WHEN SINGLE-LOOP METHODS FAIL: RANDOMIZED CORRECTION FOR CONTEXTUAL STOCHASTIC BILEVEL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Directly extending single-loop stochastic bilevel methods to contextual stochastic bilevel optimization (CSBO) may optimize a different objective. We characterize the averaged surrogate induced by shared-state transfers and give a smooth counterexample where surrogate stationarity vanishes while the CSBO stationarity error remains nonzero. We then propose randomly corrected single-loop gradient averaging (RCSL-GA), which retains cheap shared-state updates and invokes contextual refinement with probability p. Inverse-probability weighting corrects the shared-state bias up to finite-solve error, with p controlling a computation–variance trade-off. For smooth CSBO with strongly convex unconstrained lower-level problems, the raw-update variant reaches $\mathbb{E}\|\nabla F(\widehat{x})\|^2 \le \epsilon^2$ in $\mathcal{O}(\epsilon^{-4})$ outer iterations for fixed $p > 0$ with accuracy-dependent refinement. Its first- and second-order oracle complexities are $\widetilde{\mathcal{O}}(\epsilon^{-6})$ and $\widetilde{\mathcal{O}}(\epsilon^{-4})$; the total bound improves to $\widetilde{\mathcal{O}}(\epsilon^{-4})$ when conditional lower-gradient noise vanishes. We also identify when infrequent correction lowers expected per-iteration cost relative to double-loop methods. Experiments on a structural-bias example, Tiny-ImageNet meta-learning, and a task-aggregated power-scheduling surrogate show favorable solution quality and wall-clock gains, with infrequent correction on Tiny-ImageNet nearly matching correction at every iteration at substantially lower cost.

1 INTRODUCTION

Contextual stochastic bilevel optimization (CSBO), introduced by Hu et al. (2023), models learning and decision problems in which a shared upper-level decision must account for lower-level responses that vary with an observed context. It takes the form

$$\begin{aligned} \min_{x\in\mathbb{R}^{d_x}} \quad & F(x) := \mathbb{E}_{\substack{\xi\sim\mathbb{P}_\xi \\ \eta\sim\mathbb{P}_{\eta|\xi}}} \left[f\big(x, y^\star(x;\xi);\eta,\xi\big)\right], \\ \text{where} \quad & y^\star(x;\xi) := \operatorname*{argmin}_{y\in\mathbb{R}^{d_y}} \mathbb{E}_{\eta\sim\mathbb{P}_{\eta|\xi}} \left[g(x,y;\eta,\xi)\right], \quad \forall x,\xi. \end{aligned} \tag{1}$$

Here x is shared across contexts, whereas the lower response $y^\star(x;\xi)$ adapts to the context ξ and its conditional data η. This additional dependence gives CSBO substantially greater modeling flexibility than ordinary stochastic bilevel optimization (SBO): it covers task adaptation, personalized learning, and decision-focused learning, where predictions are evaluated through their induced decisions (Donti et al., 2017; Ji et al., 2021; Hu et al., 2023). We allow the upper objective to be nonconvex and assume that each contextual lower problem is strongly convex. The sampling oracle draws contexts from $\mathbb{P}_\xi$ and, for a fixed context, observations from $\mathbb{P}_{\eta|\xi}$.

The modeling power of CSBO comes with a substantial computational burden. Each sampled context defines a different lower problem, so constructing an accurate hypergradient generally requires approximating a new $y^\star(x;\xi)$ and its sensitivity to x. Existing general-purpose CSBO methods therefore build a context-specific estimator at every upper iteration, using either repeated lower-level optimization and second-order products (Hu et al., 2023) or penalty and multilevel constructions based on first-order information (Yang et al., 2025). These estimators attain provable guarantees,

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.

but the repeated contextual solves make individual iterations expensive. Reduction to ordinary SBO can avoid this procedure under additional structure on the response map and context distribution (Bouscary et al., 2026), but does not cover general CSBO instances.

Single-loop methods offer a compelling alternative in ordinary SBO. They maintain persistent lower and auxiliary states and update them together with the upper variable, avoiding a high-accuracy lower solve at every iteration (Hong et al., 2023; Khanduri et al., 2021; Chen et al., 2021; 2022; Dagréou et al., 2022; Chen et al., 2024). Their low per-iteration cost and warm-started recursions are especially attractive for large stochastic problems. The same mechanism, however, is not immediately available in CSBO. While an SBO state tracks a single response map $y^\star(x)$, independently resampled contexts present a sequence of different targets $y^\star(x;\xi)$; one persistent state cannot generally track all of them. This leads to the central question of this work:

Can we build a fully single-loop algorithm for CSBO problems, with convergence guarantees?

Here for a broad class of shared-state direct transfers, the answer is negative. We show that they target an averaged surrogate bilevel problem rather than the original CSBO objective, and construct a smooth example in which surrogate stationarity coexists with a nonzero CSBO stationarity error.

To address this structural bias, we propose randomized corrected single-loop gradient-averaging (RCSL-GA) methods. RCSL-GA retains a cheap shared-state direction at every iteration and, with probability p, adds an inverse-probability-weighted context-specific correction. This preserves an inexpensive single-loop backbone and pays for contextual refinement only at randomly selected iterations. Our main contributions are summarized as follows.

- We characterize the population target of a class of shared-state direct transfers of SBO single-loop methods and show that they optimize an averaged surrogate rather than the original CSBO objective. A smooth nonconvex–strongly-convex counterexample shows that surrogate stationarity can coexist with a nonzero CSBO stationarity error.
- We introduce the RCSL framework and its gradient-averaging instantiation, RCSL-GA. An inverse-probability-weighted contextual correction, activated with probability p, removes the shared-state bias up to finite-solve error. We characterize the remaining bias and the resulting computation–variance trade-off.
- RCSL-GA reaches ϵ-stationarity in $\mathcal{O}((1+p^{-1})\epsilon^{-4})$ outer iterations under our regularity assumptions. For fixed $p > 0$, its expected first- and second-order primitive-call complexities are $\widetilde{\mathcal{O}}(\epsilon^{-6})$ and $\widetilde{\mathcal{O}}(\epsilon^{-4})$, respectively. We further derive conditions under which infrequent correction reduces expected per-iteration cost relative to the compared double-loop methods.
- Experiments on a structural-bias example, Tiny-ImageNet meta-learning, and a task-aggregated power-scheduling surrogate demonstrate favorable solution quality and wall-clock efficiency, with small correction probabilities often sufficient in practice.

2 Related Work

Stochastic bilevel optimization. For strongly convex lower problems, implicit-differentiation methods for stochastic bilevel optimization (SBO) approximate the lower solution and an inverse-Hessian–vector product (Ghadimi & Wang, 2018; Ji et al., 2021). Alternating and single-loop methods such as TTSA, SUSTAIN, ALSET, and STABLE track the lower solution as the upper iterate evolves (Hong et al., 2023; Khanduri et al., 2021; Chen et al., 2021; 2022), while SOBA, SABA, and MA-SOBA additionally track an auxiliary linear-system variable (Dagréou et al., 2022; Chen et al., 2024). Fully first-order methods instead rely on penalty or value-function formulations (Kwon et al., 2023; 2024b;a). These tracking mechanisms exploit a single response map $y^\star(x)$, which is absent when independently sampled contexts induce different lower responses.

Contextual stochastic bilevel optimization. For finite-task SBO, Guo et al. (2021) develop randomized variance-reduced methods that update only a constant number of tasks per iteration, although their complexity retains task-count dependence. Hu et al. (2023) introduce CSBO with potentially continuous contexts; their DL-SGD and RT-MLMC methods attain $\widetilde{\mathcal{O}}(\epsilon^{-6})$ and $\widetilde{\mathcal{O}}(\epsilon^{-4})$

sample complexities, respectively, using second-order products. Yang et al. (2025) avoid second-order oracles through an optimality-gap penalty, obtaining $\widetilde{\mathcal{O}}(\epsilon^{-8})$ for their basic method and $\widetilde{\mathcal{O}}(\epsilon^{-6})$ with RT-MLMC. Bouscary et al. (2026) instead approximate the response map with basis functions and reduce CSBO to SBO using joint samples, attaining $\widetilde{\mathcal{O}}(\epsilon^{-3})$ under additional approximation, regularity, and conditioning assumptions. Here ϵ-stationarity means $\mathbb{E}\|\nabla F(\widehat{x})\|^2 \le \epsilon^2$; these rates are not directly comparable in computational cost because the methods use different oracle models and assumptions.

Relation to our work. We study shared-state direct transfers of single-loop SBO methods to general CSBO, rather than finite-task update selection (Guo et al., 2021) or structured response-map approximation (Bouscary et al., 2026). Building on the tracking difficulty noted by Hu et al. (2023), we characterize the averaged surrogate induced by such transfers and show that surrogate stationarity need not imply CSBO stationarity. RCSL-GA addresses this mismatch by combining shared lower and auxiliary states with context-specific corrections sampled with probability p. Unlike the RT-MLMC methods of Hu et al. (2023); Yang et al. (2025), RCSL-GA estimates the contextual–shared direction difference only intermittently; inverse-probability weighting corrects the resulting mean bias up to finite contextual-solve error.

Integrated estimation–optimization and decision-focused learning. Integrated estimation–optimization (IEO) and decision-focused learning train predictive models using downstream decision quality, including predict-then-optimize pipelines (Donti et al., 2017; Elmachtoub & Grigas, 2022), differentiable optimization layers (Amos & Kolter, 2017; Agrawal et al., 2019), and conditional estimation–optimization formulations (Qi et al., 2026; Elmachtoub et al., 2023); see Mandi et al. (2024); Sadana et al. (2025) for surveys. Such problems can often be cast as CSBO by treating tasks or features as contexts and downstream decisions as lower responses.

3 From Single-Loop SBO to CSBO: Surrogate Targets and Structural Bias

For each context ξ, define

$$f_\xi(x,y) := \mathbb{E}_{\eta|\xi}[f(x,y;\eta,\xi)], \qquad G_\xi(x,y) := \mathbb{E}_{\eta|\xi}[g(x,y;\eta,\xi)]. \tag{2}$$

Then problem Equation (1) can be written as

$$F(x) := \mathbb{E}_\xi\left[f_\xi\big(x, y^\star(x;\xi)\big)\right], \qquad y^\star(x;\xi) := \operatorname*{argmin}_y G_\xi(x,y). \tag{3}$$

We use ∇_1, ∇_2 for derivatives in the first and second arguments and $\nabla^2_{12}, \nabla^2_{22}$ for the corresponding Hessian blocks, following Hu et al. (2023). The regularity constants below use their notation; the oracle bounds are stated conditionally on ξ.

Assumption 1. *Let $\mathcal{U} \subseteq \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ be a convex region with nonempty interior containing the query points and lower solution maps. The following conditions hold with constants uniform over samples and contexts.*

(i) For every (η,ξ), $f(\cdot,\cdot;\eta,\xi)$ is continuously differentiable and $g(\cdot,\cdot;\eta,\xi)$ is twice continuously differentiable. The expectations defining f_ξ, G_ξ and their context averages are finite.

(ii) $g(x,\cdot;\eta,\xi)$ is μ_g-strongly convex on $\mathbb{R}^{d_y}$ for every (x,η,ξ), where $\mu_g > 0$.

(iii) $f, g, \nabla f, \nabla g$, and $\nabla^2 g$ are, respectively, $L_{f,0}, L_{g,0}, L_{f,1}, L_{g,1}$, and $L_{g,2}$-Lipschitz continuous in (x,y) on $\mathcal{U}$.

(iv) The conditional derivative oracles are unbiased, and for $(x,y) \in \mathcal{U}$ their centered second moments satisfy

$$\mathbb{E}_{\eta|\xi}\|\nabla f(x,y;\eta,\xi) - \nabla f_\xi(x,y)\|^2 \le \sigma_f^2,$$
$$\mathbb{E}_{\eta|\xi}\|\nabla^j g(x,y;\eta,\xi) - \nabla^j G_\xi(x,y)\|^2 \le \sigma_{g,j}^2, \quad j \in \{1,2\}.$$

(v) Differentiation through order one for f and order two for g commutes with conditional and context expectations. Distinct update blocks use independent samples unless coupling is explicitly specified.

Strong convexity implies a unique $y^\star(x;\xi)$ and a nonsingular lower Hessian. Appendix B specifies the norm conventions and the resulting bounds for the shared-state oracles. The Lipschitz conditions apply on $\mathcal{U}$, not necessarily on the whole space.

3.1 A Direct Shared-State Extension of Single-Loop SBO

SOBA-type methods update the lower solution, an auxiliary linear-system variable, and the upper variable together (Dagréou et al., 2022); direction averaging is used in MA-SOBA (Chen et al., 2024). For their shared-state extension, define

$$\begin{aligned} H_\xi(x,y) &:= \nabla^2_{22} G_\xi(x,y), & A_\xi(x,y) &:= \nabla^2_{12} G_\xi(x,y), \\ z^\star(x;\xi) &:= H_\xi\big(x,y^\star(x;\xi)\big)^{-1} \nabla_2 f_\xi\big(x,y^\star(x;\xi)\big), & D_\xi(x,y,z) &:= \nabla_1 f_\xi(x,y) - A_\xi(x,y) z. \end{aligned} \tag{4}$$

We have the exact CSBO hypergradient $\nabla F(x) = \mathbb{E}_\xi\big[D_\xi\big(x,y^\star(x;\xi),z^\star(x;\xi)\big)\big]$ following implicit differentiation. A direct shared-state implementation replaces the context-indexed families $\{y^\star(\cdot;\xi)\}_\xi$ and $\{z^\star(\cdot;\xi)\}_\xi$ with one shared pair (y_t, z_t), carried across iterations irrespective of the context. Each derivative estimate below uses a fresh context and a conditional sample, independently across estimates. Thus $\widehat{g}_{y,t}$, $\widehat{H}_t$, $\widehat{b}_t$, $\widehat{f}_{x,t}$, and $\widehat{A}_t$ estimate the context averages of $\nabla_2 G_\xi$, H_ξ, $\nabla_2 f_\xi$, $\nabla_1 f_\xi$, and A_ξ, respectively, at the current shared states. The resulting recursion is

$$\begin{aligned} x_{t+1} &= x_t - \alpha_t h_t, \\ y_{t+1} &= y_t - \beta_t \widehat{g}_{y,t}(x_t,y_t), \\ z_{t+1} &= z_t - \gamma_t\big(\widehat{H}_t(x_t,y_t) z_t - \widehat{b}_t(x_t,y_t)\big), \\ h_{t+1} &= (1-\theta_t) h_t + \theta_t\big(\widehat{f}_{x,t}(x_t,y_t) - \widehat{A}_t(x_t,y_t) z_t\big). \end{aligned} \tag{5}$$

Here h_t averages the outer direction. The z-update is a stochastic gradient step for the context-averaged quadratic $\mathbb{E}_\xi[\frac{1}{2} z^\top H_\xi(x,y) z - \langle \nabla_2 f_\xi(x,y), z\rangle]$, with x, y held fixed. This is a natural transfer of an SBO architecture, not yet an algorithm with a guarantee for CSBO.

3.2 The Surrogate Problem Induced by Direct Transfer

Let $\bar{f} := \mathbb{E}_\xi f_\xi$ and $\bar{G} := \mathbb{E}_\xi G_\xi$, and define

$$\bar{y}(x) := \operatorname*{argmin}_y \bar{G}(x,y), \qquad \Phi(x) := \bar{f}\big(x,\bar{y}(x)\big), \tag{6}$$

with

$$\bar{z}(x) := \big[\nabla^2_{22}\bar{G}(x,\bar{y}(x))\big]^{-1} \nabla_2 \bar{f}(x,\bar{y}(x)) \tag{7}$$

Definition 1 (Shared-state direct transfer)**.** *A shared-state direct transfer preserves an SBO algorithm's initialization, updates, stepsizes, and output rule, while (a) using one shared lower state, with contexts appearing only in sample evaluations, and (b) replacing SBO samples by joint samples $\omega = (\xi,\eta) \sim \mathbb{P}_\xi \mathbb{P}_{\eta|\xi}$ while preserving the original independence, minibatching, and sample-reuse rules, without additional contextual solves or fixed-context conditional sampling.*

Theorem 1 (Surrogate target of shared-state direct transfer)**.** *Under Assumption 1:*

(i) For fixed x, the lower and auxiliary mean equations of Equation (5) have the unique joint equilibrium $(\bar{y}(x),\bar{z}(x))$, at which $\mathbb{E}_\xi D_\xi(x,\bar{y}(x),\bar{z}(x)) = \nabla\Phi(x)$.

(ii) Every direct transfer satisfying (a)–(b) has the same trajectory and output law as the original SBO algorithm on $f_\omega(x,y) := f(x,y;\eta,\xi)$ and $g_\omega(x,y) := g(x,y;\eta,\xi)$, whose expectations are $(\bar{f},\bar{G})$. the two outputs have the same distribution: $\widehat{x}_T^{\mathrm{tr}} \overset{\mathrm{d}}{=} \widehat{x}_T^{\mathrm{SBO}}$.

Proofs are given in Appendices C.1 and C.2. Part (i) identifies the equilibrium direction rather than an unbiased estimate of $\nabla\Phi(x_t)$ at arbitrary states; part (ii) transfers any applicable SBO guarantee to Φ, not to F. For the raw direction $\widehat{S}_t$ in Equation (5), let $\mathcal{F}_t$ denote the history before the fresh samples are drawn. Then

$$\begin{aligned} \widehat{S}_t &:= \widehat{f}_{x,t}(x_t,y_t) - \widehat{A}_t(x_t,y_t) z_t, \\ S_t &:= \mathbb{E}[\widehat{S}_t \mid \mathcal{F}_t] = \mathbb{E}_\xi D_\xi(x_t,y_t,z_t). \end{aligned} \tag{8}$$

The estimation error decomposes as

$$\widehat{S}_t - \nabla F(x_t) = \underbrace{\big(\nabla\Phi(x_t) - \nabla F(x_t)\big)}_{\text{structural bias } B_{\mathrm{ctx}}(x_t)} + \underbrace{\big(S_t - \nabla\Phi(x_t)\big)}_{\text{tracking error}} + \underbrace{\big(\widehat{S}_t - S_t\big)}_{\text{zero-mean noise}}. \tag{9}$$

Larger batches and accurate tracking do not remove B_{ctx}. The next example quantifies the resulting stationarity gap.

3.3 A Stationarity Gap and Failure of Direct Transfer

Consider a one-dimensional instance with $\xi = (u, \lambda)$, where u is uniform on $-1, 1$, λ is uniform on $1, 2$, and they are independent. Set $w(1) = 4$, $w(2) = -2$, so that $\mathbb{E}[w(\lambda)] = 1$ and $\mathbb{E}[\lambda w(\lambda)] = 0$. For any $A > 0$, let

$$G_\xi(x, y) = \frac{\lambda}{2}(y - u)^2, \qquad f_\xi(x, y) = 1 - \cos x + A \sin x\, w(\lambda) y u. \tag{10}$$

Since $y^\star(x; \xi) = u$,

$$F(x) = 1 - \cos x + A \sin x, \qquad \nabla F(x) = \sin x + A \cos x. \tag{11}$$

In contrast, $\nabla_2 \bar{G}(x, y) = \frac{3}{2} y$, so $\bar{y}(x) = 0$ and

$$\Phi(x) = 1 - \cos x, \qquad \nabla\Phi(x) = \sin x. \tag{12}$$

Hence every $x = k\pi$ is stationary for Φ, while $|\nabla F(k\pi)| = A$.

Theorem 2 (Surrogate stationarity does not transfer to CSBO)**.** *For instance Equation (10) with any $A > 0$, let $\widehat{x}_T$ be any sequence of random outputs and define $q_T := \mathbb{E}|\nabla\Phi(\widehat{x}_T)|^2 \in [0, 1]$. Then*

$$\big|\mathbb{E}|\nabla F(\widehat{x}_T)|^2 - A^2\big| \le |1 - A^2| q_T + 2A\sqrt{q_T(1 - q_T)}. \tag{13}$$

In particular, $q_T \to 0$ implies $\mathbb{E}|\nabla F(\widehat{x}_T)|^2 \to A^2 > 0$. Moreover, $(x, y, z, h) = (0, 0, 0, 0)$ is a fixed point of the exact population recursion Equation (5), with $|\nabla F(0)| = A$. The instance satisfies Assumption 1 on invariant bounded lower/auxiliary-state regions for the stepsizes specified in Appendix C.3.

The proof is given in Appendix C.3. By Theorem 1(ii), an admissible direct transfer inherits the original SBO algorithm's stationarity guarantee for Φ, whenever its convergence assumptions hold. On this instance, however, vanishing surrogate stationarity error implies $\mathbb{E}|\nabla F(\widehat{x}_T)|^2 \to A^2 > 0$. Thus, convergence guarantees for the surrogate do not generally transfer to CSBO. This motivates correcting the shared-state direction while retaining the shared lower and auxiliary states.

4 Randomly Corrected Single-Loop Methods

We retain the shared states but occasionally correct their outer direction using context-specific conditional sampling.

4.1 Randomized Correction Principle

For the shared population direction S_t in Equation (8), define

$$C_t := \nabla F(x_t) - S_t = \mathbb{E}_\xi\big[D_\xi\big(x_t, y^\star(x_t; \xi), z^\star(x_t; \xi)\big) - D_\xi(x_t, y_t, z_t)\big]. \tag{14}$$

Thus C_t corrects both the structural bias and tracking error in Equation (9). Let $\widehat{S}_t$ satisfy $\mathbb{E}[\widehat{S}_t \mid \mathcal{F}_t] = S_t$, and let an inexact contextual correction satisfy

$$\mathbb{E}[\widetilde{C}_t \mid \mathcal{F}_t] = C_t + e_t,$$

where e_t is its finite-solve error. Drawing $I_t \mid \mathcal{F}_t \sim \mathrm{Bernoulli}(p_t)$ independently and setting

$$v_t := \widehat{S}_t + \frac{I_t}{p_t}\widetilde{C}_t \qquad \Longrightarrow \qquad \mathbb{E}[v_t \mid \mathcal{F}_t] = \nabla F(x_t) + e_t \tag{15}$$

shows that random activation adds no bias beyond the contextual-solve error. Under conditional independence,

$$\mathbb{E}\Big[\|v_t - \mathbb{E}[v_t \mid \mathcal{F}_t]\|^2 \Big| \mathcal{F}_t\Big] \le \sigma_{S,t}^2 + \frac{M_{C,t}^2}{p_t}, \tag{16}$$

where $\sigma_{S,t}^2 := \mathbb{E}[\|\widehat{S}_t - S_t\|^2 \mid \mathcal{F}_t]$ and $M_{C,t}^2 := \mathbb{E}[\|\widetilde{C}_t\|^2 \mid \mathcal{F}_t]$. If the two estimators cost $c_{S,t}$ and $c_{C,t}$, respectively, the expected per-iteration cost is $c_{S,t} + p_t c_{C,t}$. Hence smaller p_t lowers correction cost at the price of increased variance.

4.2 The RCSL-GA Algorithm

RCSL-GA combines the shared-state direction in Equation (5) with the randomized correction above. At each iteration it computes the cheap estimator $\widehat{S}_t$ and, with probability p_t, invokes Algorithm 2 to estimate the contextual–shared direction difference.

Algorithm 2 (Appendix A) warm-starts B_c independent contextual solves from (y_t, z_t) and averages their contextual–shared direction differences. Correction samples are independent of the shared-state estimates, and each paired difference uses an independent terminal batch. Writing the terminal contextual states as $y_t^c(\xi)$ and $z_t^c(\xi)$ gives

$$\mathbb{E}[\widetilde{C}_t^{\mathrm{GA}} \mid \mathcal{F}_t] = C_t + e_t^{\mathrm{GA}}, \qquad e_t^{\mathrm{GA}} = \mathbb{E}\big[D_\xi\big(x_t, y_t^c(\xi), z_t^c(\xi)\big) - D_\xi\big(x_t, y^\star(x_t;\xi), z^\star(x_t;\xi)\big)\big|\mathcal{F}_t\big]. \tag{17}$$

Hence the only correction bias is the mean hypergradient error of the finite contextual solves, while averaging B_c independent contexts reduces the correction variance by a factor B_c; precise bounds appear in Appendix D.1.

For the direction-averaged variant, replace the raw outer update by

$$x_{t+1} = x_t - \alpha_t h_t, \qquad h_{t+1} = (1-\theta_t)h_t + \theta_t v_t,$$

with $h_0 = 0$ and $\theta_t \in (0, 1]$.

5 Convergence and Complexity

5.1 Convergence and Oracle Complexity of RCSL-GA

We analyze the raw update $x_{t+1} = x_t - \alpha v_t$ with the GA correction in Algorithms 1–2. Assume Assumption 1, including a common convex region $\mathcal{U}$ containing all shared and contextual query points and lower solution maps almost surely, with constants independent of T. Let $F_{\inf} := \inf_x F(x) > -\infty$ and $\Delta := F(x_0) - F_{\inf}$. The sample-level Lipschitz bounds in Assumption 1 already imply the oracle moment bounds needed below. Set

$$\beta_t = \gamma_t = \lambda_y = \lambda_z = \tau := \frac{1}{2L_{g,1}}, \qquad q := 1 - \mu_g \tau \in (0,1). \tag{18}$$

Theorem 3 (Exact convergence in expected stationarity)**.** *Fix deterministic initial states, $p \in (0,1]$, and positive integers m_s, B_c, N_e independently of T, where m_s is the shared primitive batch size. For $T \ge 1$, use Equation (18), $N_z = 1$, $N_y = \max\{1, \lceil \sigma_{g,1}^2 \sqrt{T} \rceil\}$, and*

$$\alpha = \frac{1}{\sqrt{(1+p^{-1})T}}, \qquad K_y = K_z = \max\left\{1, \left\lceil \frac{\log T}{4\log(1/q)} \right\rceil\right\}. \tag{19}$$

Then, for $R \sim \mathrm{Unif}\{0, \dots, T-1\}$ and $\widehat{x}_T = x_R$, we have $\mathbb{E}\|\nabla F(\widehat{x}_T)\|^2 \le \frac{C_p}{\sqrt{T}}$, where C_p is independent of T and $C_p = O(\sqrt{1+p^{-1}})$ for fixed problem parameters and initialization.

The proof controls the conditional mean, rather than the mean-square error, of the auxiliary solve. Once the contextual lower output is fixed, the auxiliary mean contracts to the corresponding linear-system solution; because the independent terminal direction is affine in the auxiliary variable, $N_z = 1$ introduces no bias floor. The lower-solve noise is controlled through N_y.

Algorithm 1 RCSL-GA framework

Input: (x_0, y_0, z_0), T, stepsizes $\{\alpha_t, \beta_t, \gamma_t\}$, probabilities $\{p_t\}$, and GA parameters

for $t = 0, \dots, T-1$ **do**
  Compute $\widehat{S}_t$ from Equation (5) at (x_t, y_t, z_t)
  Draw $I_t \sim \text{Bernoulli}(p_t)$ and set $\widetilde{C}_t \leftarrow 0$
  if $I_t = 1$ **then**
    $\widetilde{C}_t \leftarrow \mathsf{GA}(x_t, y_t, z_t)$ (Algorithm 2)
  end if
  $v_t \leftarrow \widehat{S}_t + (I_t/p_t)\widetilde{C}_t, \qquad x_{t+1} \leftarrow x_t - \alpha_t v_t$
  Update (y_{t+1}, z_{t+1}) from Equation (5) at (x_{t+1}, y_t, z_t)
end for

Output: $\widehat{x}_T \sim \text{Unif}\{x_0, \dots, x_{T-1}\}$

Corollary 4 (Iteration and oracle complexity). *Let* $T = \max\{1, \lceil C_p^2 \epsilon^{-4} \rceil\}$ *yields*

$$\mathbb{E}\|\nabla F(\widehat{x}_T)\|^2 \le \epsilon^2, \qquad T = O((1 + p^{-1})\epsilon^{-4}).$$

For fixed p, m_s, B_c, N_e, *the expected total first- and second-order oracle calls satisfy* $\mathbb{E}N_g = \widetilde{O}(\epsilon^{-6})$ *and* $\mathbb{E}N_h = \widetilde{O}(\epsilon^{-4})$,

Proofs are in Appendix E. The guarantee concerns expected stationarity of a random output under a horizon-dependent refinement schedule; it is not a last-iterate or fixed-inner-accuracy result, and it does not cover direction-averaged or Adam-type outer updates.

5.2 Per-Iteration Oracle Cost

To compare fixed-parameter per-iteration work, let c_g and c_h be the costs of a partial gradient and a Hessian– or mixed-Jacobian–vector product, and set $\rho := c_h/c_g$. Using common primitive batch size m, take $N_y = N_z = N_e = m$ and $K_y = K_z = K$ for RCSL-GA. For RT-MLMC, let $\chi := \mathbb{E}[2^L]$, where L is the sampled level; the Hessian variant samples Neumann lengths from $\{0, \dots, N-1\}$. Define $D(\rho) := B_c[2K + 2 + \rho(K+2)]$. Then the expected per-iteration work is $mc_g\Gamma$, with Γ given in Table 1.

Table 1: Expected per-iteration work, normalized by mc_g.

Method	Γ
RCSL-GA	$3 + 2\rho + pD(\rho)$
FFO-RT-MLMC	$9 + 5\chi$
Hessian-RT-MLMC	$2\chi + 4 + \frac{3}{2}\rho(N+1)$

RCSL-GA pays the shared-update cost every iteration and the contextual-refinement cost only with probability p. The resulting expressions give explicit thresholds on p for lower expected per-iteration work than the compared baselines; derivations and examples are in Appendix F.

This fixed-parameter comparison does not imply lower total work at matched stationarity accuracy: controlling conditional lower-gradient noise requires increasing refinement accuracy, giving $\widetilde{O}(\epsilon^{-6})$ total primitive calls in the general stochastic case, while second-order calls remain $\widetilde{O}(\epsilon^{-4})$. Smaller p also increases estimator variance. Section 6 evaluates the resulting wall-clock trade-off.

6 Experiments

We first illustrate the shared-state bias, then compare optimization progress and wall-clock performance on meta-learning and task-aggregated power scheduling. All methods use common data partitions and evaluation routines; details are in Appendix G.

6.1 Structural-Bias Example

We use Equation (10) with $A = 4$ and compare the shared-state SL transfer Equation (5), based on SOBA/MA-SOBA (Dagréou et al., 2022; Chen et al., 2024), with RCSL-GA. Here all stochasticity

comes from the discrete contexts in Section 3.3. The true objective has minimizer $x_F^\star = -\arctan A$ and value $1 - \sqrt{1 + A^2}$. We report x_t, $|\nabla F(x_t)|^2$, and $F(x_t) - F(x_F^\star)$ over 20 seeds. We set $p = 1$ to isolate contextual correction rather than oracle cost; further settings are in Appendix G.1.

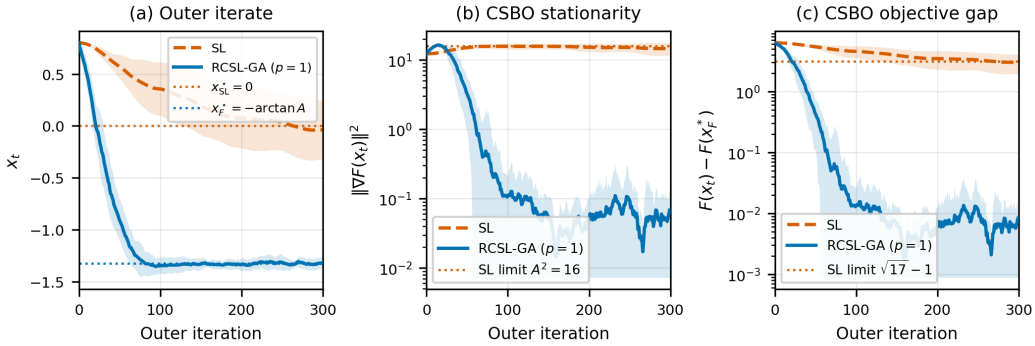


Figure 1: Structural-bias example ($A = 4$). Curves and bands show the mean and one standard deviation over 20 seeds; the dashed line marks the population SL stationarity error A^2.

As shown in Figure 1, SL approaches the surrogate stationary point $x_{\mathrm{SL}}^\star = 0$ while retaining nonzero CSBO stationarity and objective errors, whereas RCSL-GA approaches $x_F^\star$ and reduces both errors by several orders of magnitude.

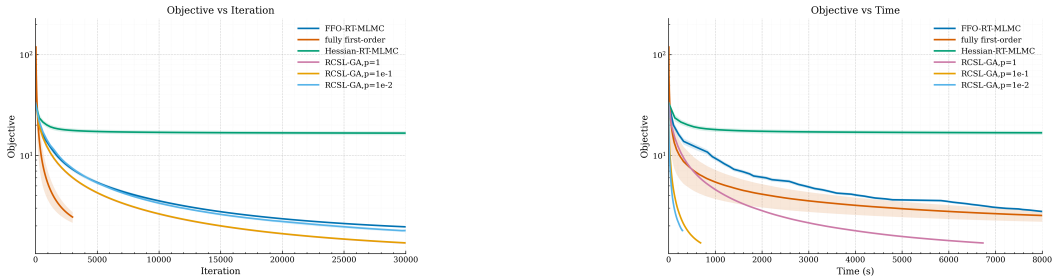
6.2 Meta-Learning

In regularization-based meta-learning, each context is a classification task whose classifier adapts to a shared regularization center x (Hu et al., 2023):

$$\begin{aligned} \min_{x} \quad & F_{\mathrm{meta}}(x) := \mathbb{E}_{i\sim\mu}\mathbb{E}_{D_i^{\mathrm{val}}\sim\mathcal{P}_i^{\mathrm{val}}}\left[\ell_i\big(y_i^\star(x); D_i^{\mathrm{val}}\big)\right], \\ & y_i^\star(x) := \operatorname*{argmin}_{y_i} \mathbb{E}_{D_i^{\mathrm{tr}}\sim\mathcal{P}_i^{\mathrm{tr}}}\left[\ell_i(y_i; D_i^{\mathrm{tr}}) + \frac{\lambda}{2}\|y_i - x\|^2\right]. \end{aligned} \tag{20}$$

We use $\lambda = 2$ and construct ten 20-way Tiny-ImageNet tasks covering all 200 classes (mnmoustafa & Ali, 2017), with fixed 512-dimensional features from an ImageNet-pretrained ResNet-18 (He et al., 2016; Deng et al., 2009). The validation split supplies both the upper training objective and the reported loss, so the experiment measures optimization on the trained tasks.

We compare RCSL-GA with FFO, FFO-RT-MLMC (Yang et al., 2025), and Hessian-RT-MLMC (Hu et al., 2023). FFO uses $T_k = k$ inner iterations and is run for 3,000 outer iterations, versus 30,000 for the other methods. Each configuration uses ten seeds. At reporting points, classifiers are recomputed with a common adaptation solver and evaluated on the full validation split; evaluation time is excluded. Full settings are in Appendices G.2.2 and G.2.3.



(a) Objective versus outer iterations

(b) Objective versus wall-clock time

Figure 2: Meta-learning upper objective under a common adaptation solver; mean and one standard deviation over ten seeds.

Figure 2 shows that RCSL-GA attains the lowest final objective among methods run for 30,000 iterations and reaches low objective values earlier in wall-clock time. Increasing p from 0.1 to 1 improves the final objective by only 0.25% while increasing runtime by 9.63×, indicating that correction at every iteration is unnecessary here. Additional summaries and ablations appear in Appendices G.2.4–G.2.6.

6.3 Task-Aggregated Power Scheduling

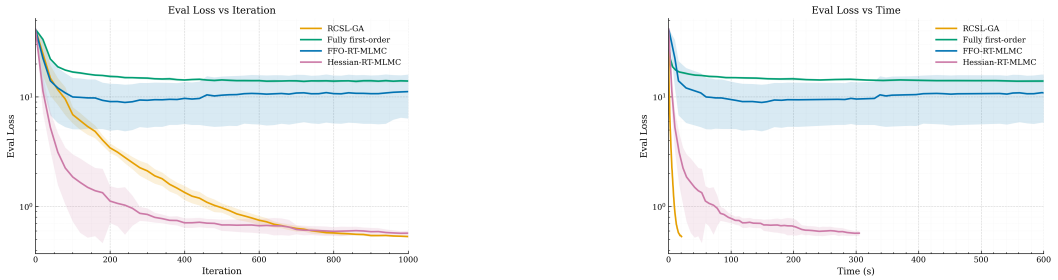
We consider a task-aggregated surrogate motivated by day-ahead load forecasting and generator scheduling (Donti et al., 2017; Mandi et al., 2024). Context $i \sim \mu$ indexes a temporal task, and $D = (a, d) \sim \mathcal{P}_i^{\mathrm{tr}}$ contains features $a \in \mathbb{R}^{125}$ and next-day hourly load $d \in \mathbb{R}^{24}$. For each task, the predictor induces one representative 24-hour schedule, which set fixed across its evaluation days.

With affine predictor $r_x(a) = W^\top a + b$, ramp-feasible set $\mathcal{Z} = \{s \in \mathbb{R}^{24} : |s_j - s_{j-1}| \le c_{\mathrm{ramp}},\ j = 2, \dots, 24\}$, and the smoothed scheduling cost c_τ defined in Appendix G.3.2, we solve

$$\begin{aligned} \min_x \quad & F_{\mathrm{power},\tau}(x) := \mathbb{E}_{i\sim\mu}\mathbb{E}_{(a,d)\sim\mathcal{P}_i^{\mathrm{tr}}}\left[c_\tau\left(s_i^\star(x), d\right)\right], \\ & s_i^\star(x) := \operatorname*{argmin}_{s\in\mathcal{Z}} \mathbb{E}_{(a,d)\sim\mathcal{P}_i^{\mathrm{tr}}}\left[c_\tau\left(s, r_x(a)\right) + \frac{\gamma}{2}\|s - r_x(a)\|^2\right]. \end{aligned} \tag{21}$$

Here $x = (\mathrm{vec}(W), b)$ is shared across tasks, and the quadratic term makes the lower objective strongly convex. Appendix G.3.1 relates this surrogate to feature-conditioned ETO.

We form four rolling PJM tasks, each with five training years followed by one evaluation year. Because all tasks jointly train x, earlier evaluation observations may enter later training windows; the split is therefore chronological within each task but not a global future holdout. We compare projected versions of the same four methods over ten seeds. Reporting uses the original piecewise-linear per-hour cost, with each task schedule recomputed from its training data; evaluation time is excluded. Details are in Appendices G.3.3 and G.3.4.



(a) Decision loss versus outer iterations

(b) Decision loss versus wall-clock time

Figure 3: Task-aggregated scheduling loss, averaged first over evaluation days and then equally over tasks; mean and one standard deviation over ten seeds.

Figure 3 shows that RCSL-GA and Hessian-RT-MLMC reach lower decision loss per iteration than the FFO methods, while RCSL-GA reaches the low-loss region much earlier in wall-clock time. The constrained lower problem lies outside our smooth unconstrained theory, so this experiment provides empirical evidence on the surrogate benchmark rather than a stationarity guarantee for the constrained bilevel objective.

7 Conclusion

We characterized the surrogate target and structural bias of a class of shared-state transfers from SBO to CSBO. Our RCSL-GA method combines inexpensive shared updates with occasional contextual corrections, supported by convergence analysis and favorable empirical wall-clock efficiency. Future work includes adaptive correction probabilities, more efficient contextual refinement, and extensions to constrained lower-level problems.

AI USE STATEMENT

In this work, we used generative AI tools, primarily ChatGPT, to assist with exploring candidate algorithmic approaches and proof strategies, developing and refining mathematical arguments, surveying relevant literature, writing and debugging parts of experimental code, and drafting and polishing parts of the manuscript. The authors formulated the research questions and problem settings, critically evaluated and selected among candidate approaches, manually reviewed and verified the mathematical claims and proofs included in the final paper, and reviewed and tested the AI-assisted code used in the final experiments. The authors have reviewed all the work and take full responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of both the theoretical and empirical results presented in this work. The assumptions and complete proofs of the main theoretical results are provided in the appendix. Detailed descriptions of the experimental settings, including problem formulations, dataset preprocessing, model configurations, hyperparameters, and implementation details, are also included in the appendix. In addition, the supplementary material contains the source code used to implement the proposed methods and reproduce the experiments and figures reported in the paper.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019. 2, G.3.2

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pp. 136–145. PMLR, 2017. 2, G.3.2

Maxime Bouscary, Jiawei Zhang, and Saurabh Amin. Reducing contextual stochastic bilevel optimization via structured function approximation. In *International Conference on Learning Representations*, volume 2026, pp. 87592–87622, 2026. 1, 2

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34: 25294–25307, 2021. 1, 2

Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 2466–2488. PMLR, 2022. 1, 2

Xuxing Chen, Tesi Xiao, and Krishnakumar Balasubramanian. Optimal algorithms for stochastic bilevel optimization under relaxed smoothness conditions. *Journal of Machine Learning Research*, 25(151):1–51, 2024. 1, 2, 3.1, 6.1

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35:26698–26710, 2022. 1, 2, 3.1, 6.1

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. How to compute hessian-vector products? In *The Third Blogpost Track at ICLR 2024*, 2024. F.1, F.7

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. 6.2, G.2.2

Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, 30, 2017. 1, 2, 6.3, G.3.1, G.3.2, G.3.3

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management science*, 68(1): 9–26, 2022. 2, G.3.1

Adam N Elmachtoub, Henry Lam, Haofeng Zhang, and Yunfan Zhao. Estimate-then-optimize versus integrated-estimation-optimization versus sample average approximation: a stochastic dominance perspective. *arXiv preprint arXiv:2304.06833*, 2023. 2, G.3.1

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018. 2

Zhishuai Guo, Quanqi Hu, Lijun Zhang, and Tianbao Yang. Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021. 2

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 6.2, G.2.2

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023. 1, 2

Yifan Hu, Jie Wang, Yao Xie, Andreas Krause, and Daniel Kuhn. Contextual stochastic bilevel optimization. *Advances in Neural Information Processing Systems*, 36:78412–78434, 2023. 1, 1, 2, 3, 6.2, 6.2, F.4, F.6, G.2.1, G.2.3, 3, G.3.4, 6

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pp. 4882–4892. PMLR, 2021. 1, 2

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in neural information processing systems*, 34:30271–30283, 2021. 1, 2

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pp. 18083–18113. PMLR, 2023. 2

Jeongyeol Kwon, Dohyun Kwon, and Hanbaek Lyu. On the complexity of first-order methods in stochastic bilevel optimization. *arXiv preprint arXiv:2402.07101*, 2024a. 2

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. In *International Conference on Learning Representations*, 2024b. 2

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024. 2, 6.3, G.3.1

mnmoustafa and Mohammed Ali. Tiny imagenet. https://www.kaggle.com/competitions/tiny-imagenet, 2017. Kaggle. 6.2, G.2.2

Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994. F.1

Meng Qi, Paul Grigas, and Zuo-Jun Shen. Integrated conditional estimation-optimization. *Operations Research*, 74(3):1604–1625, 2026. 2

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2025. 2

Zhichun Yang, Ying Lin, Daniel Kuhn, Yifan Hu, and Man-Chung Yue. Fully first-order methods for contextual stochastic bilevel optimization. *OpenReview preprint*, 2025. 1, 2, 6.2, F.3, F.6, G.2.3, 3, 1, G.3.4, 6

A GA CORRECTION

Algorithm 2 GA Correction

Input: State (x, y, z), context batch size B_c, inner lengths K_y, K_z, batch sizes N_y, N_z, N_e, and stepsizes λ_y, λ_z

$\widetilde{C} \leftarrow 0$
for $r = 1, \dots, B_c$ **do**
  Sample $\xi_r \sim \mathbb{P}_\xi$ and set $y_{r,0} \leftarrow y$
  for $k = 0, \dots, K_y - 1$ **do**
    Sample $\{\eta^y_{r,k,j}\}_{j=1}^{N_y} \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{\eta|\xi_r}$, update $y_{r,k+1} \leftarrow y_{r,k} - \dfrac{\lambda_y}{N_y} \displaystyle\sum_{j=1}^{N_y} \nabla_2 g(x, y_{r,k}; \eta^y_{r,k,j}, \xi_r)$
  end for
  Set $z_{r,0} \leftarrow z$
  for $k = 0, \dots, K_z - 1$ **do**
    Sample $\{\eta^z_{r,k,j}\}_{j=1}^{N_z} \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{\eta|\xi_r}$
    $z_{r,k+1} \leftarrow z_{r,k} - \dfrac{\lambda_z}{N_z} \displaystyle\sum_{j=1}^{N_z} \big[\nabla^2_{22} g(x, y_{r,K_y}; \eta^z_{r,k,j}, \xi_r) z_{r,k} - \nabla_2 f(x, y_{r,K_y}; \eta^z_{r,k,j}, \xi_r)\big]$
  end for
  Sample $\{\eta^e_{r,j}\}_{j=1}^{N_e} \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{\eta|\xi_r}$
  $\widehat{c}_r \leftarrow \dfrac{1}{N_e} \displaystyle\sum_{j=1}^{N_e} \big[\nabla_1 f(x, y_{r,K_y}; \eta^e_{r,j}, \xi_r) - \nabla^2_{12} g(x, y_{r,K_y}; \eta^e_{r,j}, \xi_r) z_{r,K_z}$
  $\qquad - \nabla_1 f(x, y; \eta^e_{r,j}, \xi_r) + \nabla^2_{12} g(x, y; \eta^e_{r,j}, \xi_r) z\big]$
  $\widetilde{C} \leftarrow \widetilde{C} + \widehat{c}_r / B_c$
end for

Output: $\widetilde{C}$

B NOTATION AND REGULARITY BOUNDS

For a function $\psi(x, y)$, $\nabla\psi$ and $\nabla^2\psi$ denote its full gradient and Hessian, while $\nabla_1\psi, \nabla_2\psi$ and $\nabla^2_{12}\psi, \nabla^2_{22}\psi$ denote the partial gradients, mixed block, and lower Hessian. The mixed block has size $d_x \times d_y$; the Jacobian $J_{\bar{y}}$ has size $d_y \times d_x$. Vector norms are Euclidean and matrix norms are operator norms. Conditional expectations given the algorithmic history are written as $\mathbb{E}[\cdot \mid \mathcal{F}_t]$.

Regularity constants. For $v, v' \in \mathcal{U}$, Assumption 1(iii) means $\|\psi(v; \eta, \xi) - \psi(v'; \eta, \xi)\| \le L_\psi \|v - v'\|$, with absolute values for scalar functions and

$$(\psi, L_\psi) \in \{(f, L_{f,0}), (g, L_{g,0}), (\nabla f, L_{f,1}), (\nabla g, L_{g,1}), (\nabla^2 g, L_{g,2})\}.$$

Differentiability and these Lipschitz bounds imply

$$\|\nabla f\| \le L_{f,0}, \qquad \|\nabla g\| \le L_{g,0}, \qquad \|\nabla^2 g\| \le L_{g,1} \quad \text{on } \mathcal{U}. \tag{22}$$

For example, at an interior point, applying the Lipschitz inequality to $v + tu$ and v for a unit vector u, dividing by $|t|$, and taking $t \to 0$ bounds the directional derivative. Taking the supremum over u gives each displayed bound; continuity gives the boundary case. Conditional and context expectations inherit the same Lipschitz constants by the triangle inequality. Averaging the sample strong-convexity inequality also gives $\nabla^2_{22} G_\xi \succeq \mu_g I$ and $\nabla^2_{22} \widetilde{G} \succeq \mu_g I$. The regional qualification is essential: global Lipschitz continuity of g is not required on the unbounded strongly convex lower domain.

Oracle moments. Conditional unbiasedness means

$$\mathbb{E}_{\eta|\xi} \nabla f(x, y; \eta, \xi) = \nabla f_\xi(x, y), \qquad \mathbb{E}_{\eta|\xi} \nabla^j g(x, y; \eta, \xi) = \nabla^j G_\xi(x, y), \quad j \in \{1, 2\}.$$

The symbols $\sigma_f, \sigma_{g,1}, \sigma_{g,2}$ bound conditional derivative noise; the last two distinguish gradient and Hessian noise. They do not by themselves bound variation between contexts. For a fresh joint

sample $\omega = (\xi, \eta)$, however, Equation (22) gives

$$\mathbb{E}_\omega\|\nabla f_\omega - \nabla \bar{f}\|^2 \le L_{f,0}^2, \qquad \mathbb{E}_\omega\|\nabla g_\omega - \nabla \bar{G}\|^2 \le L_{g,0}^2,$$
$$\mathbb{E}_\omega\|\nabla^2 g_\omega - \nabla^2 \bar{G}\|^2 \le 4L_{g,1}^2.$$

The vector bounds use the centered-second-moment identity; the matrix bound uses $\|M - \mathbb{E}M\| \le 2L_{g,1}$ for $M = \nabla^2 g_\omega$. Thus, for a history-measurable vector z with $\|z\| \le R_z$, centered Hessian– or mixed-block–vector estimates have second moment at most $4L_{g,1}^2 R_z^2$. Bounded auxiliary states are an additional requirement when a uniform update-variance bound is used; they are verified for the counterexample in Appendix C.3.

C Proofs for the Structural-Bias Results

Write $\omega = (\xi, \eta)$ with $\mathbb{P}(d\omega) = \mathbb{P}_\xi(d\xi)\mathbb{P}_{\eta|\xi}(d\eta)$. The direct-transfer protocol in Section 3.2 is formalized below.

Definition 2 (Joint-sample shared-state direct transfer)**.** *Given a single-loop SBO algorithm, its direct transfer retains its initialization, update maps, stepsize rules, and output rule, subject to the following oracle protocol:*

1. *It maintains one shared lower variable, possibly with auxiliary, momentum, or oracle-memory states, but no context-indexed lower solutions or parameterized context-to-response map. Context labels enter only through sample-function evaluations.*
2. *Every fresh sample is a draw of the entire pair ω from $\mathbb{P}$. Independence, minibatching, and sample-reuse rules match the original SBO algorithm: reuse retains the entire pair, including when a sample is evaluated at multiple query points. No context-specific lower solves or additional fixed-context conditional-sampling requests are introduced.*

C.1 Proof of Theorem 1(i)

For this proof, abbreviate

$$\begin{aligned} \bar{H}(x) &:= \mathbb{E}_\xi H_\xi(x, \bar{y}(x)) = \nabla_{22}^2 \bar{G}(x, \bar{y}(x)), \\ \bar{b}(x) &:= \mathbb{E}_\xi \nabla_2 f_\xi(x, \bar{y}(x)) = \nabla_2 \bar{f}(x, \bar{y}(x)), \qquad \bar{z}(x) = \bar{H}(x)^{-1}\bar{b}(x). \end{aligned} \tag{23}$$

Proof. Step 1: conditional mean updates. Let $\mathcal{F}_t$ contain the states and stepsizes before the fresh samples for iteration t are drawn. By conditional unbiasedness, fresh context sampling, and the tower property,

$$\mathbb{E}[\widehat{g}_{y,t}(x_t, y_t) \mid \mathcal{F}_t] = \mathbb{E}_\xi[\nabla_2 G_\xi(x_t, y_t)] = \nabla_2 \bar{G}(x_t, y_t).$$

The last equality uses Assumption 1(v). The same argument applies to $\widehat{H}_t$, $\widehat{b}_t$, $\widehat{f}_{x,t}$, and $\widehat{A}_t$. Since z_t is $\mathcal{F}_t$-measurable, it can be taken outside each conditional expectation of a matrix–vector product. Consequently, all four updates satisfy

$$\begin{aligned} \mathbb{E}[x_{t+1} \mid \mathcal{F}_t] &= x_t - \alpha_t h_t, \\ \mathbb{E}[y_{t+1} \mid \mathcal{F}_t] &= y_t - \beta_t \nabla_2 \bar{G}(x_t, y_t), \\ \mathbb{E}[z_{t+1} \mid \mathcal{F}_t] &= z_t - \gamma_t\big(\nabla_{22}^2 \bar{G}(x_t, y_t) z_t - \nabla_2 \bar{f}(x_t, y_t)\big), \\ \mathbb{E}[h_{t+1} \mid \mathcal{F}_t] &= (1 - \theta_t) h_t + \theta_t\big(\nabla_1 \bar{f}(x_t, y_t) - \nabla_{12}^2 \bar{G}(x_t, y_t) z_t\big). \end{aligned} \tag{24}$$

These are conditional drifts evaluated at the current random states; they are not a closed recursion for the unconditional state means.

Step 2: the shared-state equilibria. Averaging the strong-convexity inequality for $G_\xi(x, \cdot)$ gives, for any y, v,

$$\bar{G}(x, v) \ge \bar{G}(x, y) + \langle \nabla_2 \bar{G}(x, y), v - y\rangle + \frac{\mu_g}{2}\|v - y\|^2.$$

Thus $\bar{G}(x, \cdot)$ has the unique minimizer $\bar{y}(x)$, characterized by $\nabla_2 \bar{G}(x, \bar{y}(x)) = 0$. For fixed x and positive lower stepsize, this is exactly the zero-drift condition of the lower update. Moreover, $\bar{H}(x) = \nabla_{22}^2 \bar{G}(x, \bar{y}(x)) \succeq \mu_g I$. At $y = \bar{y}(x)$, the auxiliary zero-drift equation is therefore $\bar{H}(x) z = \bar{b}(x)$ and has the unique solution $\bar{z}(x)$.

Step 3: the outer target. The implicit function theorem applies because $\bar{H}(x)$ is nonsingular. Differentiating the lower optimality condition gives

$$\nabla^2_{21}\bar{G}(x,\bar{y}(x)) + \bar{H}(x)J_{\bar{y}}(x) = 0, \qquad J_{\bar{y}}(x) = -\bar{H}(x)^{-1}\nabla^2_{21}\bar{G}(x,\bar{y}(x)),$$

where $J_{\bar{y}}$ is the $d_y \times d_x$ Jacobian. Using symmetry of $\bar{H}$ and the chain rule, we obtain

$$\begin{aligned}\nabla\Phi(x) &= \nabla_1\bar{f}(x,\bar{y}(x)) + J_{\bar{y}}(x)^\top\nabla_2\bar{f}(x,\bar{y}(x)) \\ &= \nabla_1\bar{f}(x,\bar{y}(x)) - \nabla^2_{12}\bar{G}(x,\bar{y}(x))\bar{H}(x)^{-1}\bar{b}(x) \\ &= \mathbb{E}_\xi[D_\xi(x,\bar{y}(x),\bar{z}(x))].\end{aligned}$$

This proves Theorem 1 (i). For positive averaging weight, the mean h-equation has equilibrium $h = \nabla\Phi(x)$ at these shared states. Hence, with a positive upper stepsize, a joint population fixed point also requires $\nabla\Phi(x) = 0$, not necessarily $\nabla F(x) = 0$. □

C.2 Proof of Theorem 1(ii)

Proof. Step 1: identify the stochastic SBO problem. By the definition of the joint law and iterated expectation,

$$\mathbb{E}_\omega f_\omega(x,y) = \mathbb{E}_\xi\mathbb{E}_{\eta|\xi}f(x,y;\eta,\xi) = \bar{f}(x,y), \qquad \mathbb{E}_\omega g_\omega(x,y) = \bar{G}(x,y).$$

The SBO problem associated with these two population losses has lower response $\bar{y}(x) = \arg\min_y \bar{G}(x,y)$ and outer objective $\Phi(x) = \bar{f}(x,\bar{y}(x))$. For derivative oracles, the corresponding population identities follow from Assumption 1(v). This identifies the oracle's population losses, without taking expectations through inverses or other nonlinear algorithmic operations.

Step 2: couple the two executions. Call the directly transferred execution run C and the original SBO algorithm on (f_ω, g_ω) run S. Couple their initial states, internal random choices, and fresh joint samples to be identical. For every fresh sample identifier, both runs receive the same draw of $\omega = (\xi,\eta)$ from $\mathbb{P}$. Identical identifiers refer to identical stored pairs whenever samples are reused. This coupling is valid because Definition 2 preserves the entire sampling protocol, not merely the marginal law of one derivative.

We prove equality of histories by induction over successive algorithmic operations, including oracle-based initialization. Before the first oracle call, the initial states and histories agree by construction. Suppose they agree before the next operation. The update or query rule uses the same history and internal randomness, so both runs choose the same query point, oracle type, and sample identifiers. At a fresh identifier they receive the same newly drawn pair; at a reused identifier they retrieve the same pair from their identical histories. They consequently evaluate the same sample function or derivative at the same point, and receive the same response. If the operation instead updates a state or a stepsize, the same update map applied to identical inputs gives identical results. The histories therefore agree after the operation. Induction establishes equality at every finite iteration, including all auxiliary states and oracle memory. Minibatches are covered by applying this argument to each of their sample identifiers.

Step 3: outputs and guarantees. The output rules also agree, including any stopping decisions or randomized iterate selection, because the histories and internal randomness agree. Thus the outputs are pathwise equal under the coupling and have the same distribution without it. Any SBO stationarity bound valid for run S therefore holds for run C with the same objective Φ. For example, if the original theorem gives $\mathbb{E}\|\nabla\Phi(\widehat{x}^{\mathrm{S}}_T)\|^2 \le r_T$, then the same bound holds for $\widehat{x}^{\mathrm{C}}_T$. The argument does not establish the original theorem's assumptions, nor does it replace Φ by F in that bound. □

C.3 Proof of Theorem 2

Proof. Step 1: exact and surrogate objectives. The derivatives of the lower objective are $\nabla_2 G_\xi = \lambda(y-u)$, $\nabla^2_{22}G_\xi = \lambda$, and $\nabla^2_{12}G_\xi = 0$. Since $1 \le \lambda \le 2$, the lower objective is 1-strongly convex and 2-smooth, with unique minimizer $y^\star(x;\xi) = u$. Using $u^2 = 1$ and $\mathbb{E}[w(\lambda)] = (4-2)/2 = 1$ gives

$$F(x) = 1 - \cos x + A\sin x\,\mathbb{E}[w(\lambda)]\mathbb{E}[u^2] = 1 - \cos x + A\sin x.$$

Independence of u and λ, together with $\mathbb{E}u = 0$, gives

$$\bar{G}(x,y) = \tfrac{3}{4}(y^2+1), \qquad \bar{y}(x) = 0, \qquad \bar{f}(x,y) = 1-\cos x, \qquad \Phi(x) = 1-\cos x.$$

Differentiation now yields $\nabla F(x) = \sin x + A\cos x$ and $\nabla\Phi(x) = \sin x$. In particular, the surrogate stationary set is $\{k\pi : k \in \mathbb{Z}\}$, and $|\nabla F(k\pi)| = A$ at every such point.

Step 2: the population fixed point. We have $\bar{H} = 3/2$ and $\bar{b} = A\sin x\,\mathbb{E}[w(\lambda)u] = 0$, so $\bar{z} = 0$. At $(x, y, z, h) = (0, 0, 0, 0)$ the four drifts in Equation (24) are respectively $-\alpha h = 0$, $-\beta(3y/2) = 0$, $-\gamma(3z/2) = 0$, and $\theta(\sin x - h) = 0$. The exact population recursion therefore stays at this point, although $|\nabla F(0)| = A$. This is a fixed point of the deterministic population recursion, not a claim that each stochastic realization stays there.

Step 3: failure of stationarity transfer for arbitrary outputs. Let $s_T = \sin\widehat{x}_T$ and $c_T = \cos\widehat{x}_T$. Both are bounded, so all expectations below exist for any distribution of $\widehat{x}_T$. Since $q_T = \mathbb{E}s_T^2$ and $\mathbb{E}c_T^2 = 1 - q_T$,

$$\begin{aligned}\mathbb{E}|\nabla F(\widehat{x}_T)|^2 &= \mathbb{E}(s_T + Ac_T)^2 \\ &= q_T + 2A\mathbb{E}[s_T c_T] + A^2(1 - q_T) \\ &= A^2 + (1 - A^2)q_T + 2A\mathbb{E}[s_T c_T].\end{aligned}$$

Cauchy–Schwarz gives $|\mathbb{E}[s_T c_T]| \le \sqrt{\mathbb{E}s_T^2\,\mathbb{E}c_T^2} = \sqrt{q_T(1 - q_T)}$. Subtracting A^2 and using the triangle inequality proves Equation (13). If $q_T \to 0$, both terms on its right-hand side vanish, proving $\mathbb{E}|\nabla F(\widehat{x}_T)|^2 \to A^2$. This argument requires neither convergence of $\widehat{x}_T$ nor a particular initialization or update rule.

Step 4: regularity and invariant state regions. The context has four possible values and there is no conditional data noise. The functions are smooth, and their remaining first and second derivatives are

$$\begin{aligned}\nabla_1 f_\xi &= \sin x + A\cos x\, w(\lambda)yu, \quad & \nabla_2 f_\xi &= A\sin x\, w(\lambda)u, \\ \nabla^2_{11} f_\xi &= \cos x - A\sin x\, w(\lambda)yu, \quad & \nabla^2_{12} f_\xi &= A\cos x\, w(\lambda)u, \quad & \nabla^2_{22} f_\xi &= 0.\end{aligned}$$

For any finite $R_y \ge 1$, these derivatives are uniformly bounded on $\mathbb{R}\times[-R_y, R_y]$ because $|w(\lambda)| \le 4$ and $|\sin x|, |\cos x| \le 1$. In the notation of Assumption 1, valid constants on this strip are

$$\mu_g = 1, \quad L_{f,0} = L_{f,1} = 1 + 4A(R_y + 1), \quad L_{g,0} = 2(R_y + 1), \quad L_{g,1} = 2, \quad L_{g,2} = 0.$$

Indeed, $\|\nabla f_\xi\| \le |\nabla_1 f_\xi| + |\nabla_2 f_\xi| \le 1 + 4AR_y + 4A$. Its Hessian is the sum of a diagonal matrix with norm at most $1 + 4AR_y$ and an off-diagonal matrix with norm at most $4A$, giving the same bound for $\|\nabla^2 f_\xi\|$. Also, $\|\nabla G_\xi\| \le 2(R_y + 1)$, while its full Hessian is constant with norm at most 2. Integrating these derivative bounds along segments in the strip proves the required Lipschitz inequalities for the functions and gradients. With no conditional data noise, we may take $\sigma_f = \sigma_{g,1} = \sigma_{g,2} = 0$; the joint-sample variances are still bounded by Appendix B, not necessarily zero. Differentiation commutes with the finite context sum. The solution maps lie in the same bounded-state regions, since

$$|y^\star(x;\xi)| = 1, \quad \bar{y}(x) = 0, \quad z^\star(x;\xi) = \frac{A\sin x\, w(\lambda)u}{\lambda}, \quad |z^\star(x;\xi)| \le 4A, \quad \bar{z}(x) = 0.$$

For completeness, these regions can contain entire trajectories of the baseline, rather than being assumed to do so. Choose $R_y \ge \max\{1, |y_0|\}$, $R_z \ge \max\{4A, |z_0|\}$, and $R_h \ge \max\{1 + 4AR_y, |h_0|\}$. Use $0 < \beta_t, \gamma_t \le 1/2$ and $0 < \theta_t \le 1$. For a fresh lower-update sample,

$$y_{t+1} = (1 - \beta_t\lambda_t)y_t + \beta_t\lambda_t u_t.$$

Both coefficients are nonnegative and sum to one, so $|y_t| \le R_y$ implies $|y_{t+1}| \le R_y$. For the independent auxiliary samples, write $\widehat{H}_t = \lambda_t^H \in [1, 2]$ and $|\widehat{b}_t| \le 4A$. Then, whenever $|z_t| \le R_z$,

$$|z_{t+1}| \le (1 - \gamma_t\lambda_t^H)R_z + 4A\gamma_t \le R_z,$$

where the last inequality uses $\lambda_t^H R_z \ge R_z \ge 4A$. Since $\widehat{A}_t = 0$, the raw outer direction is bounded by $1 + 4AR_y \le R_h$, and its convex average with h_t remains in $[-R_h, R_h]$. Induction proves all three state bounds. The same bounds hold for averages of independent samples, or exact population updates, by convexity of these intervals. No bound on x_t is needed: the derivative bounds above hold for all real x. Consequently, the update-estimate variances, including matrix–vector products involving z_t, are uniformly bounded on the specified region. This verifies every part of Assumption 1 for the stated baseline settings. For other algorithms in the direct-transfer class, their own convergence assumptions must be verified separately; Step 3 applies to their outputs whenever surrogate stationarity tends to zero. □

D Correction Modules and Auxiliary Bounds

D.1 RCSL-GA Bias and Variance

On a region containing the shared and contextual auxiliary states with $\|z\| \le R_z$, take $R_z \ge L_{f,0}/\mu_g$. This also bounds $\|z^\star(x;\xi)\|$, since $\|H_\xi^{-1}\nabla_2 f_\xi\| \le \mu_g^{-1} L_{f,0}$ at the lower solution. Assumption 1 gives

$$L_{D,y} := L_{f,1} + L_{g,2}R_z, \qquad L_{D,z} := L_{g,1}.$$

To verify these constants, hold x, ξ fixed and suppress them below. Then

$$\begin{aligned}\|D(y,z) - D(y',z')\| &\le \|\nabla_1 f_\xi(y) - \nabla_1 f_\xi(y')\| + \|A_\xi(y) - A_\xi(y')\|\,\|z\| + \|A_\xi(y')\|\,\|z - z'\| \\ &\le L_{D,y}\|y - y'\| + L_{D,z}\|z - z'\|.\end{aligned}$$

Let $e_t^{\mathrm{GA}} := \mathbb{E}[\widetilde{C}_t^{\mathrm{GA}} \mid \mathcal{F}_t] - C_t$. Independence of terminal samples from the inner solves gives $e_t^{\mathrm{GA}} = \mathbb{E}[D_\xi(x_t, y_t^c, z_t^c) - D_\xi(x_t, y^\star, z^\star) \mid \mathcal{F}_t]$, where the solution maps are evaluated at $(x_t;\xi)$. Conditional Jensen's inequality, the bound above, and $(a+b)^2 \le 2a^2 + 2b^2$ yield

$$\begin{aligned}\|e_t^{\mathrm{GA}}\|^2 \le\ & 2L_{D,y}^2\,\mathbb{E}\left[\|y_t^c(\xi) - y^\star(x_t;\xi)\|^2 \mid \mathcal{F}_t\right] \\ & + 2L_{D,z}^2\,\mathbb{E}\left[\|z_t^c(\xi) - z^\star(x_t;\xi)\|^2 \mid \mathcal{F}_t\right].\end{aligned} \tag{25}$$

The terminal evaluation batch is independent of the inner-solver samples. Let U collect the latter for one sampled context, and write $q_t(\xi,U) = D_\xi(x_t, y_t^c, z_t^c) - D_\xi(x_t, y_t, z_t)$ and $\bar{q}_t(\xi) = \mathbb{E}[q_t(\xi,U) \mid \mathcal{F}_t, \xi]$. Define the conditional variances

$$\begin{aligned}\nu_{\mathrm{eval},t}^2 &:= \mathbb{E}[\mathrm{Var}(\widehat{c}_t^{(1)} \mid \mathcal{F}_t, \xi, U) \mid \mathcal{F}_t], \\ \nu_{\mathrm{solve},t}^2 &:= \mathbb{E}[\|q_t(\xi,U) - \bar{q}_t(\xi)\|^2 \mid \mathcal{F}_t], \\ \nu_{\mathrm{ctx},t}^2 &:= \mathbb{E}[\|\bar{q}_t(\xi) - \mathbb{E}[\bar{q}_t(\xi) \mid \mathcal{F}_t]\|^2 \mid \mathcal{F}_t],\end{aligned} \tag{26}$$

where $\widehat{c}_t^{(1)}$ is a paired direction difference evaluated using one fresh conditional sample and vector variance means the expected squared Euclidean deviation from the mean. Conditional independence of the B_c context/solver draws and the law of total variance give

$$\mathrm{Var}(\widetilde{C}_t^{\mathrm{GA}} \mid \mathcal{F}_t) = \frac{1}{B_c}\left(\frac{\nu_{\mathrm{eval},t}^2}{N_e} + \nu_{\mathrm{solve},t}^2 + \nu_{\mathrm{ctx},t}^2\right). \tag{27}$$

Only terminal evaluation noise decreases as $1/N_e$. Inner-solver noise must be controlled through the inner steps and batches; it cannot be removed by enlarging the evaluation batch alone. Common random numbers couple the two terminal directions but do not guarantee variance reduction without a corresponding covariance or regularity condition.

E Convergence proofs

E.1 Proof of Theorem 3

Proof. Write $\mathcal{F}_t$ for the history before constructing v_t, and abbreviate $\mu = \mu_g$, $L = L_{g,1}$, $M = L_{f,0}$, and $\sigma = \sigma_{g,1}$. All constants below are independent of T, the inner lengths, and batch sizes. Norms are Euclidean for vectors and induced operator norms for matrices.

Inherited bounds and regularity. As shown in Appendix A, Assumption 1 implies, at every point of U and uniformly over samples,

$$\|\nabla f\| \le M, \qquad \|\nabla g\| \le L_{g,0}, \qquad \mu I \preceq \nabla_{22}^2 g \preceq LI, \qquad \|\nabla_{12}^2 g\| \le L. \tag{28}$$

These bounds follow by differentiating the sample-level Lipschitz inequalities on the interior of U and extending by continuity, together with sample-level strong convexity. The population derivatives inherit the same bounds. For $w = (x,y)$, define

$$\begin{aligned} z_\xi^\dagger(w) &:= H_\xi(w)^{-1}\nabla_2 f_\xi(w), & \psi_\xi(w) &:= \nabla_1 f_\xi(w) - A_\xi(w) z_\xi^\dagger(w), \\ L_z &:= \frac{L_{f,1}}{\mu} + \frac{L_{g,2}M}{\mu^2}, & J &:= L_{f,1} + \frac{L_{g,2}M}{\mu} + LL_z, \end{aligned}$$

and set $L_F := J(1 + L/\mu)$. The inverse identity $H^{-1} - H'^{-1} = H^{-1}(H' - H)H'^{-1}$ gives $\|z_\xi^\dagger\| \le M/\mu$ and an L_z-Lipschitz bound for $z_\xi^\dagger$ on U. Applying the product difference to $A_\xi z_\xi^\dagger$ gives a J-Lipschitz bound for ψ_ξ there.

The solution map is L/μ-Lipschitz in x. Indeed, integrate $\nabla_2 G_\xi$ along the segment between $w = (x, y^\star(x;\xi))$ and $w' = (x', y^\star(x';\xi))$, which lies in U. The two optimality conditions give $0 = A_{\mathrm{av}}^\mathsf{T}(x - x') + H_{\mathrm{av}}(y^\star(x;\xi) - y^\star(x';\xi))$, where $\|A_{\mathrm{av}}\| \le L$ and $H_{\mathrm{av}} \succeq \mu I$. Taking increments to zero in this identity gives the solution-map derivative $-H_\xi^{-1} A_\xi^\mathsf{T}$. The chain rule gives a composed gradient bounded by $M + LM/\mu$, which justifies differentiating the outer expectation. Consequently,

$$\nabla F(x) = \mathbb{E}_\xi \psi_\xi(x, y^\star(x;\xi)), \qquad \|\nabla F(x) - \nabla F(x')\| \le L_F \|x - x'\|. \tag{29}$$

Finiteness of $F(x_0)$ follows from integrability of $f_\xi(x_0, y_0)$, the bound on f_ξ's gradient, and the lower-solution distance bound below. Thus Δ is finite.

State and direction bounds. Let

$$R_y := \frac{L_{g,0}}{\mu}, \qquad Z := \max\{\|z_0\|, M/\mu\}, \qquad D := M + LZ, \qquad P := 1 + p^{-1}. \tag{30}$$

For every shared query (x_t, y_t) and every context, strong monotonicity and Equation (28) give

$$\|y_t - y^\star(x_t;\xi)\| \le \mu^{-1} \|\nabla_2 G_\xi(x_t, y_t)\| \le R_y. \tag{31}$$

This bound is a consequence of the stated regional assumptions; no trajectory-containment claim is being proved here.

Every fresh Hessian batch at a query in U has eigenvalues in $[\mu, L]$. Hence an auxiliary update satisfies, pathwise,

$$\|(I - \tau \widehat{H})z + \tau \widehat{b}\| \le q\|z\| + \tau M.$$

Since $1 - q = \tau\mu$, induction gives $\|z_t\| \le Z$ and $\|z_t^c\| \le Z$ for all shared and contextual auxiliary states. This applies to the shared updates at (x_{t+1}, y_t) exactly as written in Algorithm 1, since their batches are fresh after forming x_{t+1} and their queries lie in U. A sampled direction $\nabla_1 f - \nabla_{12}^2 g\, z$ therefore has norm at most D. Averaging preserves this bound, and a paired terminal difference has norm at most $2D$.

Introduce a potential GA correction $\widetilde{C}_t$ with Algorithm 2's law, independent of the Bernoulli trigger conditional on $\mathcal{F}_t$; it is evaluated only when triggered. Then $\|\widehat{S}_t\| \le D$, $\|\widetilde{C}_t\| \le 2D$, and

$$\mathbb{E}[\|v_t\|^2 \mid \mathcal{F}_t] \le 2D^2 + \frac{2}{p}\mathbb{E}[\|\widetilde{C}_t\|^2 \mid \mathcal{F}_t] \le 8PD^2. \tag{32}$$

The factor p^{-1}, rather than p^{-2}, follows from averaging the independent Bernoulli variable.

Conditional correction bias. Condition on $\mathcal{F}_t$ and a fresh correction context ξ, and write $y^\star = y^\star(x_t;\xi)$. The lower loop starts from $u_0 = y_t$. Integrating H_ξ between u_k and $y^\star$ shows that its population gradient step contracts by q. By Assumption 1(iv), the fresh size-N_y lower-gradient batch has centered variance at most σ^2/N_y, so

$$\mathbb{E}[\|u_{k+1} - y^\star\|^2 \mid \mathcal{F}_t, \xi, u_k] \le q^2 \|u_k - y^\star\|^2 + \frac{\tau^2\sigma^2}{N_y}.$$

Summing the geometric recursion, using Equation (31) and $1 - q^2 = \tau\mu(2 - \tau\mu) \ge \tau\mu$, yields for $u = u_{K_y}$

$$\mathbb{E}[\|u - y^\star\|^2 \mid \mathcal{F}_t, \xi] \le q^{2K_y} R_y^2 + \frac{\tau\sigma^2}{\mu N_y}. \tag{33}$$

Now also condition on the frozen lower output u, and put $H = H_\xi(x_t, u)$ and $z^\dagger = z_\xi^\dagger(x_t, u)$. For the auxiliary iterates $a_0 = z_t$, conditional unbiasedness of each fresh batch gives

$$\mathbb{E}[a_{k+1} \mid \mathcal{F}_t, \xi, u, a_k] = (I - \tau H)a_k + \tau \nabla_2 f_\xi(x_t, u).$$

Iterating this affine identity yields, for every $N_z \ge 1$,

$$\mathbb{E}[a_{K_z} \mid \mathcal{F}_t, \xi, u] = z^\dagger + (I - \tau H)^{K_z}(z_t - z^\dagger). \tag{34}$$

Since the terminal batch is independent of both solves and the direction is affine in a_{K_z}, cancellation of the shared mean and Bernoulli weighting give

$$\mathbb{E}[v_t \mid \mathcal{F}_t] = \nabla F(x_t) + e_t,$$
$$e_t = \mathbb{E}\big[\psi_\xi(x_t, u) - \psi_\xi(x_t, y^\star) - A_\xi(x_t, u)(I - \tau H)^{K_z}(z_t - z^\dagger) \mid \mathcal{F}_t\big]. \tag{35}$$

Here the expectation includes both the context and the lower-loop randomness; averaging B_c copies does not change the mean. Conditional Jensen's inequality, the J-Lipschitz bound, $\|(I-\tau H)^{K_z}\| \le q^{K_z}$, and $\|z_t - z^\dagger\| \le 2Z$ imply

$$\|e_t\|^2 \le 2J^2\left(q^{2K_y}R_y^2 + \frac{\tau\sigma^2}{\mu N_y}\right) + 8L^2Z^2q^{2K_z} =: \mathcal{B}^2. \tag{36}$$

Thus no auxiliary-noise floor is needed; the proof uses its exact conditional mean, not an assertion that its mean-square error vanishes.

Descent and the horizon schedule. By Equation (29) and Equation (35), conditional smoothness gives

$$\mathbb{E}[F(x_{t+1}) \mid \mathcal{F}_t] \le F(x_t) - \alpha\|\nabla F(x_t)\|^2 - \alpha\langle \nabla F(x_t), e_t\rangle + \frac{L_F\alpha^2}{2}\mathbb{E}[\|v_t\|^2 \mid \mathcal{F}_t]$$
$$\le F(x_t) - \frac{\alpha}{2}\|\nabla F(x_t)\|^2 + \frac{\alpha}{2}\mathcal{B}^2 + 4L_F\alpha^2PD^2.$$

These expectations are finite by the direction bound and the bounded gradient of F. Taking expectations, summing, using $F(x_T) \ge F_{\inf}$, and drawing the independent uniform output gives

$$\mathbb{E}\|\nabla F(\widehat{x}_T)\|^2 \le \frac{2\Delta}{\alpha T} + \mathcal{B}^2 + 8L_F\alpha PD^2. \tag{37}$$

For Equation (19), $q^{2K_y} = q^{2K_z} \le T^{-1/2}$ and $\sigma^2/N_y \le T^{-1/2}$, including the case $\sigma = 0$. Substitution into Equation (37) proves Theorem 3 with

$$C_p := (2\Delta + 8L_FD^2)\sqrt{1+p^{-1}} + 2J^2\left(R_y^2 + \frac{\tau}{\mu}\right) + 8L^2Z^2. \tag{38}$$

All constants are independent of T. This also proves the stated dependence on p. □

E.2 Proof of Corollary 4

Proof. The chosen T gives $C_p/\sqrt{T} \le \epsilon^2$ (also when $C_p = 0$). Equation Equation (38) gives $T = O((1+p^{-1})\epsilon^{-4})$. For fixed p and problem constants, the horizon schedule satisfies

$$K_y, K_z = O(\log(2/\epsilon)), \qquad N_y = O(1 + \sigma_{g,1}^2\epsilon^{-2}), \qquad N_z = 1.$$

Under Appendix E's counting convention, count one partial gradient or one Hessian–vector product at one sample and one query point as one primitive call. Two endpoints of a paired evaluation count separately even when they share data. Each shared iteration uses three gradient batches and two product batches, while a triggered correction at one context uses K_yN_y lower gradients, K_zN_z upper partial gradients and lower Hessian products, and $2N_e$ upper partial gradients and mixed products. Thus, over the full run,

$$\mathbb{E}N_{\nabla_{2g}} = T[m_s + pB_cK_yN_y],$$
$$\mathbb{E}(N_{\nabla_{1f}} + N_{\nabla_{2f}}) = T[2m_s + pB_c(K_zN_z + 2N_e)],$$
$$\mathbb{E}N_h = T[2m_s + pB_c(K_zN_z + 2N_e)]. \tag{39}$$

In particular, upper-loss gradients and second-order products cost $\widetilde{O}(\epsilon^{-4})$, while lower gradients cost $\widetilde{O}(\epsilon^{-4} + \sigma_{g,1}^2\epsilon^{-6})$. If $\sigma_{g,1} = 0$, the schedule has $N_y = 1$ and the total is $\widetilde{O}(\epsilon^{-4})$. These are expected upper bounds for the stated implementation and schedule, not optimality or lower-bound claims. □

F Oracle-Cost Derivations and Comparisons

F.1 Cost Model and Random-Level Work

We count a partial-gradient evaluation on one sample as one first-order call and a Hessian– or mixed-Jacobian–vector product as one second-order call. If their counts are N_g and N_h, the work model is

$$c_g(N_g + \rho N_h), \qquad \rho := c_h/c_g. \tag{40}$$

Products are evaluated by automatic differentiation without forming dense Hessians (Pearlmutter, 1994; Dagréou et al., 2024). We count distinct partial gradients separately, without derivative fusion or caching at coincident states. Shared samples couple estimators but do not remove function evaluations at different iterates. These derivative counts therefore need not equal the number of distinct data samples.

For the finite-parameter comparison, both RT-MLMC methods use a common level law and primitive batch size m. Let

$$\mathbb{P}(L = \ell) = \frac{q(1-q)^{\ell-1}}{1-(1-q)^{L_{\max}}}, \qquad \ell = 1, \dots, L_{\max}, \quad 0 < q < 1. \tag{41}$$

The relevant inner-work quantity is $\chi = \mathbb{E}[2^L]$, not $\mathbb{E}[L]$. Summing the geometric series gives, for $q \neq 1/2$,

$$\chi = \frac{2q}{1-(1-q)^{L_{\max}}} \frac{1-[2(1-q)]^{L_{\max}}}{1-2(1-q)}. \tag{42}$$

At $q = 1/2$ this becomes

$$\mathbb{P}(L = \ell) = \frac{2^{-\ell}}{1-2^{-L_{\max}}}, \qquad \chi = \frac{L_{\max}}{1-2^{-L_{\max}}}. \tag{43}$$

In particular, $L_{\max} = 10$ gives $\chi = 10240/1023 \approx 10.009775$. The published accuracy schedules need not use identical level caps; if their laws differ, use χ_{F} and χ_{H} in the corresponding cost formulas instead.

F.2 RCSL-GA: Shared Updates and Corrections

A shared step evaluates three first-order quantities, $\nabla_2 g$, $\nabla_2 f$, and $\nabla_1 f$, and two second-order products, Hz and Az. Each correction context requires K_y lower updates, K_z auxiliary updates, and a paired terminal direction evaluation. With shared batch size m_s and correction batch sizes N_y, N_z, N_e, Algorithm 2 therefore gives the expected primitive-call counts in one outer iteration:

$$\begin{aligned} \mathbb{E}N_g^{\mathrm{GA}} &= 3m_s + pB_c(K_y N_y + K_z N_z + 2N_e), \\ \mathbb{E}N_h^{\mathrm{GA}} &= 2m_s + pB_c(K_z N_z + 2N_e). \end{aligned} \tag{44}$$

Indeed, a lower update needs one gradient, an auxiliary update needs one gradient and one Hessian product, and each of the two terminal directions needs one gradient and one mixed product. The correction is activated with probability p; the shared step is performed every iteration.

Setting $m_s = N_y = N_z = N_e = m$ and $K_y = K_z = K$ yields

$$\begin{aligned} \mathbb{E}N_g^{\mathrm{GA}} &= m[3 + pB_c(2K+2)], \\ \mathbb{E}N_h^{\mathrm{GA}} &= m[2 + pB_c(K+2)], \\ \Gamma_{\mathrm{GA}} &= 3 + 2\rho + pD(\rho), \qquad D(\rho) = B_c[2K + 2 + \rho(K+2)]. \end{aligned} \tag{45}$$

The importance factor $1/p$ rescales the correction but adds no derivative calls. It affects estimator variance rather than this work count.

The equal-batch specialization is used only for the fixed-parameter per-iteration comparison. Under the accuracy schedule in the convergence analysis, N_y generally grows with the target accuracy while $N_z = 1$. Total-work bounds must therefore use the general counts in Equation (44).

F.3 FFO-RT-MLMC: First-Order Penalty Updates

The penalty in Yang et al. (2025) has the form $\mathcal{L}(x, z, y, \lambda) = f(x, z) + \lambda[g(x, z) - g(x, y)]$, with data arguments suppressed. Here z is a penalized lower variable, not the linear-system variable in RCSL-GA. Each inner iteration of their Algorithm 2 updates one y trajectory and two penalized z trajectories. The former uses one gradient; each latter uses $\nabla_2\mathcal{L} = \nabla_2 f + \lambda\nabla_2 g$, hence two gradients. The displayed loop executes 2^L iterations, giving $5 \cdot 2^L$ calls.

Each of the three terminal RT-MLMC estimates uses $\nabla_1\mathcal{L} = \nabla_1 f + \lambda[\nabla_1 g(x, z) - \nabla_1 g(x, y)]$, which costs three first-order calls. Thus, under direct evaluation of the three estimates,

$$N_g^{\mathrm{F}}(L) = m(5 \cdot 2^L + 9), \qquad N_h^{\mathrm{F}}(L) = 0, \qquad \Gamma_{\mathrm{F}} = 9 + 5\chi. \tag{46}$$

This count follows the full inner loop as written, including both penalized updates; early stopping or derivative reuse would change its constants.

F.4 Hessian-RT-MLMC: Epoch and Neumann Work

In Algorithms 1 and 3 of Hu et al. (2023), epoch k performs 2^k lower-gradient updates. A sampled level L therefore costs

$$W_L = \sum_{k=1}^{L} 2^k = 2^{L+1} - 2. \tag{47}$$

One trajectory supplies the initial, coarse, and fine states; it is not necessary to solve three independent lower problems. Each of the three terminal directions uses $\nabla_1 f$, $\nabla_2 f$, one mixed product, and $\widehat{N}$ Hessian products, where

$$\widehat{N} \sim \mathrm{Unif}\{0, \dots, N-1\}, \qquad \mathbb{E}(\widehat{N} + 1) = \frac{N+1}{2}. \tag{48}$$

Consequently,

$$\begin{aligned} N_g^{\mathrm{H}}(L) &= m(W_L + 6) = m(2^{L+1} + 4), \\ \mathbb{E}N_h^{\mathrm{H}} &= 3m\,\mathbb{E}(\widehat{N} + 1) = \tfrac{3}{2}m(N+1), \\ \Gamma_{\mathrm{H}} &= 2\chi + 4 + \tfrac{3}{2}\rho(N+1). \end{aligned} \tag{49}$$

We count all three directions without cancellations at $L = 1$ or caching identical derivatives. Here $N - 1$, rather than N, is the largest sampled Neumann length. This distinction accounts for the factor $(N+1)/2$. The second line is an expectation: the three terminal directions may use different sampled Neumann lengths, each with the stated marginal law. Their expected costs add regardless of the coupling between these lengths.

F.5 Per-Iteration Cost Thresholds

The following result compares the fixed-parameter costs derived above. The inner lengths, batch sizes, and RT-MLMC parameters are held fixed when varying the correction probability p.

Theorem 5 (Per-iteration oracle-cost advantage)**.** *Under the counting convention in Appendix F.1, let $\rho > 0$, $B_c, K \geq 1$, and $p \in (0, 1]$. Use the common primitive batch size m and the common level-work quantity χ in the normalized comparison, and define*

$$\begin{aligned} p^\star_{\mathrm{GA,F}} &:= \frac{6 + 5\chi - 2\rho}{D(\rho)}, \\ p^\star_{\mathrm{GA,H}} &:= \frac{2\chi + 1 + \rho[\frac{3}{2}(N+1) - 2]}{D(\rho)}, \\ D(\rho) &:= B_c[2K + 2 + \rho(K+2)]. \end{aligned} \tag{50}$$

RCSL-GA has strictly lower expected per-iteration oracle cost than both baselines if and only if

$$p < \min\{p^\star_{\mathrm{GA,F}}, p^\star_{\mathrm{GA,H}}\}.$$

A feasible $p \in (0, 1]$ exists if and only if this minimum is positive.

Proof. The positive factor mc_g cancels in a single-iteration comparison. Since $D(\rho) > 0$, the cost differences satisfy

$$\begin{aligned}\Gamma_{\mathrm{F}} - \Gamma_{\mathrm{GA}} &= 6 + 5\chi - 2\rho - pD(\rho) = D(\rho)(p^\star_{\mathrm{GA,F}} - p),\\ \Gamma_{\mathrm{H}} - \Gamma_{\mathrm{GA}} &= 2\chi + 1 + \rho[\tfrac{3}{2}(N+1) - 2] - pD(\rho) = D(\rho)(p^\star_{\mathrm{GA,H}} - p).\end{aligned} \tag{51}$$

Both differences are positive exactly under the stated condition. Equality at either threshold gives equal cost with that baseline. If the minimum threshold is nonpositive, no positive p satisfies both strict inequalities; if it is positive, a sufficiently small $p \in (0, 1]$ satisfies both. □

For unequal batches, write the expected GA cost as $c_g(a + pb)$, where

$$\begin{aligned}a &= m_s(3 + 2\rho),\\ b &= B_c[K_yN_y + K_zN_z + 2N_e + \rho(K_zN_z + 2N_e)].\end{aligned} \tag{52}$$

If baseline $j \in \{\mathrm{F}, \mathrm{H}\}$ has cost $c_g m_j \Gamma_j$, then the corresponding strict threshold is $p < (m_j\Gamma_j - a)/b$. The coefficient Γ_j uses that baseline's own level law and, where applicable, Neumann cap. All parameters other than p are held fixed in this comparison. These thresholds compare expected work per iteration; they do not compare the accuracy attained or total work required by the methods.

F.6 Total Work under the Accuracy Schedule

We use the stationarity criterion $\mathbb{E}\|\nabla F(\widehat{x})\|^2 \le \epsilon^2$. The symbol $\sigma^2_{g,1}$ denotes the conditional gradient-noise variance bound in Assumption 1(iv). The per-iteration counts in Appendix F.2 must be combined with the inner accuracy schedule when deriving total oracle complexity.

Exact total-call accounting. For a deterministic horizon T and parameters fixed within a run, let N_g^{tot} and N_h^{tot} denote the total numbers of first- and second-order primitive calls. Summing Equation (44) gives

$$\begin{aligned}\mathbb{E}N_g^{\mathrm{tot}} &= T\big[3m_s + pB_c(K_yN_y + K_zN_z + 2N_e)\big],\\ \mathbb{E}N_h^{\mathrm{tot}} &= T\big[2m_s + pB_c(K_zN_z + 2N_e)\big].\end{aligned} \tag{53}$$

Consequently, the expected weighted derivative work is

$$\begin{aligned}\mathcal{W}_{\mathrm{GA}} &:= c_g\mathbb{E}N_g^{\mathrm{tot}} + c_h\mathbb{E}N_h^{\mathrm{tot}}\\ &= c_gT\big\{m_s(3 + 2\rho) + pB_c[K_yN_y + (1 + \rho)(K_zN_z + 2N_e)]\big\}.\end{aligned} \tag{54}$$

This identity allows unequal inner batches and does not assume that refinement cost is independent of the target accuracy.

Consequences of the convergence schedule. For the raw-update method, the convergence analysis gives

$$T = \mathcal{O}\big((1 + p^{-1})\epsilon^{-4}\big).$$

For fixed $p > 0$, m_s, B_c, and N_e, with problem constants and initialization fixed, the corresponding inner schedule satisfies

$$K_y, K_z = \mathcal{O}(\log(2/\epsilon)), \qquad N_y = \mathcal{O}(1 + \sigma^2_{g,1}\epsilon^{-2}), \qquad N_z = 1. \tag{55}$$

Substituting into Equation (53) yields

$$\begin{aligned}\mathbb{E}N_g^{\mathrm{tot}} &= \widetilde{\mathcal{O}}\big(\epsilon^{-4} + \sigma^2_{g,1}\epsilon^{-6}\big),\\ \mathbb{E}N_h^{\mathrm{tot}} &= \widetilde{\mathcal{O}}(\epsilon^{-4}).\end{aligned} \tag{56}$$

Equivalently,

$$\mathcal{W}_{\mathrm{GA}}(\epsilon) = \widetilde{\mathcal{O}}\big((c_g + c_h)\epsilon^{-4} + c_g\sigma^2_{g,1}\epsilon^{-6}\big)\,. \tag{57}$$

The ϵ^{-6} term comes from conditional lower-gradient sampling. Upper-objective gradients and second-order products retain $\widetilde{\mathcal{O}}(\epsilon^{-4})$ bounds. Thus the total primitive-call bound is $\widetilde{\mathcal{O}}(\epsilon^{-6})$ for fixed nonzero conditional lower-gradient noise. If $\sigma_{g,1} = 0$, the schedule has $N_y = 1$ and the total bound improves to $\widetilde{\mathcal{O}}(\epsilon^{-4})$. This conditional noiselessness concerns sampling given a context; context sampling may remain stochastic.

Geometric contraction in the inner iteration count alone does not yield a polylogarithmic correction cost under noisy conditional sampling. The work required to control the lower-solve noise floor must also be counted. Fixed stochastic inner lengths and batches generally do not provide an arbitrary-accuracy guarantee.

Comparison with the baseline guarantees. Under their respective assumptions and accuracy schedules, FFO-RT-MLMC and Hessian-RT-MLMC have published total derivative-call bounds of $\widetilde{\mathcal{O}}(\epsilon^{-6})$ and $\widetilde{\mathcal{O}}(\epsilon^{-4})$, respectively (Yang et al., 2025; Hu et al., 2023). Our general-noise total-call upper bound has the same ϵ exponent as the former and a larger exponent than the latter. The fixed-parameter thresholds in Appendix F.5 therefore do not establish an improved asymptotic total-call bound for RCSL-GA. These upper bounds also do not determine an ordering of actual runtimes: their constants, logarithmic factors, oracle requirements, and assumptions differ.

Total-work comparisons and correction frequency. For a prescribed deterministic budget $T_j(\epsilon)$, let $c_{j,t}$ denote method j's derivative work at iteration t. The expected total work is

$$\mathcal{W}_j(\epsilon) = \sum_{t=0}^{T_j(\epsilon)-1} \mathbb{E}[c_{j,t}]. \tag{58}$$

When the settings are fixed within a run, write $\bar{c}_j(\epsilon)$ for its expected per-iteration work, so $\mathcal{W}_j(\epsilon) = T_j(\epsilon)\bar{c}_j(\epsilon)$. For positive budgets and costs, comparison with $j \in \{\mathrm{F}, \mathrm{H}\}$ then gives

$$\mathcal{W}_{\mathrm{GA}}(\epsilon, p) < \mathcal{W}_j(\epsilon) \quad \Longleftrightarrow \quad \frac{T_{\mathrm{GA}}(\epsilon, p)}{T_j(\epsilon)} < \frac{\bar{c}_j(\epsilon)}{\bar{c}_{\mathrm{GA}}(\epsilon, p)}. \tag{59}$$

An accuracy-based comparison requires budgets justified for the same stationarity criterion. Reducing p lowers the correction component of a fixed-parameter iteration, but also increases the variance contribution in Equation (16) and the iteration bound. The horizon-dependent inner lengths and batches may then change as well. In particular, the fixed-p rates above should not be used unchanged for probabilities that vanish with ϵ. The per-iteration thresholds do not by themselves select a probability minimizing total work; the experiments assess wall-clock behavior at the finite settings used in each benchmark.

F.7 Numerical Analysis for the Illustrative Parameters

Take

$$m = 100, \quad B_c = 5, \quad K = 10, \quad p = 0.1, \quad N = 12, \quad q = 1/2, \quad L_{\max} = 10. \tag{60}$$

We examine $\rho \in [2, 4]$ as a sensitivity range, not a universal HVP timing law: automatic-differentiation benchmarks show that the ratio depends on the architecture, implementation, and differentiation mode (Dagréou et al., 2024). These parameters illustrate the cost model and are not asserted to be the experimental settings or a common-accuracy schedule for all three methods.

Using $\chi = 10240/1023$ without intermediate rounding gives

$$\begin{aligned} \Gamma_{\mathrm{GA}} &= 14 + 8\rho, \\ \Gamma_{\mathrm{F}} &= \frac{60407}{1023} \approx 59.048876, \\ \Gamma_{\mathrm{H}} &= \frac{24572}{1023} + 19.5\rho \approx 24.019550 + 19.5\rho, \\ D(\rho) &= 110 + 60\rho. \end{aligned} \tag{61}$$

Table 2 lists the cost coefficients and thresholds. All costs are normalized by mc_g; actual per-iteration costs multiply the entries by $100c_g$.

The comparison holds throughout the interval, not just at its endpoints. Set $u = 6 + 5\chi$ and $v = 2\chi + 1$. Then

$$\begin{aligned} p^\star_{\mathrm{GA,F}} &= \frac{u - 2\rho}{110 + 60\rho}, & \frac{dp^\star_{\mathrm{GA,F}}}{d\rho} &= \frac{-220 - 60u}{(110 + 60\rho)^2} < 0, \\ p^\star_{\mathrm{GA,H}} &= \frac{v + 17.5\rho}{110 + 60\rho}, & \frac{dp^\star_{\mathrm{GA,H}}}{d\rho} &= \frac{1925 - 60v}{(110 + 60\rho)^2} > 0. \end{aligned} \tag{62}$$

Table 2: Per-iteration coefficients and probability thresholds for $B_c = 5$, $K = 10$, $N = 12$, and $L_{\max} = 10$. GA costs use $p = 0.1$.

ρ	Γ_{GA}	Γ_{F}	Γ_{H}	$p^\star_{\mathrm{GA,F}}$	$p^\star_{\mathrm{GA,H}}$
2	30	59.0489	63.0196	0.2263	0.2436
3	38	59.0489	82.5196	0.1726	0.2535
4	46	59.0489	102.0196	0.1373	0.2601

Thus the uniform threshold is

$$\min_{\rho\in[2,4]} \min\{p^\star_{\mathrm{GA,F}}, p^\star_{\mathrm{GA,H}}\} = \min\{p^\star_{\mathrm{GA,F}}(4), p^\star_{\mathrm{GA,H}}(2)\} = \frac{49154}{358050} \approx 0.1372825. \tag{63}$$

In particular, $p = 0.1$ gives strictly lower expected per-iteration costs throughout $[2, 4]$ for these illustrative parameters. This calculation does not compare achieved stationarity, total derivative work, or total runtime. A total-work comparison must additionally account for the iteration budgets and accuracy-dependent parameters, as in Appendix F.6 and Equation (59).

G Experimental Details

At each evaluation point, context-specific lower solutions are recomputed with the evaluation procedure specified for the corresponding experiment. All methods are implemented in Python 3.12 with PyTorch 2.8 and evaluated on the same CPU-only environment with 32 allocated cores on an AMD EPYC 9654 and 60 GB of memory. Stepsizes and method-specific parameters are tuned separately, while data partitions and evaluation routines are shared. We report performance against both outer iterations and wall-clock time.

G.1 Structural-Bias Example

Data and evaluation. The experiment uses the synthetic distribution in Section 3.3: u is uniform on $\{-1, 1\}$, λ is uniform on $\{1, 2\}$, and the two variables are independent. There is no conditional data variable η, so all stochasticity comes from sampling contexts. We set $A = 4$ and evaluate the exact population quantities x_t, $|\nabla F(x_t)|^2$, and $F(x_t) - F(x^\star_F)$ using the closed-form expressions in Section 3.3. The curves summarize 20 independent runs by their mean and standard deviation.

Implementation details. Both methods start from $x_0 = 0.8$ and run for 1000 outer iterations. They use outer stepsize $\alpha = 0.005$, shared-state stepsize $\beta = 0.1$, and direction-averaging weight $\theta = 0.05$. Thus the plotted RCSL-GA implementation uses the averaged-direction variant described after Algorithm 1. RCSL-GA uses $p = 1$ to isolate the effect of contextual correction, with $B_c = 4$, $K_y = 20$, and contextual stepsize 0.4. Since $\nabla^2_{12} G_\xi = 0$ in this example, the outer direction does not depend on the auxiliary state or its inner solve length.

G.2 Meta-Learning

G.2.1 Detailed Formulation

We follow the regularization-based contextual meta-learning formulation of Hu et al. (2023). Let M be the number of tasks, let μ be uniform on $[M]$, and let $D = (a, b)$ denote a feature–label pair. For task i, the feature is $a \in \mathbb{R}^d$, the one-hot label is $\boldsymbol{b} \in \{0, 1\}^C$, and the task classifier is $W_i \in \mathbb{R}^{d\times C}$. Writing the shared meta-parameter as $X \in \mathbb{R}^{d\times C}$, the detailed form of Equation (20) is

$$\begin{aligned} \min_{X\in\mathbb{R}^{d\times C}} \quad & F_{\mathrm{meta}}(X) := \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{(a,b)\sim\mathcal{P}_i^{\mathrm{val}}}\left[\ell\big(W_i^\star(X); (a, b)\big)\right], \\ \text{where} \quad & W_i^\star(X) := \operatorname*{argmin}_{W\in\mathbb{R}^{d\times C}} \left\{ \mathbb{E}_{(a,b)\sim\mathcal{P}_i^{\mathrm{tr}}}\left[\ell\big(W; (a, b)\big)\right] + \frac{\lambda}{2}\|W - X\|_F^2 \right\}. \end{aligned} \tag{64}$$

The loss is multiclass logistic regression,

$$\ell(W; (a, b)) = -\boldsymbol{b}^\top W^\top a + \log\left(\mathbf{1}^\top \exp(W^\top a)\right). \tag{65}$$

Here the exponential acts componentwise. In the implementation, $x = \mathrm{vec}(X)$ and $y_i = \mathrm{vec}(W_i)$, so Equation (64) has exactly the CSBO form in Equation (20). The shared center X transfers information across tasks, and the lower problem adapts it to task i using that task's training split. The validation split supplies the upper training objective and is also used for reporting. Thus the reported metric measures optimization of the fitted upper objective on the trained tasks, rather than performance on held-out tasks or a separate final test split. We set $\lambda = 2$, which makes every lower problem strongly convex in the classifier parameters.

G.2.2 Dataset and Preprocessing

Tiny-ImageNet contains 100,000 training images from 200 classes, with 500 images per class (mnmoustafa & Ali, 2017). Each image is resized to 224×224, normalized with the standard ImageNet channel statistics, and passed through an ImageNet-pretrained ResNet-18 (He et al., 2016; Deng et al., 2009). Removing the final classification layer produces a $d = 512$ dimensional feature vector. The feature extractor is fixed; only the task classifiers and their shared regularization center are optimized.

These 200 classes are divided into $M = 10$ disjoint tasks, with each task containing $C = 20$ categories[1]. Within every class, a seeded random permutation assigns 450 examples to task adaptation and 50 examples to validation. Consequently, each task contains 9,000 training examples and 1,000 validation examples. Labels are converted to task-local 20-dimensional one-hot vectors, and the task distribution μ is uniform. The same task construction and 90/10 split are reused by every algorithm. At each reporting point, we recompute all task-specific lower solutions with a common task-adaptation routine and evaluate the objective on the complete validation split.

G.2.3 Implementation and Hyperparameters

Dataset construction and feature extraction are implemented in `utils/ImageNet.py`, the experiment driver and evaluation protocol in `run.py`, and all optimizers in `src/algos.py`. Every stochastic upper- and lower-level oracle call uses a minibatch of 100 examples. RCSL-GA, FFO-RT-MLMC, and Hessian-RT-MLMC run for 30,000 outer iterations and are evaluated every 300 iterations; FFO runs for 3,000 iterations and is evaluated every 30 iterations because of its growing inner-loop cost. Evaluation time is excluded from the reported optimization time. We run ten independent seeds and plot the mean together with one standard deviation.

Table 3 summarizes the settings that differ across methods. RCSL-GA uses correction probability $p = 0.1$ in the main comparison, five independently sampled correction contexts, and ten inner steps for each contextual lower and auxiliary solve. Baseline parameters not included in the grid search follow the cited supplemental implementations (Yang et al., 2025; Hu et al., 2023); the tuned parameters are listed in Table 4.

Table 3: Meta-learning settings. Other baseline parameters follow the cited supplemental implementations.

Method	Iterations	Eval. gap	Method-specific settings
RCSL-GA	30,000	300	$\alpha = 0.12$, $\beta = 0.05$, $\lambda_y = \lambda_z = 0.05$, $p = 0.1$, $B_c = 5$, and $K_y = K_z = 10$
FFO	3,000	30	$\alpha_0 = 25$, inner scale $\beta = 500$ and other settings from (Yang et al., 2025)
FFO-RT-MLMC	30,000	300	outer stepsize $\alpha_0 \times \epsilon^4 = 4 \times 10^{-1}$, inner scale $\beta = 500$ and other settings from (Yang et al., 2025)
Hessian-RT-MLMC	30,000	300	base outer stepsize 0.5, inner scale $\beta = 70$ and other settings from (Hu et al., 2023)

The parameters in Table 4 are selected by grid search on on three different seed and the same task split of the main experiment, using the final validation objective diagnostic. The table lists the candidate grids and selected values.

[1] Yang et al. (2025) used 21 classes in three disjoint 7-way tasks. Our ten 20-way tasks cover all 200 classes, providing a larger finite-context benchmark with broader class coverage.

Table 4: Meta-learning hyperparameter grids and selected values.

Method	Parameter	Candidate values	Selected value
RCSL-GA	α	$\{0.5, 0.3, 0.12, 0.1, 0.05, 0.01\}$	0.12
RCSL-GA	β	$\{0.5, 0.3, 0.12, 0.1, 0.05, 0.01\}$	0.05
RCSL-GA	λ_z	$\{0.5, 0.3, 0.12, 0.1, 0.05, 0.01\}$	0.05
RCSL-GA	λ_y	$\{0.5, 0.3, 0.12, 0.1, 0.05, 0.01\}$	0.05
RCSL-GA	p	$\{0.5, 0.2, 0.1, 0.05, 0.01\}$	0.1
FFO	α	$\{50, 25, 0.5, 0.1, 0.01\}$	25
FFO	β	$\{500, 250, 100, 50, 5, 0.5\}$	500
FFO-RT-MLMC	$\alpha_0 * \epsilon^4$	$\{1, 0.4, 0.2, 0.1, 0.05, 0.01\}$	0.4
FFO-RT-MLMC	β	$\{500, 250, 100, 50, 5, 0.5\}$	500
Hessian-RT-MLMC	α	$\{1, 0.5, 0.1, 0.05, 0.01\}$	0.5
Hessian-RT-MLMC	β	$\{120, 70, 12, 8, 0.5, 0.1\}$	70

G.2.4 Final Quantitative Results

Table 5 complements Figure 2 with metrics at each method's final reporting point. All metrics use the common evaluation routine and the averaged outer iterate $\bar{x}$ available at that point. The validation objective and classification error are averaged over tasks using the complete validation split.

We additionally report the task-gradient diagnostic

$$\widehat{\mathcal{S}}_{\mathrm{task}}(\bar{x}) := \frac{1}{M}\sum_{i=1}^{M}\left\|\widehat{h}_i(\bar{x})\right\|^2, \tag{66}$$

where $\widehat{h}_i(\bar{x})$ is the task-specific hypergradient estimate returned by the evaluation routine. This averages the squared task-wise norms, whereas stationarity of F_{meta} depends on the squared norm of the averaged hypergradient. We therefore treat $\widehat{\mathcal{S}}_{\mathrm{task}}$ as an auxiliary diagnostic, not a stationarity residual for F_{meta}.

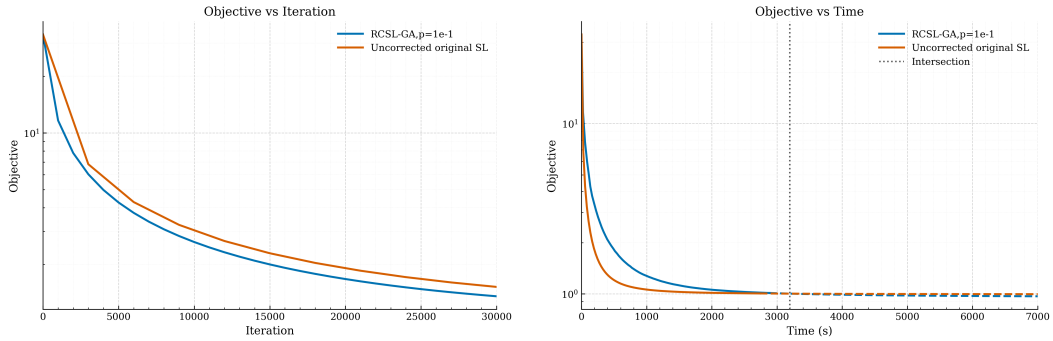
Table 5: Final Tiny-ImageNet metrics at the stated iteration budgets (mean $\pm$ one standard deviation over ten seeds). Bold marks the lowest mean in each metric column; $\widehat{\mathcal{S}}_{\mathrm{task}}$ is defined in Equation (66).

Method	Iterations	Validation objective	Validation error (%)	$\widehat{\mathcal{S}}_{\mathrm{task}}$
RCSL-GA ($p = 0.01$)	30,000	1.795312 ± 0.065299	41.097 ± 0.500	0.536178 ± 0.040364
RCSL-GA ($p = 0.1$)	30,000	1.355561 ± 0.001911	38.152 ± 0.068	0.479431 ± 0.000559
RCSL-GA ($p = 1$)	30,000	$\mathbf{1.351993} \pm 0.000488$	$\mathbf{38.066} \pm 0.055$	0.481546 ± 0.000085
FFO	3,000	2.445720 ± 0.298249	54.353 ± 0.784	0.520846 ± 0.309939
FFO-RT-MLMC	30,000	1.955320 ± 0.020845	52.398 ± 0.299	$\mathbf{0.326537} \pm 0.003377$
Hessian-RT-MLMC	30,000	16.651341 ± 0.759415	88.254 ± 0.244	2.200504 ± 0.052576

These endpoints reflect the stated iteration budgets; per-iteration costs differ across methods and correction probabilities. With these budgets, RCSL-GA at $p = 1$ attains the lowest mean validation objective and classification error. The default $p = 0.1$ is close on both metrics, with a classification-error gap of only 0.086 percentage points; reducing p to 0.01 worsens both metrics. FFO-RT-MLMC has the lowest task-gradient diagnostic, whose ranking differs from that of the validation metrics. Wall-clock comparisons are given in Figure 2, and the correction-cost trade-off is examined in Appendix G.2.6.

G.2.5 Ablation of the Contextual Correction

We isolate contextual correction by comparing RCSL-GA with $p = 0.1$ against its uncorrected SL counterpart. The task partitions, initialization, shared-update settings, evaluation protocol, and ten random seeds are kept fixed; only contextual correction is enabled or disabled.



(a) Objective versus outer iterations (b) Objective versus wall-clock time

Figure 4: Contextual-correction ablation on Tiny-ImageNet: RCSL-GA ($p = 0.1$) versus uncorrected SL. Curves show means over ten seeds; the vertical dotted line marks the crossover in panel (b).

Figure 4(a) shows that RCSL-GA achieves lower validation objectives within the displayed iteration range. In wall-clock time, however, uncorrected SL initially progresses faster because its iterations are cheaper. The plotted curves intersect at approximately $3{,}196.2$ s, corresponding to $111{,}107$ outer iterations for RCSL-GA and $349{,}576$ for uncorrected SL. Beyond this point, the RCSL-GA curve remains lower. These results illustrate a budget-dependent trade-off: contextual correction improves progress per iteration, while its computational overhead delays the wall-clock advantage.

G.2.6 Sensitivity to the Correction Probability

Figure 5 shows a clear trade-off between the correction frequency and computational cost. Most of the improvement in the final objective is already achieved at $p = 0.1$. Increasing p from 0.1 to 1 decreases the average final objective only from 1.35556 to 1.35199 (about 0.25%), while increasing the average runtime from 684.9 s to 6595.3 s, or approximately $9.63\times$. In particular, the final objectives at $p = 0.5$ and $p = 1$ are virtually indistinguishable, despite the additional runtime incurred by $p = 1$. This indicates that correction at every iteration is unnecessary, and that relatively infrequent correction is sufficient to preserve the optimization benefit while avoiding substantial computational overhead.

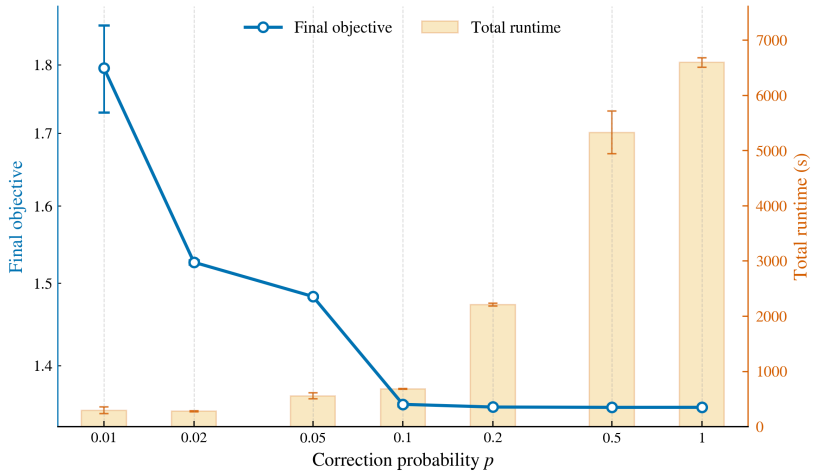


Figure 5: Correction-probability sensitivity in meta-learning. Final objective (blue, left axis) and total optimization time (bars, right axis) are shown against logarithmic p.

G.3 Task-Aggregated Power Scheduling

G.3.1 From ETO to CSBO

Estimate-then-optimize first estimates an uncertain quantity and then inserts that estimate into a downstream optimization problem. In a conditional formulation, features a parameterize a predictive distribution $Q_x(\cdot \mid a)$ and induce the decision

$$z_x^\star(a) \in \operatorname*{argmin}_{z \in \mathcal{Z}} \mathbb{E}_{\widetilde{d} \sim Q_x(\cdot|a)}[C(z, \widetilde{d})]. \tag{67}$$

Classical ETO estimates x using a prediction loss before solving Equation (67). Decision-focused or integrated training instead chooses x using the realized downstream loss $\mathbb{E}_{(a,d)\sim\mathcal{P}}[C(z_x^\star(a), d)]$ (Donti et al., 2017; Elmachtoub & Grigas, 2022; Elmachtoub et al., 2023; Mandi et al., 2024). This is a bilevel problem, and it becomes CSBO by taking the observed feature (or a task–feature pair) as the context, the demand as the conditional sample, and $z_x^\star(a)$ as the contextual lower response.

Our benchmark uses a task-aggregated version suited to the temporal task construction. The context is a rolling task $\xi = i$, a conditional observation is $\eta = D = (a, d) \sim \mathcal{P}_i^{\mathrm{tr}}$, and the deterministic predictor $r_x(a)$ supplies a collection of load scenarios. The lower problem optimizes one representative schedule $s_i^\star(x)$ over those scenarios, rather than a different schedule for each feature vector. Consequently, Equation (21) has the CSBO identification

$$\begin{aligned} \xi = i, \qquad \eta = (a, d), \qquad y = s, \\ f(x, s; D, i) = c_\tau(s, d), \\ g(x, s; D, i) = c_\tau(s, r_x(a)) + \frac{\gamma}{2}\|s - r_x(a)\|^2, \end{aligned} \tag{68}$$

together with the common feasible set $\mathcal{Z}$. For fixed x, the lower problem is ETO over the predicted task scenarios; optimizing x through the realized upper cost embeds that ETO map in CSBO. This task aggregation defines the surrogate benchmark studied here. In particular, the experiment does not evaluate the feature-conditioned decision rule $z_x^\star(a)$ in Equation (67): at each reporting point, the representative schedule for a task is held fixed across its evaluation days.

G.3.2 Detailed Formulation and Scope

For a schedule $s \in \mathbb{R}^{24}$ and a load vector $d \in \mathbb{R}^{24}$, define the original scheduling loss and ramp-feasible set by

$$c(s, d) := \sum_{j=1}^{24} \left(\gamma_{\mathrm{under}}[d_j - s_j]_+ + \gamma_{\mathrm{over}}[s_j - d_j]_+\right), \qquad [u]_+ := \max\{u, 0\}, \tag{69}$$

$$\mathcal{Z} := \left\{s \in \mathbb{R}^{24} : |s_j - s_{j-1}| \le c_{\mathrm{ramp}},\ j = 2, \dots, 24\right\}. \tag{70}$$

where γ_{under} and γ_{over} penalize under- and over-generation. The affine predictor is

$$r_x(a) = W^\top a + b, \qquad W \in \mathbb{R}^{125\times 24}, \quad b \in \mathbb{R}^{24}, \tag{71}$$

with $x = (\mathrm{vec}(W), b)$. Because $[u]_+$ is nondifferentiable at zero, training uses $\phi_\tau(u) = \tau \log(1 + \exp(u/\tau))$ and the smooth cost

$$c_\tau(s, q) = \sum_{j=1}^{24} \left(\gamma_{\mathrm{under}}\phi_\tau(q_j - s_j) + \gamma_{\mathrm{over}}\phi_\tau(s_j - q_j)\right). \tag{72}$$

Using independent notation $D^u = (a^u, d^u)$ and $D^\ell = (a^\ell, d^\ell)$ for upper and lower oracle draws, the population problem is

$$\begin{aligned} \min_x \quad & F_{\mathrm{power},\tau}(x) := \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{D^u \sim \mathcal{P}_i^{\mathrm{tr}}}\left[c_\tau(s_i^\star(x), d^u)\right], \\ \text{where} \quad & s_i^\star(x) := \operatorname{argmin}_{s\in\mathbb{R}^{24}} \left\{\mathbb{E}_{D^\ell \sim \mathcal{P}_i^{\mathrm{tr}}}\left[c_\tau(s, r_x(a^\ell)) + \frac{\gamma}{2}\|s - r_x(a^\ell)\|^2\right] + \iota_{\mathcal{Z}}(s)\right\}. \end{aligned} \tag{73}$$

The quadratic term makes the smooth part of every contextual lower objective strongly convex. The upper level evaluates the decision against realized loads, whereas the lower level only uses loads predicted by r_x; the true lower-sample label d^ℓ is therefore not used by g.

The ramp-constrained lower problem lies outside the smooth unconstrained setting in Assumption 1. Training smooths the positive-part terms with ϕ_τ, but changes in the active ramp constraints can still make the lower solution map nondifferentiable. Sensitivity through constrained solutions can be analyzed using their optimality conditions under suitable regularity assumptions (Amos & Kolter, 2017; Agrawal et al., 2019). Here we use the projected implementations described below. Projection enforces ramp feasibility up to numerical accuracy, but does not by itself justify applying the unconstrained hypergradient analysis. Accordingly, this experiment provides an empirical comparison of decision loss and optimization time on a task-aggregated surrogate benchmark motivated by the scheduling application of Donti et al. (2017).

We use $\gamma_{\text{under}} = 50, \gamma_{\text{over}} = 0.5, c_{\text{ramp}} = 0.4, \gamma = 1, \tau = 0.05$. Evaluation uses the original piecewise-linear cost c.

G.3.3 Dataset and Preprocessing

We use the PJM load-and-temperature data released with the generator scheduling experiment of Donti et al. (2017). The observations span January 1, 2008 to September 28, 2016. Duplicate timestamps are removed and missing hourly values within a day are filled first backward and then forward within that day. Each target is the next-day 24-dimensional load vector. Its 125-dimensional feature vector concatenates the previous day's load (24), previous-day temperature (24), next-day temperature forecast (24), hot- and cold-temperature hinge features ($24 + 24$), four calendar and seasonal indicators, and one constant feature.

The chronological rolling tasks are

$$\begin{aligned} &2008\text{–}2012 \longrightarrow 2013, && 2009\text{–}2013 \longrightarrow 2014, \\ &2010\text{–}2014 \longrightarrow 2015, && 2011\text{–}2015 \longrightarrow 2016. \end{aligned} \tag{74}$$

For every task, feature means and standard deviations are computed only from its five-year training window and then applied to both its training and following-year evaluation splits; the constant feature is left unchanged. For task i, both upper and lower optimization samples are drawn from $\mathcal{P}_i^{\text{tr}}$, and the corresponding following-year split is used to report the decision loss of that task's representative schedule. This separation is specific to task i.

All four tasks participate in training and jointly update the shared predictor parameter x; no task is held out. Their temporal windows overlap. In particular, some observations from the 2013–2015 evaluation periods of earlier tasks also enter the training windows of later tasks and can therefore influence x through those later tasks. Consequently, the aggregate four-task results should not be interpreted as an evaluation under a globally chronological holdout for future prediction. They provide an algorithmic comparison under the stated overlapping-window protocol. The final evaluation period ends on September 28, 2016.

The reported decision metric averages the original cost over individual evaluation days within each task and then equally over the four tasks. It does not evaluate the cost at a batch-mean load.

G.3.4 Implementation, Evaluation, and Hyperparameters

Dataset construction and the experiment driver are implemented in `run.py`; the losses, training wrappers, and evaluation routine are in `src/algos.py`; and the optimizer states and ramp projection are in `src/csbo_optimizer.py`. For a minibatch $\{(a_b, d_b)\}_{b=1}^B$, the code averages the samplewise losses $c_\tau(s, d_b)$ and $c_\tau(s, r_x(a_b))$ rather than evaluating the loss at a batch-mean load. Both the scheduling cost and squared regularizer are divided by 24, a common positive scaling that preserves the lower minimizer and makes the reported decision loss a per-hour average.

Let $G_{i,\tau}(x, s)$ denote the smooth lower expected cost including the quadratic regularizer and the factor $1/24$ used in the implementation. Every lower-state update used by the four compared methods is followed by

$$s^{k+1} = \Pi_{\mathcal{Z}}\big(s^k - \lambda_k \widehat{\nabla}_2 G_{i,\tau}(x, s^k)\big). \tag{75}$$

This is the proximal update for $\iota_{\mathcal{Z}}$. The constraint couples adjacent hours, so projection is the joint quadratic problem $\min_{v\in\mathcal{Z}} \frac{1}{2}\|v-u\|^2$, not independent coordinate clipping. We use ADMM with primal and dual stopping tolerances 10^{-6} and a cap of 200 iterations. The same projection routine is used by all compared methods.

Projection is applied to every lower schedule update. The auxiliary linear-system state and the Hessian and cross-Hessian products use the smooth objective. All methods use this approximate projection, and the wall-clock curves include its cost.

Ordinary upper and lower oracle calls use $B = 64$; RCSL-GA uses batch size 128 when evaluating its contextual correction. At each reporting point, the evaluated predictor parameter x is the block average of the outer iterates accumulated since the preceding report. For each task i, an approximate representative schedule $\widehat{s}_i(x)$ is recomputed using 100 projected inner steps with minibatches from that task's training split. This schedule is then held fixed while the original scheduling cost is averaged over the full following-year evaluation split:

$$\widehat{F}_{\text{test}}(x) = \frac{1}{4}\sum_{i=1}^{4} \mathbb{E}_{(a,d)\sim\mathcal{P}_i^{\text{test}}}\left[\frac{1}{24}c(\widehat{s}_i(x), d)\right]. \tag{76}$$

Here "test" refers to the within-task evaluation split. For task i, observations from its evaluation split are excluded from its upper and lower training oracle calls and its reporting-time lower solve. This restriction is task-specific: those observations may still contribute to updates of the shared x through another task's training window. Evaluation time is excluded from optimization time.

We run RCSL-GA, FFO, FFO-RT-MLMC (Yang et al., 2025), and Hessian-RT-MLMC (Hu et al., 2023) for 1,000 outer iterations, evaluate every 20 iterations, and repeat every configuration with ten random seeds. Table 6 reports the settings used in the comparison.

Table 6: Power-scheduling settings: 1,000 iterations, evaluation gap 20, ordinary minibatches of 64, and common ramp projection. RCSL-GA correction minibatches contain 128 samples.

Method	Method-specific settings
RCSL-GA	$\alpha = 0.012$, $\beta = 0.015$, $\lambda_z = 0.015$, $p = 0.1$, $B_c = 5$, $K_y = K_z = 10$
FFO	$\alpha = 0.005$, inner scale $\beta = 50$ (Yang et al., 2025)
FFO-RT-MLMC	effective outer stepsize $\alpha_0 \times \epsilon^4 = 5 * 10^{-3}$, inner scale $\beta = 50$ (Yang et al., 2025)
Hessian-RT-MLMC	base outer stepsize 0.01, inner scale $\beta = 0.2$, $L_{g,1} = 16$ (Hu et al., 2023)

The parameters in Table 7 are selected by grid search on three different seed and the same task split of the main experiment, using the final validation objective diagnostic. The table lists the candidate grids and selected values.

Table 7: Power-scheduling hyperparameter grids and selected values.

Method	Parameter	Candidate values	Selected value
RCSL-GA	α	$\{0.5, 0.1, 0.02, 0.015, 0.012, 0.01\}$	0.012
RCSL-GA	β	$\{0.5, 0.1, 0.02, 0.015, 0.012, 0.01\}$	0.015
RCSL-GA	λ_z	$\{0.5, 0.1, 0.02, 0.015, 0.012, 0.01\}$	0.015
RCSL-GA	λ_y	$\{0.5, 0.1, 0.02, 0.015, 0.012, 0.01\}$	0.015
RCSL-GA	p	$\{0.5, 0.2, 0.1, 0.05, 0.01\}$	0.1
FFO	α	$\{0.1, 0.05, 0.01, 0.005, 0.001\}$	0.005
FFO	β	$\{500, 250, 50, 5, 0.5, 0.1\}$	50
FFO-RT-MLMC	$\alpha_0 * \epsilon^4$	$\{0.1, 0.05, 0.01, 0.005, 0.001\}$	0.005
FFO-RT-MLMC	β	$\{500, 250, 50, 5, 0.5, 0.1\}$	50
Hessian-RT-MLMC	α	$\{1, 0.5, 0.1, 0.05, 0.01\}$	0.01
Hessian-RT-MLMC	β	$\{12, 8, 0.5, 0.2, 0.1\}$	0.2