STUDIES

We conduct BiDoRA ablations along the component axes summarized in Section 5, followed by ablations of the learned response architecture. All variants are evaluated under the same experimental configuration and a 500-iteration training budget with a fixed random seed. We report the peak validation accuracy, the first iteration at which the peak is attained, and the first iteration reaching 90% validation accuracy. Since several variants achieve comparable peak accuracies, the latter metric additionally characterizes convergence efficiency in the high-accuracy regime.

Response-tracking components. We first ablate the main mechanisms involved in constructing, safeguarding, and transferring the auxiliary response. The results are summarized in Table 3.

Table 3: Ablation of the response-tracking components on BiDoRA over 500 bi-level iterations with a fixed random seed. Iter.@Peak denotes the first iteration attaining the reported peak validation accuracy, and Iter.@90% denotes the first iteration reaching 90% validation accuracy.

Variant	Peak Acc. (%) ↑	Iter.@Peak	Iter.@90% ↓
SUPRO	91.514	470	**320**
w/o Learned Network	89.220	500	–
w/o Response Mixing	91.170	450	350
w/o Learned Prediction	91.284	430	350
Linear Extrapolation	91.284	430	350
w/o R Candidate	**91.743**	430	390
Two-step GD Reference	91.170	450	350

Variant definitions. For *w/o Learned Network*, we remove all learned P/C/U outputs: the prediction reduces to its gradient-based component and the learned corrections are set to zero, while the reactive candidate, safeguarded R/P selection, and response-guided mixing are retained. For *w/o Learned Prediction*, only the learned prediction residual is disabled, such that the prediction reduces to its gradient-based component, while the learned C/U heads, candidate safeguard, and response-guided mixing remain unchanged. *Linear Extrapolation* replaces the learned prediction residual with a deterministic extrapolation based on the previous response displacement, while retaining the subsequent correction and selection mechanisms. For *w/o Response Mixing*, the response-guided displacement of the explicit lower iterate is disabled while the response construction and safeguard remain unchanged. For *w/o R Candidate*, the reactive candidate is removed from the safeguarded comparison, leaving the predictor-corrected branch as the reference response candidate. Finally, the *Two-step GD Reference* removes the learned network, candidate selection, and response-guided mixing, and directly applies two consecutive gradient-based response updates. It therefore serves as a gradient-only reference rather than a single-component ablation.

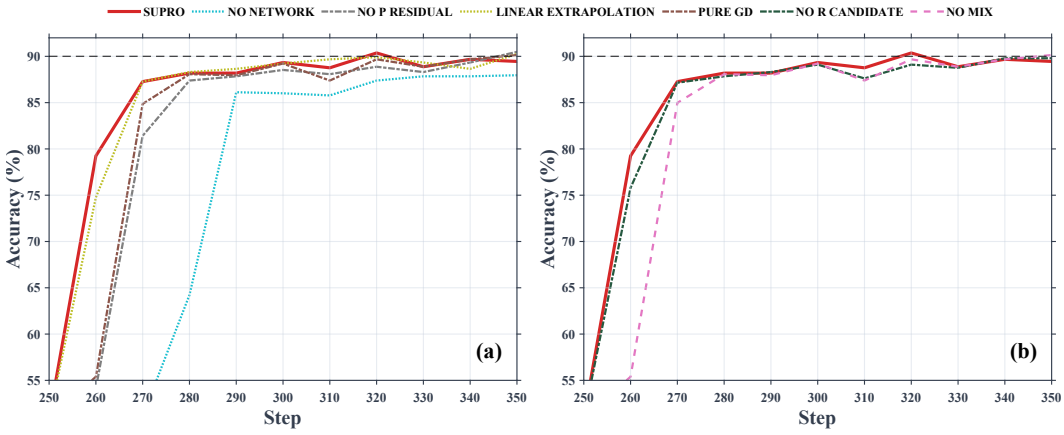


Figure 3: Validation accuracy trajectories for the response-tracking component ablations on BiDoRA. The left panel compares the learned predictive response with gradient-based and simple extrapolation alternatives, while the right panel examines the effects of the reactive candidate and response-guided mixing.

The full SUPRO reaches 90% validation accuracy after 320 iterations, earlier than all other component variants. Removing the learned response network causes the largest degradation: its peak accuracy drops to 89.220%, and it does not reach the 90% threshold within the 500-iteration budget. Disabling only the learned prediction residual or replacing it with linear extrapolation delays the threshold to 350 iterations, with both variants attaining a peak accuracy of 91.284%. The full model therefore improves both the convergence speed and the attainable peak over these two prediction alternatives, although the difference in peak accuracy is moderate.

Response-guided mixing exhibits a similar effect: removing it delays the 90% threshold from 320 to 350 iterations and decreases the peak accuracy from 91.514% to 91.170%. The two-step gradient reference reaches the same threshold at 350 iterations and the same peak accuracy, while the complete removal of the learned network results in substantially poorer performance. Together, these comparisons suggest that the learned response mechanism and its transfer to the explicit lower iterate mainly improve convergence efficiency in this setting.

Interestingly, removing the reactive candidate yields the highest single-run peak accuracy, 91.743%, but reaches 90% accuracy only after 390 iterations, compared with 320 iterations for the full method. Thus, the reactive branch is not primarily responsible for maximizing the peak accuracy of an individual trajectory. Instead, its inclusion is associated with earlier entry into the high-accuracy regime, consistent with its intended role as a prediction-independent fallback in the safeguarded response update. The convergence trajectories in Fig. 3 provide a complementary view of these differences beyond the discrete statistics reported in Table 3.

Network architecture. We next examine the internal architecture of the learned response model while keeping the remaining SUPRO components unchanged.

Table 4: Architecture ablation of the learned response model on BiDoRA under the same 500-iteration setting. All remaining SUPRO components are kept unchanged.

Variant	Peak Acc. (%) ↑	Iter.@Peak	Iter.@90% ↓
Full Model	**91.514**	470	**320**
w/o Coordinate GRU	90.596	470	350
w/o Token GRU	91.284	480	350
w/o Transformer	91.284	460	380
w/o Y/ξ Interaction	91.055	430	370

The full architecture reaches the 90% validation-accuracy threshold in 320 iterations. Removing either the coordinate GRU or the token GRU delays this threshold to 350 iterations. Removing explicit Y/ξ interaction further increases the required number of iterations to 370, while removing the Transformer produces the slowest transition among the architectural variants, reaching 90% at iteration 380.

The effects on peak accuracy provide complementary evidence. Removing the coordinate GRU causes the largest reduction among the architectural variants, from 91.514% to 90.596%, while removing explicit Y/ξ interaction reduces the peak to 91.055%. The token-GRU and Transformer ablations retain relatively high peak accuracies of 91.284%, but require more iterations to enter the high-accuracy regime. These results suggest that coordinate- and block-level temporal modeling, cross-block interaction, and joint modeling of the explicit lower iterate and auxiliary response play complementary roles, with their clearest effect appearing in response-tracking efficiency rather than uniformly large changes in single-run peak accuracy.

Taken together, the ablations indicate that the main empirical advantage of the complete SUPRO configuration lies in accelerating the transition to the high-accuracy regime. Learned response modeling, predictive tracking, response-guided mixing, the safeguarded reactive fallback, and joint Y/ξ modeling contribute to different aspects of this behavior, rather than uniformly improving every single-run metric.

A.3 Implementation Details

This subsection describes the learned response optimizer. These choices instantiate the abstract predictor and proposal heads in Section 3; the safety result itself requires only the certified candidates and acceptance rule in Eqs. (9)– (15).

A.3.1 EXPERIMENTAL SETTINGS

Non-convex numerical case. All experiments use an upper-level dimension of 1, while the lower-level dimension is varied over $d_y \in 10, 100, 500, 1000$, respectively, on an NVIDIA GeForce RTX 4090 GPU. All computations are performed in `float64`. The number of CPU threads is fixed to one. Each method is evaluated with single runs with a fixed random seed. We use e_x as the evaluation metric and regard $e_x \leq 10^{-3}$ as the target accuracy. Metrics are recorded after every outer-level update, including the initial state at iteration 0. Before timing, 10 untimed warm-up updates are performed and the algorithm state is then reset. Reported runtime is the cumulative CUDA-synchronized algorithm execution time, excluding external evaluation and logging overhead.

Representative learning tasks. We evaluate all methods on Omniglot under the 10-way 1-shot and 20-way 1-shot settings following the Vinyals split. Images are resized to 28×28, with four right-angle rotations treated as distinct classes; each task contains 1 support and 15 query examples per class. We use a four-layer 64-channel convolutional backbone with a task-specific linear classifier. The classifier is adapted for 10 support-set steps, while the backbone is optimized on the query loss. Training uses 16 tasks per batch for 5,000 batches with 5 bi-level updates per batch, totaling 25,000 updates. Evaluation is conducted every 20 batches on 1,000 test tasks, reporting mean query accuracy and standard deviation. The reported time is the earliest wall-clock time at which a method reaches 90% mean query accuracy.

Data hyper-cleaning. We evaluate data hyper-cleaning on MNIST and FashionMNIST. For each dataset, we randomly partition the original 60,000-image training pool into 5,000 lower-level training examples, 5,000 clean upper-level validation examples, and 10,000 clean held-out examples; the official test split is not used. We corrupt 50% of the training labels by replacing each selected label uniformly with one of the other nine classes, while keeping the validation and held-out sets clean. Images are converted to `float32`, normalized by the maximum pixel value of each subset, flattened to 784 dimensions, and used without augmentation. A fixed random seed is used to control data partitioning, label corruption, and model initialization. The classifier is $\mathrm{Linear}(784, 300) \to \mathrm{Linear}(300, 10)$ with biases and no intermediate activation. Let $x \in \mathbb{R}^{5000}$ denote the sample-weight logits and y the classifier parameters, with x initialized to zero so that $\sigma(x_i) = 0.5$. The lower and upper objectives are $g(x, y) = \frac{1}{5000} \sum_i \sigma(x_i)\, \mathrm{CE}(h_y(u_i), \tilde{v}_i)$ and $F(y) = \frac{1}{5000} \sum_j \mathrm{CE}(h_y(u_j^{\mathrm{val}}), v_j^{\mathrm{val}})$, respectively, where the weighted lower loss is normalized by the number of samples rather than by the sum of sample weights.

Large model post-training task. For each baseline, learning-rate settings are adopted from the corresponding original paper or official open-source implementation whenever reliable configurations are available. When no stable hyperparameter configuration is provided, we use the task-specific settings together with the parameter recommendations reported in the original paper. All remaining common experimental settings are kept identical whenever applicable and are summarized in Table 5.

Stochastic safeguard implementation. For stochastic learning tasks, all candidate comparisons within the same bi-level iteration are evaluated using the same minibatch and matched stochastic realization. In particular, the reactive, predictor-corrected, and learned candidates are compared under identical data and dropout randomness, which reduces variance in the pairwise objective comparisons. Phase-specific gradients are recomputed at their corresponding candidate points rather than reused from the previous response. We use $\epsilon_t = 0$ in these experiments.

We refer to this procedure as a stochastic or minibatch safeguard. It is an empirical implementation of the deterministic safeguard analyzed in Section 4, rather than an exact certificate with respect to the population lower-level objective. The term "certified" in the theoretical analysis refers to the objective-level condition in Eq. (16).

A.3.2 SCALE-AWARE RESPONSE OPTIMIZER

Joint coordinate and token states. Flatten y_t and ξ_{t-1} using the same parameter layout and partition their coordinates into B blocks without crossing tensor boundaries. For $q \in \{y_t, \xi_{t-1}\}$, each coordinate receives seven detached features: the signed-log normalized lower gradient, the change in its gradient EMA, a bias-corrected momentum ratio, the log second moment, the signed-log parameter value, the RMS upper displacement, and normalized time. A shared linear map

Table 5: Common experimental settings for the SST-2 experiments.

Setting	Value
Task	GLUE SST-2 sentiment classification
Backbone	RoBERTa-base
Backbone architecture	12 Transformer layers, hidden size 768, 12 attention heads
Training data split	80% lower-level training / 20% upper-level training
Lower-level training samples	53,879
Upper-level training samples	13,470
Evaluation set	Official SST-2 validation set (872 samples)
BiDoRA modules	Query and Value projections in all Transformer layers
LoRA rank r	4
LoRA α	8
LoRA scaling α/r	2
Lower-level variables	LoRA A, B matrices and classification head
Upper-level variables	Magnitude parameters of Query and Value weights
Frozen parameters	All remaining pretrained backbone parameters
Lower-level objective	Cross-entropy loss with direction-orthogonality regularization
Regularization coefficient	1×10^{-5}
Upper-level objective	Cross-entropy loss on the upper-level split
Maximum sequence length	512
Batch size	32
LoRA dropout	0
Backbone dropout	0.1
Training budget	1,500 outer iterations
Evaluation interval	Every 10 outer iterations
Random seed	0
Precision	FP32
Optimizer	Adam
Adam (β_1, β_2)	$(0.9, 0.999)$
Adam ϵ	10^{-8}
Weight decay	0
Common warm-up	100 optimizer updates

projects these features, after which separate coordinate GRUs maintain the y- and ξ-states. We additionally form four coordinate-wise relation features from the normalized position and gradient differences between the two states.

Within each block, the mean and RMS of the coordinate states are projected to tokens. Separate token GRUs retain the temporal states $T^{y}_{t,b}$ and $T^{\xi}_{t,b}$. Their interaction is represented by

$$T^{\mathrm{rel}}_{t,b} = \phi_{\mathrm{rel}}\big[T^{y}_{t,b}, T^{\xi}_{t,b}, T^{y}_{t,b} - T^{\xi}_{t,b}, T^{y}_{t,b} \odot T^{\xi}_{t,b}\big]. \tag{30}$$

The B tokens from each branch, the B relation tokens, and one global token—$3B + 1$ tokens in total—are processed by a lightweight Transformer. Its block, relation, and global outputs are fed back to the coordinate states, yielding the joint representation H_t in Eq. (6). Shared parameters do not grow with the number of coordinates, although coordinate-state memory and computation do.

Phase-specific response heads. The prediction, correction, and selection heads share H_t but are queried at different stages. The P head outputs a tanh direction. For each block b, its residual is scaled as

$$\delta p^{(b)}_t = \frac{v^{(b)}_{t,\mathrm{P}}}{\|v^{(b)}_{t,\mathrm{P}}\| + \varepsilon} r_{\mathrm{P}} \sigma(a_{\mathrm{P}}) \|p^{\mathrm{GD},(b)}_t\|, \tag{31}$$

and is accepted only through the direction test in Eq. (7). After evaluating the lower gradient at $\widehat{\xi}_t$, the C head receives this gradient, its change from g_t^0, and a moment-normalized version, and outputs a residual scaled relative to the analytic correction $\xi_t^{\mathrm{P}} - \widehat{\xi}_t$. At ξ_t^{P}, the U head receives lower-gradient and regularized upper-gradient features; its block-normalized direction and positive magnitude form u_t, multiplied by the gate a_t^{U} in Eq. (14). Thus P, C, and U respectively augment prediction, post-prediction correction, and upper-aware response selection. The certificate remains the objective-based rule in Eq. (15), not the learned scaling rules.

A.3.3 Online Training Objectives and Scheduling

Let $\mathrm{sg}(\cdot)$ denote stop-gradient and $[a]_+ = \max\{a, 0\}$. We instantiate the components of Eq. (17) as

$$\mathcal{L}_t^{\mathrm{val}} := f(x_t, \xi_t^{\mathrm{learn}}), \tag{32}$$

$$\mathcal{L}_t^{\mathrm{dom}} := \Big[f(x_t, \xi_t^{\mathrm{learn}}) - \mathrm{sg}\big(f(x_t, \xi_t^{\mathrm{safe}})\big) \Big]_+, \tag{33}$$

$$\mathcal{L}_t^{\mathrm{sel}} := a_t^{\mathrm{U}} \Big[F(x_t, \xi_t^{\mathrm{learn}}) - \mathrm{sg}\big(F(x_t, \xi_t^{\mathrm{safe}})\big) \Big]_+, \tag{34}$$

$$\mathcal{L}_t^{\mathrm{pred}} := \Big\| p_t - \mathrm{sg}\Big(p_t + \xi_t^{\mathrm{P}} - \widehat{\xi}_t + c_t \Big) \Big\|^2, \tag{35}$$

$$\mathcal{L}_t^{\mathrm{stab}} := \chi_t \|p_t + c_t + a_t^{\mathrm{U}} u_t\|^2, \qquad \chi_t := \frac{\tau_s}{\|\Delta \bar{g}_t^{\xi}\| + \|g_t^{\mathrm{L}}\| + \tau_s}, \tag{36}$$

$$\mathcal{L}_t^{\mathrm{gain}} := \tau_g \,\mathrm{softplus}\left(\frac{r_t^{\mathrm{gain}} + m_g}{\tau_g} \right), \quad r_t^{\mathrm{gain}} := \frac{f(x_t, \xi_t^{\mathrm{P}}) - \mathrm{sg}(f(x_t, \xi_t^{\mathrm{R}}))}{|\,\mathrm{sg}(f(x_t, \xi_t^{\mathrm{R}}))| + \varepsilon}. \tag{37}$$

The prediction target in (35) folds the subsequently observed certified and learned lower corrections into the next prediction while excluding u_t. This is a distillation target for the observed correction; it is not an observed exact minimizer displacement. Excluding u_t does not remove upper-level influence from the predictor state or learned correction. Here $\Delta \bar{g}_t^{\xi}$ denotes the change in the response-gradient EMA. The coefficient χ_t increases near small gradient and landscape-shift residuals; the loss penalizes, but does not rule out, unnecessary motion. The gain loss compares the predictor-corrected candidate with the reactive candidate and therefore targets the benefit required by the prediction-adaptive mechanism. In our implementation $m_g = \tau_g = 0.05$ and $\omega_{\mathrm{gain}} = 2$, while the other five loss weights are one. All values and gradients that serve as targets or optimizer inputs are detached before updating ψ_t. Consequently, differentiating the losses above requires vector–Jacobian products through the response optimizer and ordinary first-order gradients of f and F at its output, but never differentiation through a gradient evaluation.

For large models, the response optimizer and the task model operate on separate clocks. We partition ψ into lightweight head parameters and the recurrent/token-mixing backbone, and update the heads more frequently while using the same total loss. Token context may be reused between backbone refreshes, while detached objective values and gradients are cached only within the current iteration. The theoretical results in Section 4 assume fresh deterministic evaluations; stale or stochastic evaluations require the perturbation conditions discussed in Appendix A.5.3.

A.3.4 Parameter-Efficient Model Responses

When the lower response consists of trainable adapter parameters, we flatten only those parameters, not the frozen backbone. For LoRA matrices $A_\ell \in \mathbb{R}^{r_\ell \times d_\ell^{\mathrm{in}}}$ and $B_\ell \in \mathbb{R}^{d_\ell^{\mathrm{out}} \times r_\ell}$, the response is

$$\xi = \mathrm{concat}_{\ell=1}^{L} \big[\mathrm{vec}(A_\ell), \mathrm{vec}(B_\ell) \big]. \tag{38}$$

Tensor boundaries and shapes are retained so that each update can be viewed in the original parameterization without copying the frozen model. Candidate responses can share a frozen backbone and be represented as adapter displacements. This can avoid duplicating frozen weights, but distinct candidates still require objective or gradient evaluations at their own parameter values. Peak memory must include the explicit decision y_t, auxiliary response ξ_t, temporary candidates, gradients, and coordinate-wise optimizer states.

A.3.5 THE STOP-GRADIENT COMPUTATION GRAPH

Let sg be the identity in the forward evaluation with zero derivative in the online optimizer update. At iteration t, hold x_t, y_t, ξ_{t-1} and the incoming recurrent state fixed. Detach all gradient-derived features, gradient-dependent gates, comparison targets, and gradient evaluations in each certified correction. In particular use

$$u^0 = \mathrm{sg}(\xi_{t-1}) + p_t, \qquad u^{k+1} = u^k - \eta_c \,\mathrm{sg}\big(\nabla_z f(x_t, u^k)\big). \tag{39}$$

The numerical forward pass remains ordinary lower-gradient descent, so all certificates and convergence results are unchanged. In the online backward pass $\partial u^{k+1}/\partial \psi_t = \partial u^k/\partial \psi_t$, instead of differentiating a lower gradient. The residual and selection heads receive detached gradient features, and the target in the prediction loss is fully detached. The output losses require first derivatives such as $\nabla_z f(x_t, \xi_t^{\mathrm{learn}})$ and $\nabla_z F(x_t, \xi_t^{\mathrm{learn}})$, composed with the learned map's Jacobian. This defines a first-order surrogate training gradient; it is not the exact meta-gradient through the numerical GD trajectory. Detaching only predictor inputs while differentiating the correction trajectory normally would introduce Hessian-vector products.

At $\nabla_z f = 0$, the regularized projection in (12) reduces to the upper gradient; it does not project onto the tangent space of the minimizer manifold. Even away from zero it is not an exact projection because of its denominator regularization. The upper gate is a proposal heuristic. Only the objective comparison certifies the relative lower-level quality, and the upper comparison concerns $F(x_t, \xi_t)$, not global optimistic response selection or guaranteed improvement of $F(x_t, y_t)$. Moreover, Lemma A.4 makes $\nabla_x f(x, z^\star)$ identical at all exact lower minimizers for a fixed x. Upper-guided auxiliary selection therefore does not change the exact value gradient; any benefit must arise from finite-accuracy tracking or subsequent learning.

A.4 ALGORITHM

Algorithm 1 Safe Upper-Guided Predictive Response Optimization (SUPRO)

Require: Initial variables x_0, y_0, ξ_0; optimizer state (ψ_0, H_0); penalty λ; response steps η_p, η_b, η_c; correction budgets K_b, K_c; mixing cap $\nu_{\max}$; horizon T.

1: Set $x_1 \leftarrow x_0$, $y_1 \leftarrow y_0$, and $\xi_{-1} \leftarrow \xi_0$, $\psi_1 \leftarrow \psi_0$; initialize optimizer statistics.
2: **for** $t = 1, \ldots, T$ **do**
3: Evaluate the lower states at y_t and ξ_{t-1}; encode $H_t \leftarrow \mathcal{E}_{\psi_t}(o_t^y, o_t^\xi, r_t^{y\xi}, H_{t-1})$.
4: Form p_t^{GD} and the bounded residual δp_t; apply Eq. (7) to obtain p_t and $\widehat{\xi}_t$.
5: Construct ξ_t^{R} and ξ_t^{P} using Eq. (9).
6: Select ξ_t^{safe} using Eq. (10).
7: Decode c_t and u_t from H_t with their phase-specific gradient features; set $\xi_t^{\mathrm{learn}} \leftarrow \xi_t^{\mathrm{P}} + c_t + a_t^{\mathrm{U}} u_t$.
8: Accept or reject ξ_t^{learn} according to Eq. (15), yielding ξ_t.
9: Update ψ_{t+1} with Eq. (17) using the detached correction graph in Eq. (39).
10: Form the tracked-penalty gradients at (x_t, y_t, ξ_t); obtain $\bar{y}_{t+1}$ and x_{t+1} from the original task optimizers.
11: Transfer the certified response to y using Eq. (20), yielding y_{t+1}.
12: **end for**

A.5 THEORETICAL PROOFS

A.5.1 PROOFS FOR PREDICTIVE VALUE-RESPONSE TRACKING

All norms are Euclidean. The lower variable is unconstrained. We use the PL convention in (21); no convexity inequality is used below. The assumptions must hold at the candidate initializations, intermediate corrections, and all points needed in the stated smoothness arguments. In particular, a PL inequality imposed only at the accepted iterates is insufficient for these proofs.

Consequences of smoothness and uniform PL geometry

Lemma A.1 (Gap, gradient, and solution-set distance)**.** *Under Assumption 4.1, for $z^\star \in \mathcal{S}(x)$,*

$$\|\nabla_z f(x,z)\|^2 \le 2L_z G(x,z), \qquad G(x,z) \le \tfrac{L_z}{2}\|z - z^\star\|^2, \tag{40}$$

$$\operatorname{dist}(z,\mathcal{S}(x))^2 \le \tfrac{2}{\mu} G(x,z), \qquad \operatorname{dist}(z,\mathcal{S}(x)) \le \tfrac{1}{\mu}\|\nabla_z f(x,z)\|. \tag{41}$$

Proof. For $h(z) = f(x,z)$, the descent lemma at $z^+ = z - L_z^{-1}\nabla h(z)$ gives $h(z^+) \le h(z) - (2L_z)^{-1}\|\nabla h(z)\|^2$. Since $h(z^+) \ge v(x)$, this proves the first inequality. Expanding h at an unconstrained minimizer, where $\nabla h(z^\star) = 0$, proves the second.

For completeness, consider the gradient flow $\dot u(s) = -\nabla h(u(s))$, $u(0) = z$, and $a(s) = h(u(s)) - v(x)$. The globally Lipschitz lower gradient guarantees existence for all $s \ge 0$. Along the flow, $\dot a = -\|\nabla h(u)\|^2 \le -2\mu a$, so $a(s) \le e^{-2\mu s} a(0)$. Whenever $a(s) > 0$,

$$-\frac{d}{ds}\sqrt{a(s)} = \frac{\|\nabla h(u(s))\|^2}{2\sqrt{a(s)}} \ge \sqrt{\frac{\mu}{2}}\|\nabla h(u(s))\|.$$

Integrating shows that the total path length is at most $\sqrt{2a(0)/\mu}$. If the gap reaches zero, the flow remains at its minimizer. Otherwise finite length makes $u(s)$ Cauchy, and continuity gives a limit $u_\infty \in \mathcal{S}(x)$. Hence $\operatorname{dist}(z,\mathcal{S}(x)) \le \|z - u_\infty\| \le \sqrt{2G(x,z)/\mu}$. Combining this bound with PL proves the last inequality. The flow is only a proof device, not an algorithmic step. □

The nearest-point projection onto $\mathcal{S}(x)$ exists because this set is nonempty and closed in finite dimensions. Thus the comparators in Section 4 satisfy $\|\xi_t - z_t^\star\|^2 \le \kappa\Delta_t$ with $\kappa = 2/\mu$. Existence of such a comparator does not imply smooth comparator motion.

Lemma A.2 (Finite-step certified correction)**.** *For $0 < \eta \le 1/L_z$, put $q(\eta) = 1 - 2\mu\eta(1 - L_z\eta/2) \in [0,1)$. Then*

$$G\big(x, \mathrm{GD}_{x,\eta}^K(u)\big) \le q(\eta)^K G(x,u), \qquad K \ge 1. \tag{42}$$

Proof. The descent lemma and PL give $G(x, u - \eta\nabla_z f(x,u)) \le G(x,u) - \eta(1 - L_z\eta/2)\|\nabla_z f(x,u)\|^2 \le q(\eta)G(x,u)$. Iteration proves the result. Since $0 < \mu \le L_z$ and $0 < L_z\eta \le 1$, the specified $q(\eta)$ lies in $[0,1)$. □

Two candidates and the prediction-dependent bound

Lemma A.3 (Candidate recursions)**.** *With the constants in (23),*

$$G(x_t, \xi_t^{\mathrm{R}}) \le \rho_b\Delta_{t-1} + c_b\|d_t^\star\|^2, \tag{43}$$

$$G(x_t, \xi_t^{\mathrm{P}}) \le \rho_c\Delta_{t-1} + c_c\|p_t - d_t^\star\|^2. \tag{44}$$

Proof. The predictive initialization satisfies $\widehat{\xi}_t - z_t^\star = (\xi_{t-1} - z_{t-1}^\star) + (p_t - d_t^\star)$. Young's inequality gives

$$\|\widehat{\xi}_t - z_t^\star\|^2 \le (1+\theta_c)\kappa\Delta_{t-1} + (1+\theta_c^{-1})\|p_t - d_t^\star\|^2.$$

Apply the smooth gap upper bound (40), followed by K_c lower-gradient corrections. The reactive argument uses $\xi_{t-1} - z_t^\star = (\xi_{t-1} - z_{t-1}^\star) - d_t^\star$ and K_b corrections. □

Proof of Theorem 4.2. The acceptance rule compares against the lower-value certified candidate, so, whether the learned proposal is accepted or rejected,

$$\Delta_t \le \min\{G(x_t, \xi_t^{\mathrm{R}}), G(x_t, \xi_t^{\mathrm{P}})\} + \epsilon_t.$$

This is a relative objective certificate, not a test that $G(x_t, \xi_t) = 0$. The candidate recursions imply

$$\Delta_t \le \bar\rho\Delta_{t-1} + b_t^{\min} + \epsilon_t, \qquad b_t^{\min} = \min\{c_b\|d_t^\star\|^2, c_c\|p_t - d_t^\star\|^2\}. \tag{45}$$

Summing and using $\Delta_T \ge 0$ gives

$$(1-\bar\rho)\sum_{t=1}^T \Delta_t \le \bar\rho(\Delta_0 - \Delta_T) + \sum_{t=1}^T (b_t^{\min} + \epsilon_t) \le \bar\rho\Delta_0 + W_T + \mathcal{E}_T.$$

Since a sum of minima is at most either sum, $W_T \le \min\{c_b V_T, c_c E_T\}$. The lower-gradient bound follows from (40). It also bounds the smallest squared lower-gradient residual and the expected squared residual at an iterate sampled uniformly from $\{1, \dots, T\}$. □

For prescribed $\theta_j > 0$ and $q(\eta_j) \in (0,1)$, one sufficient iteration-independent budget is

$$K_j = \max\left\{1, \left\lfloor \frac{\log[(L_z\kappa/2)(1+\theta_j)]}{-\log q(\eta_j)} \right\rfloor + 1\right\}. \tag{46}$$

If $q(\eta_j) = 0$, take $K_j = 1$. These budgets may depend on conditioning; they are constant in T, not uniform over all problem instances. Theorem 4.2 is valid for arbitrary causal predictors and accepted learned proposals obeying the safeguard. Small prediction residual is a sufficient condition for a sharper bound, not a consequence proved for the learned optimizer. Selecting a candidate does not remove the cost of constructing either branch.

A.5.2 Coupled Convergence and the Value-Function Penalty

This subsection first justifies the value gradient, then closes the interaction between response tracking and outer descent. The penalty interpretation is separated from the algorithmic stationarity result, following the proof organization of V-PBGD (Shen & Chen, 2023). All constants below use our PL normalization in (21).

Value regularity without a unique response

Lemma A.4 (Differentiable value and consistent minimizer gradients)**.** *Under Assumption 4.1 and the cross-variable Lipschitz conditions of Assumption 4.3, v is continuously differentiable on $\mathcal{X}$ and*

$$\nabla v(x) = \nabla_x f(x, z^\star) \quad (z^\star \in \mathcal{S}(x)), \qquad L_v \le L_{xx} + \frac{L_{xz}L_{zx}}{\mu}. \tag{47}$$

Proof. Fix any $z^\star \in \mathcal{S}(x)$ and let $x + h \in \mathcal{X}$. Since $\nabla_z f(x, z^\star) = 0$, cross smoothness and PL at $x + h$ give

$$0 \le f(x+h, z^\star) - v(x+h) \le \frac{\|\nabla_z f(x+h, z^\star)\|^2}{2\mu} \le \frac{L_{zx}^2}{2\mu}\|h\|^2. \tag{48}$$

Taylor expansion of $f(\cdot, z^\star)$ at x therefore shows $v(x+h) = v(x) + \langle \nabla_x f(x, z^\star), h\rangle + O(\|h\|^2)$. This proves differentiability and the gradient identity for every minimizer. In particular, all minimizers have the same x-gradient under these uniform neighborhood assumptions.

For $x, x' \in \mathcal{X}$, choose a nearest minimizer $z' \in \mathcal{S}(x')$ to $z^\star \in \mathcal{S}(x)$. By (41), $\|z' - z^\star\| \le \|\nabla_z f(x', z^\star)\|/\mu \le L_{zx}\|x' - x\|/\mu$. Thus

$$\|\nabla v(x') - \nabla v(x)\| \le L_{xx}\|x' - x\| + L_{xz}\|z' - z^\star\| \le (L_{xx} + L_{xz}L_{zx}/\mu)\|x' - x\|.$$

□

If F and f additionally have joint gradient Lipschitz constants L_F and L_f, a sufficient penalty smoothness constant is $L_P = L_F + \lambda(L_f + L_v)$. Boundedness below remains a separate assumption. One sufficient condition is $F \ge F_{\inf}$, since $G \ge 0$. More generally, suppose $\|\nabla_y F\| \le M$ globally in y and the feasible upper objective is bounded below by $F_{\mathrm{feas,inf}}$. A nearest solution and (41) then give

$$P_\lambda(x,y) \ge F_{\mathrm{feas,inf}} - M\sqrt{2G(x,y)/\mu} + \lambda G(x,y) \ge F_{\mathrm{feas,inf}} - \frac{M^2}{2\mu\lambda}.$$

Smoothness and lower-level PL alone do not imply a finite lower bound.

Tracked-gradient error and the prediction-adaptive outer bound

Write $w_t = (x_t, y_t)$ and $g_t = \nabla P_\lambda(w_t) = (g_t^x, g_t^y)$. Let $\widetilde{g}_t^x$ denote the implemented x-direction in Eq. (19).

Lemma A.5 (Inexact descent with response-guided mixing)**.** *With $A_\lambda = 2\lambda^2 L_{xz}^2/\mu$,*

$$\widetilde{g}_t^x = g_t^x + e_t^x, \qquad e_t^x = \lambda[\nabla v(x_t) - \nabla_x f(x_t, \xi_t)], \qquad \|e_t^x\|^2 \le A_\lambda \Delta_t. \tag{49}$$

Suppose $\eta_x = \eta_y = \alpha$ and $\|m_t\| \le \nu_t\alpha\|g_t^y\|$. Define $\phi_t = (1-\nu_t)[1 - L_P\alpha(1-\nu_t)/2]$. Then

$$P_\lambda(w_{t+1}) \le P_\lambda(w_t) - \frac{\alpha}{2}\|g_t^x\|^2 - \alpha\phi_t\|g_t^y\|^2 + \frac{\alpha}{2}\|e_t^x\|^2 - \frac{\alpha}{2}(1 - L_P\alpha)\|\widetilde{g}_t^x\|^2. \tag{50}$$

Proof. Lemma A.4 identifies the exact penalty gradient. Only the x-block is inexact, and its error is bounded by $\lambda L_{xz}\,\mathrm{dist}(\xi_t, \mathcal{S}(x_t))$. Equation (41) proves (49). For the x-block, the standard identity gives the first, third, and fourth terms in (50). For the y-block, write $d_t^y = -\alpha g_t^y + m_t$. Maximizing $\langle g_t^y, d_t^y\rangle + L_P\|d_t^y\|^2/2$ over $\|m_t\| \le \nu_t\alpha\|g_t^y\|$ yields $-\alpha(1-\nu_t)[1 - L_P\alpha(1-\nu_t)/2]\|g_t^y\|^2$. Adding the two blockwise bounds proves the claim. □

Corollary A.6 (Prediction-adaptive penalty stationarity)**.** *Under the assumptions of Theorem 4.2, outer regularity, $0 < \alpha \le 1/L_P$, and $\inf_t \phi_t \ge 1/2$,*

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla P_\lambda(w_t)\|^2 \le \frac{2[P_\lambda(w_1) - P_{\inf}]}{\alpha T} + \frac{A_\lambda \mathcal{B}_T}{T}. \tag{51}$$

Adding $2L_z\mathcal{B}_T/T$ yields the corresponding joint residual bound.

Proof. Use $\phi_t \ge 1/2$, drop the last nonpositive term in (50), sum, use $P_\lambda(w_{T+1}) \ge P_{\inf}$, and substitute $\sum_t \Delta_t \le \mathcal{B}_T$. □

This conditional bound does not by itself control V_T or E_T. The following argument supplies an independent guarantee for the actual coupled algorithm.

Direct gap transport and Lyapunov descent

Lemma A.7 (Transport of the square-root value gap)**.** *For $x, x' \in \mathcal{X}$ and any z,*

$$\left|\sqrt{G(x', z)} - \sqrt{G(x, z)}\right| \le \frac{L_{xz}}{\sqrt{2\mu}}\|x' - x\|. \tag{52}$$

Consequently, for r, B in (26),

$$\Delta_{t+1} \le r\Delta_t + B\|x_{t+1} - x_t\|^2 + \epsilon_{t+1}. \tag{53}$$

Proof. By Lemma A.4, a nearest minimizer gives $\|\nabla_x G(x, z)\| \le L_{xz}\sqrt{2G(x, z)/\mu}$. For $a > 0$, the function $H_a(x) = \sqrt{G(x, z) + a}$ therefore satisfies $\|\nabla H_a(x)\| \le L_{xz}/\sqrt{2\mu}$. Integrate along the segment $[x, x'] \subset \mathcal{X}$ and let $a \downarrow 0$. This proves (52), including points at which $G = 0$.

The reactive branch at $t+1$ starts from ξ_t, so its corrected gap is at most $q_b G(x_{t+1}, \xi_t)$. Apply (52) and Young's inequality to obtain

$$q_b G(x_{t+1}, \xi_t) \le q_b(1 + \vartheta)\Delta_t + \frac{q_b(1 + \vartheta^{-1})L_{xz}^2}{2\mu}\|x_{t+1} - x_t\|^2.$$

The safeguard is at least as good as this reactive bound up to ϵ_{t+1}. No control on tangential motion inside $\mathcal{S}(x_t)$ is used. □

Proof of Theorem 4.4. Put $\Psi_t = P_\lambda(w_t) + \omega\Delta_t$. Because $\|x_{t+1} - x_t\| = \alpha\|\widetilde{g}_t^x\|$ and $\phi_t \ge \phi_{\min} \ge 1/2$, Lemmas A.5 and A.7 give

$$\begin{aligned}\Psi_{t+1} - \Psi_t \le -\frac{\alpha}{2}\|g_t\|^2 + \left[\frac{\alpha A_\lambda}{2} - \omega(1 - r)\right]\Delta_t \\ + \left[-\frac{\alpha}{2}(1 - L_P\alpha) + \omega B\alpha^2\right]\|\widetilde{g}_t^x\|^2 + \omega\epsilon_{t+1}.\end{aligned} \tag{54}$$

Since $\omega(1-r) = \alpha C_\lambda$ and $C_\lambda = A_\lambda + 2L_z$, the coefficient of Δ_t is at most $-\alpha C_\lambda/2$. The stepsize conditions ensure $1 - L_P\alpha \ge 1/2$ and $\omega B\alpha^2 = \alpha^3 C_\lambda B/(1-r) \le \alpha/4$. Thus

$$\Psi_{t+1} \le \Psi_t - \frac{\alpha}{2}\big(\|g_t\|^2 + C_\lambda\Delta_t\big) + \omega\epsilon_{t+1}. \tag{55}$$

For a finite run ending with the outer update w_{T+1}, define only for this proof a terminal response $\xi_{T+1} = \mathrm{GD}_{x_{T+1},\eta_b}^{K_b}(\xi_T)$ and set $\epsilon_{T+1} = 0$. This is a virtual point for telescoping; the algorithm need not compute it. Its nonnegative gap and the lower bound on P_λ give $\Psi_{T+1} \ge P_{\inf}$. Summing (55) now yields

$$\sum_{t=1}^{T}\big(\|g_t\|^2 + C_\lambda\Delta_t\big) \le \frac{2(\Psi_1 - P_{\inf})}{\alpha} + \frac{2\omega}{\alpha}\sum_{t=2}^{T}\epsilon_t. \tag{56}$$

Finally, $\|\nabla_z f(x_t, \xi_t)\|^2 \le 2L_z\Delta_t \le C_\lambda\Delta_t$, and $\sum_{t=2}^{T}\epsilon_t \le \mathcal{E}_T$. This proves (28). □

For $0 < q_b < 1$, the explicit choice $\vartheta = (1 - q_b)/(2q_b)$ gives $r = (1 + q_b)/2 < 1$; if $q_b = 0$, any $\vartheta > 0$ works. Thus even $K_b = 1$ suffices for this coupled guarantee after choosing the outer step. It need not suffice for the distinct $\bar{\rho} < 1$ condition in the comparator-based bound. If $\sum_t \epsilon_t < \infty$, an infinite run has $\sum_t (\|g_t\|^2 + \Delta_t) < \infty$, hence both residuals tend to zero. This does not by itself establish convergence of the iterates. At fixed λ and fixed problem constants, a uniformly sampled iterate has expected joint squared residual $O(T^{-1})$; achieving an expected squared residual at most δ^2 takes $O(\delta^{-2})$ outer iterations. Dependence on λ, initialization, smoothness, and conditioning remains explicit in (56). In particular, $A_\lambda = O(\lambda^2)$ and L_P may grow with λ, so this is not a penalty-uniform complexity claim.

Unequal upper and explicit lower steps

The updates in Section 3.4 also allow distinct $\eta_x, \eta_y > 0$. The x-block obeys the inexact-gradient identity with coefficient η_x, while normalized mixing gives the y-block coefficient

$$\phi_y = (1 - \nu_{\max}) \left[1 - \frac{L_P \eta_y}{2} (1 - \nu_{\max}) \right].$$

Assume $\phi_y \geq 1/2$, let $d_{\min} = \min\{\eta_x, \eta_y\}$, choose $\omega_D = \eta_x C_\lambda / (1 - r)$, and require

$$\eta_x \leq \frac{1}{2L_P}, \qquad \eta_x^2 \leq \frac{1 - r}{4C_\lambda B}.$$

Because response-gap transport depends only on $x^+ - x = -\eta_x \widetilde{g}^x$, the same Lyapunov argument gives

$$\Psi_{t+1}^D \leq \Psi_t^D - \frac{d_{\min}}{2} \left(\|g_t\|^2 + C_\lambda \Delta_t \right) + \omega_D \epsilon_{t+1}, \qquad \Psi_t^D = P_\lambda(w_t) + \omega_D \Delta_t.$$

Thus (28) holds with α replaced by $d_{\min}$ and ω by ω_D. Arbitrarily unbalanced steps can worsen the bound through $\eta_x / d_{\min}$.

What penalty stationarity implies for the explicit response

Proposition A.8 (Approximate lower feasibility)**.** *If $\|\nabla_y F(x, y)\| \leq M$ and $\|\nabla P_\lambda(x, y)\| \leq \delta$, then*

$$G(x, y) \leq \frac{(M + \delta)^2}{2\mu\lambda^2}, \qquad \operatorname{dist}(y, \mathcal{S}(x)) \leq \frac{M + \delta}{\mu\lambda}. \tag{57}$$

If the gradient bound holds at all iterates, then

$$\frac{1}{T} \sum_{t=1}^{T} G(x_t, y_t) \leq \frac{M^2 + T^{-1} \sum_{t=1}^{T} \|g_t\|^2}{\mu\lambda^2}. \tag{58}$$

Proof. The exact y-block is $\nabla_y P_\lambda = \nabla_y F + \lambda \nabla_y f$, so $\lambda \|\nabla_y f\| \leq M + \delta$. Use PL and (41). For the average, use $(M + \|g_t\|)^2 \leq 2M^2 + 2\|g_t\|^2$ and sum. □

The guarantee for ξ_t concerns an auxiliary value response; (57) is the separate statement for the explicit decision y_t. In particular, fixed λ leaves a possible $O(\lambda^{-2})$ value-gap floor even at an exact penalty stationary point. At a random iterate, the averaged squared-residual theorem must be used in expectation, not as an almost-sure bound on its norm.

For a connection with penalty-based bi-level formulations, define $\mathcal{C}_\varepsilon = \{(x, y) : G(x, y) \leq \varepsilon\}$. If w_λ is an exact local (respectively global) minimizer of P_λ, it is a local (respectively global) minimizer of F over $\mathcal{C}_{G(w_\lambda)}$. Indeed, any nearby (respectively any) w with $G(w) \leq G(w_\lambda)$ and $F(w) < F(w_\lambda)$ would also have $P_\lambda(w) < P_\lambda(w_\lambda)$. This is an attained-gap relaxed problem, and the minimizer premise is stronger than stationarity. Analogously, an ε_o-global penalty minimizer is only an ε_o-global minimizer of that relaxation. We do not claim exact penalty equivalence at finite λ or a hypergradient stationarity guarantee for the original optimistic bi-level objective.

Discussion of assumptions and guarantees

The outer descent stepsize alone does not close tracking. Let $F = 0$, $f(x, z) = (z - x)^2/2$, $\lambda = 1$, and $\alpha = 1/2 = 1/L_P$ with $L_P = 2$. Set $\eta_b = \eta_c = 0.1$, $K_b = K_c = 1$, $p_t = 0$, and $\epsilon_t = 0$; there is no learned proposal. Then $L_z = \mu = 1$, $\kappa = 2$, and $\theta_b = \theta_c = 0.1$ give $\bar{\rho} = 0.891 < 1$. Nevertheless, with $a_t = y_t - x_t$ and $b_t = \xi_t - x_t$,

$$\begin{pmatrix} a_{t+1} \\ b_{t+1} \end{pmatrix} = \begin{pmatrix} 0 & 0.5 \\ -0.45 & 1.35 \end{pmatrix} \begin{pmatrix} a_t \\ b_t \end{pmatrix}.$$

The largest eigenvalue is $(27 + \sqrt{369})/40 > 1$. Initialization with a nonzero component in its eigendirection makes the tracking and penalty residuals grow. The prediction-dependent bound remains valid because $V_T = E_T$ grows as well. This example motivates the coupled stepsize condition in Theorem 4.4.

Fixed-penalty minimizers need not be original bi-level stationary. Take $f(x, y) = (y - x)^2/2$, $F(x, y) = x^2/2 - x + \sin(y + 1)$, and $\lambda = 1$. At $(x, y) = (0, -1)$ the penalty gradient vanishes and its Hessian is $\left(\begin{smallmatrix} 2 & -1 \\ -1 & 1 \end{smallmatrix}\right) \succ 0$. This is a strict local penalty minimizer, but the original reduced objective is $H(x) = x^2/2 - x + \sin(x + 1)$ and $H'(0) = \cos(1) - 1 \neq 0$. Its lower gap is $1/2$. Thus even a strict local minimum of a smooth fixed penalty need not yield stationarity of the original reduced objective.

Nonunique responses can move without value error. For $f(x, (z_1, z_2)) = (z_1 - x)^2/2$, the solution set is $\{(x, s) : s \in \mathbb{R}\}$. With $F = 0$ an accepted exact response may move arbitrarily in s while $\Delta_t = 0$. No estimate of comparator motion by $\|x_t - x_{t-1}\|$ follows from the value safeguard alone. The direct gap transport proof avoids this unsupported branch-continuity inference.

A.5.3 Inexact Evaluations and Perturbation Bounds

The main theorems concern fresh deterministic evaluations. The following bounds specify what is needed to extend them; caching or minibatch evaluation alone does not supply these bounds.

Inexact certified gradients. Suppose a correction step is $u^+ = u - \eta[\nabla_z f(x, u) + a]$, $0 < \eta \leq 1/L_z$. The identity used in (50), followed by PL, gives

$$G(x, u^+) \leq (1 - \mu\eta)G(x, u) + \tfrac{\eta}{2}\|a\|^2.$$

Therefore a K-step candidate satisfies

$$G(x, u^K) \leq \widetilde{q}^K G(x, u^0) + N, \qquad N = \frac{\eta}{2} \sum_{k=0}^{K-1} \widetilde{q}^{K-1-k} \|a_k\|^2, \quad \widetilde{q} = 1 - \mu\eta. \tag{59}$$

For prediction-adaptive tracking, replace q_j by $\widetilde{q}_j^{K_j}$ in the candidate constants and add $\max\{N_{b,t}, N_{c,t}\}$ to the per-iteration perturbation. For the coupled theorem, likewise replace q_b by $(1 - \mu\eta_b)^{K_b}$ when defining r and B, and add only the reactive error $N_{b,t+1}$ to (53).

Inexact objective certification. Suppose all three candidate evaluations share a bound $|\widehat{f}_t(\zeta) - f(x_t, \zeta)| \leq d_t$, and both safe selection and learned acceptance compare these same evaluations. If the learned proposal passes, its estimated value is at most the minimum estimated certified value plus ϵ_t; the rejected case satisfies the same inequality. Thus

$$f(x_t, \xi_t) \leq \min\{f(x_t, \xi_t^{\mathrm{R}}), f(x_t, \xi_t^{\mathrm{P}})\} + \epsilon_t + 2d_t.$$

This requires simultaneous candidate-wise error control, which is stronger than unbiasedness of a minibatch estimate.

Additional outer-gradient error. If the implemented outer direction is $g_t + e_t + \zeta_t$, then $\|e_t + \zeta_t\|^2 \leq 2A_\lambda\Delta_t + 2\|\zeta_t\|^2$. More generally, suppose the full error obeys $\|e_t^{\mathrm{all}}\|^2 \leq A'\Delta_t + \nu_t$ and direct gap transport has additive error b_{t+1}. Use $C' = A' + 2L_z$, $\omega' = \alpha C'/(1 - r)$, and the same

stepsize conditions with C' in place of C_λ, and define $\Psi'_1 = P_\lambda(w_1) + \omega'\Delta_1$. The identical proof yields

$$\frac{1}{T}\sum_{t=1}^{T}\big(\|g_t\|^2 + 2L_z\Delta_t\big) \le \frac{2(\Psi'_1 - P_{\inf})}{\alpha T} + \frac{1}{T}\sum_{t=1}^{T}\nu_t + \frac{2\omega'}{\alpha T}\sum_{t=2}^{T+1} b_t. \tag{60}$$

The same virtual exact terminal correction may be used to set $b_{T+1} = 0$. Summable errors recover inverse-linear averaged squared residuals; nonvanishing average error leaves a residual-floor term in the bound. Under bounded delays and Lipschitz gradients, stale-gradient errors can be bounded by the corresponding intervening iterate displacements, but that bound and its absorption must be established for the actual refresh schedule. The fresh theorem cannot simply be applied to arbitrary cached or stochastic updates.

A.5.4 Local-PL Extensions of Theorems 4.2 and 4.4

The global lower-level PL condition in Assumption 4.1 is convenient for stating the main results, but is stronger than what is actually used by the analysis. The proofs only require PL geometry on the region visited by the accepted responses, the predictive and reactive correction trajectories, and, for the coupled outer analysis, the response tube swept along consecutive upper iterates. We formalize this observation below.

Admissible response tubes. For each x, let $\mathcal{B}(x) \subseteq \mathbb{R}^{d_y}$ be a closed response region and define

$$S_{\mathcal{B}}(x) := \mathcal{S}(x) \cap \mathcal{B}(x).$$

We call

$$\mathcal{T} := \{(x, z) : x \in \mathcal{X}_{\mathrm{loc}},\ z \in \mathcal{B}(x)\}$$

an admissible response tube if the following conditions hold.

Assumption A.9 (Local lower-level PL geometry)**.** There exist constants $L_z > 0$ and $\mu > 0$ such that:

1. For every $x \in \mathcal{X}_{\mathrm{loc}}$, $S_{\mathcal{B}}(x)$ is nonempty.
2. For every $(x, z) \in \mathcal{T}$, $f(x, \cdot)$ is L_z-smooth on $\mathcal{B}(x)$ and satisfies the local PL inequality
$$\frac{1}{2}\|\nabla_z f(x, z)\|^2 \ge \mu G(x, z), \qquad G(x, z) := f(x, z) - v(x). \tag{61}$$
3. The lower-gradient flow initialized at every response point used in the analysis remains in $\mathcal{B}(x)$ and converges to a point in $S_{\mathcal{B}}(x)$.
4. At iteration t, the accepted response ξ_t, the predictive initialization $\widehat{\xi}_t$, the reactive initialization ξ_{t-1}, and every intermediate lower-gradient iterate used to construct ξ_t^{R} and ξ_t^{P} belong to $\mathcal{B}(x_t)$.

Assumption A.9 replaces global PL geometry by PL geometry only on an invariant response tube. In particular, no PL condition is required outside $\mathcal{T}$.

Local gap–distance and correction bounds. The arguments of Lemmas A.1–A.2 localize directly.

Lemma A.10 (Local gap, distance, and correction)**.** *Under Assumption A.9, for every response point $z \in \mathcal{B}(x)$ appearing in the algorithm or proof,*

$$\|\nabla_z f(x, z)\|^2 \le 2L_z G(x, z), \tag{62}$$

and

$$\operatorname{dist}\big(z, S_{\mathcal{B}}(x)\big)^2 \le \frac{2}{\mu} G(x, z). \tag{63}$$

Moreover, if the K gradient-correction iterates initialized at u remain inside $\mathcal{B}(x)$ and $0 < \eta \le 1/L_z$, then

$$G\Big(x, \mathsf{GD}_{x,\eta}^{K}(u)\Big) \le q(\eta)^K G(x, u), \tag{64}$$

where

$$q(\eta) = 1 - 2\mu\eta\left(1 - \frac{L_z\eta}{2}\right) \in [0, 1). \tag{65}$$

Proof. The smooth-gap bound follows by the same descent-lemma argument as in Lemma A.1, since the descent step remains in the local smoothness region. For the distance bound, apply the same gradient-flow argument as in Lemma A.1. By Assumption A.9, the entire flow remains inside $\mathcal{B}(x)$, where the PL inequality (61) is valid, and converges to a point in $S_{\mathcal{B}}(x)$. This gives

$$\operatorname{dist}\big(z, S_{\mathcal{B}}(x)\big) \le \sqrt{\frac{2G(x,z)}{\mu}}.$$

Finally, the standard descent lemma followed by the local PL inequality (61) yields the one-step contraction whenever the correction step remains in $\mathcal{B}(x)$; iteration gives (64). □

Local-PL prediction-adaptive tracking. For each t, choose

$$z_t^\star \in \operatorname{proj}_{S_{\mathcal{B}}(x_t)}(\xi_t), \qquad d_t^\star := z_t^\star - z_{t-1}^\star, \tag{66}$$

and define V_T and E_T as in (22) using these local comparators.

Theorem A.11 (Local-PL version of Theorem 4.2)**.** *Suppose Assumption A.9 and the certified candidate-selection condition (16) hold. Define q_j, ρ_j, and c_j, $j \in \{b, c\}$, exactly as in (23), with the local PL constant μ, and suppose*

$$\bar\rho := \max\{\rho_b, \rho_c\} < 1.$$

Then

$$\sum_{t=1}^{T} \Delta_t \le \frac{\bar\rho\Delta_0 + \sum_{t=1}^{T} \min\left\{c_b\|d_t^\star\|^2, c_c\|p_t - d_t^\star\|^2\right\} + \mathcal{E}_T}{1-\bar\rho}, \tag{67}$$

and consequently

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla_z f(x_t, \xi_t)\|^2 \le \frac{2L_z B_T^{\mathrm{loc}}}{T}, \tag{68}$$

where

$$B_T^{\mathrm{loc}} := \frac{\bar\rho\Delta_0 + \min\{c_b V_T, c_c E_T\} + \mathcal{E}_T}{1-\bar\rho}.$$

Proof. Combining Lemma A.10 with the same Young's-inequality argument used in Lemma A.3 gives the two candidate recursions

$$G(x_t, \xi_t^{\mathrm{R}}) \le \rho_b\Delta_{t-1} + c_b\|d_t^\star\|^2,$$

and

$$G(x_t, \xi_t^{\mathrm{P}}) \le \rho_c\Delta_{t-1} + c_c\|p_t - d_t^\star\|^2.$$

The certified-selection argument based on (16) and the telescoping step are therefore identical to the proof of Theorem 4.2. □

Local outer-tube compatibility. The coupled result requires the local tube to remain valid not only at the accepted upper iterates, but also along the upper-variable segments used in the value-transport argument.

Assumption A.12 (Outer-compatible local tube)**.** Assumption 4.3 holds on an open neighborhood containing

$$\bigcup_t [x_t, x_{t+1}]$$

and the corresponding response tube. In addition, for every x, x' lying on the same segment $[x_t, x_{t+1}]$ and every $z^\star \in S_{\mathcal{B}}(x)$ used in the local value-function argument, the point $(x', z^\star)$ remains inside the local tube on which Assumption A.9 holds.

Under Assumption A.12, the proof of Lemma A.4 localizes: for $x + h$ sufficiently close to x and $z^\star \in S_{\mathcal{B}}(x)$,

$$0 \le f(x+h, z^\star) - v(x+h) \le \frac{\|\nabla_z f(x+h, z^\star)\|^2}{2\mu} \le \frac{L_{zx}^2}{2\mu}\|h\|^2. \tag{69}$$

Hence the same value-gradient identity holds locally,

$$\nabla v(x) = \nabla_x f(x, z^\star), \qquad z^\star \in S_{\mathcal{B}}(x), \tag{70}$$

and the same local smoothness constant

$$L_v \le L_{xx} + \frac{L_{xz}L_{zx}}{\mu} \tag{71}$$

is valid on the realized outer trajectory.

The square-root gap transport bound also localizes:

$$\left|\sqrt{G(x', z)} - \sqrt{G(x, z)}\right| \le \frac{L_{xz}}{\sqrt{2\mu}}\|x' - x\|, \tag{72}$$

provided the segment joining x and x' remains inside the admissible outer tube.

Theorem A.13 (Local-PL version of Theorem 4.4)**.** *Suppose Assumptions A.9 and A.12, the safeguard condition (16), and the stepsize and mixing conditions (26)– (27) hold. Let*

$$\Psi_1 = P_\lambda(x_1, y_1) + \omega\Delta_1.$$

Then

$$\frac{1}{T}\sum_{t=1}^{T}\left[\|\nabla P_\lambda(x_t, y_t)\|^2 + \|\nabla_z f(x_t, \xi_t)\|^2\right] \le \frac{2(\Psi_1 - P_{\inf}) + 2\omega\mathcal{E}_T}{\alpha T}. \tag{73}$$

Therefore, summable safeguard tolerances again yield an $O(T^{-1})$ averaged squared-residual rate.

Proof. The proof of Theorem 4.4 uses only: (i) the gap-to-solution-set bound, (ii) finite-step contraction of the reactive branch, (iii) local value-gradient regularity, (iv) square-root gap transport along $[x_t, x_{t+1}]$, and (v) the deterministic relative bound on the mixing displacement. Properties (i) and (ii) follow from Lemma A.10; properties (iii) and (iv) follow from Assumption A.12 and (70)– (72); and property (v) is unchanged from the mixing construction in (20). Thus the Lyapunov argument in the proof of Theorem 4.4 applies verbatim. □

Periodic sine example. The local formulation covers the periodic nonconvex problem used in Section 5, although that problem does not satisfy the global PL condition in Assumption 4.1.

Consider

$$f(x, y) = \sum_{i=1}^{d_y} \sin(x + y_i - c_i), \tag{74}$$

whose global value is $v(x) = -d_y$. For $k \in \mathbb{Z}^{d_y}$ and any $r \in (0, \pi)$, define the basin

$$\mathcal{B}_k(x; r) := \left\{y : \left|x + y_i - c_i + \frac{\pi}{2} - 2\pi k_i\right| \le r,\ i = 1, \dots, d_y\right\}. \tag{75}$$

Let

$$\delta_i = x + y_i - c_i + \frac{\pi}{2} - 2\pi k_i.$$

Then, on $\mathcal{B}_k(x; r)$,

$$G(x, y) = \sum_{i=1}^{d_y}(1 - \cos\delta_i), \tag{76}$$

whereas

$$\frac{1}{2}\|\nabla_y f(x, y)\|^2 = \frac{1}{2}\sum_{i=1}^{d_y}\sin^2\delta_i = \sum_{i=1}^{d_y}\frac{1 + \cos\delta_i}{2}(1 - \cos\delta_i). \tag{77}$$

Since $|\delta_i| \le r$,

$$\frac{1}{2}\|\nabla_y f(x,y)\|^2 \ge \mu_r G(x,y), \qquad \mu_r := \frac{1+\cos r}{2} > 0. \tag{78}$$

Thus the periodic sine objective satisfies a uniform PL inequality on each basin $\mathcal{B}_k(x;r)$ even though it does not satisfy global PL on $\mathbb{R}^{d_y}$.

Moreover, for fixed x and $0 < \eta \le 1$, a lower-gradient step obeys

$$\delta_i^+ = \delta_i - \eta \sin \delta_i. \tag{79}$$

For $|\delta_i| < \pi$, the update preserves the sign of δ_i and satisfies

$$|\delta_i^+| \le |\delta_i|, \tag{80}$$

so every basin $\mathcal{B}_k(x;r)$ is invariant under the certified lower-gradient correction.

Consequently, Theorems A.11 and A.13 apply to the sine example whenever the predictive initializations, accepted learned proposals, and the upper-variable transport remain inside an admissible basin tube. This can be guaranteed, for example, by a trust-region or clipping safeguard on the predictive displacement, or verified directly along the realized trajectory.

B Ethics Statement

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets u were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

C Reproducibility Statement

We have made every effort to ensure that the results presented in this paper are reproducible. All code and datasets have been made publicly available in an anonymous repository (`https://anonymous.4open.science/r/Predictive-Tracking-Bilevel-4F13`) to facilitate replication and verification. The experimental setup, including training steps, model configurations, and hardware details, is described in detail in the paper.

D AI Use Statement

We used generative AI tools to assist with theoretical analysis, including examining proof steps and checking the consistency of assumptions and stated bounds, and to improve the clarity and language of the manuscript. The authors reviewed and verified the final mathematical arguments and AI-assisted text and take full responsibility for the content of this paper.