Residual-Controlled Multiplier Learning for Stochastic Constrained Decision-Making

Anonymous authors
Paper under double-blind review

Abstract

Stochastic constrained decision-making requires optimizing performance objectives while enforcing statistical requirements such as safety or fairness. However, standard primal–dual methods struggle to update multipliers robustly under stochastic mini-batch feedback, as the noise of mini-batch gradients and constraint estimates can be directly accumulated into the multiplier memory. To address this issue, we propose Residual-Controlled Multiplier Learning (RCML), which reformulates multiplier updating as projected-pressure feedback. The central idea is to decompose the projected multiplier into an effective pressure signal for primal descent and a pressure-memory residual for finite-gain multiplier tracking. To handle heterogeneous and noisy observations, we further augment this residual-integral backbone with modular stochastic stabilization components. For the convex-affine backbone, we establish finite-gain convergence, derive a stochastic residual bound under mini-batch feedback, and show that the residual feedback law admits a local KKT-residual interpretation near regular KKT points of nonconvex problems. Experiments across optimization, allocation, and fair-ranking tasks show that RCML improves feasibility control and multiplier stability while maintaining competitive objective performance. Code is released at `https://anonymous.4open.science/r/RCML-3114/`.

1 Introduction

Stochastic constrained decision-making arises in many data-driven systems Goh et al. (2016); Chamon & Ribeiro (2020) where performance objectives must be optimized under statistical constraints (safety limits (Tessler et al., 2019; Stooke et al., 2020), fairness requirements (Agarwal et al., 2018; Cotter et al., 2019a), risk budgets, etc.). Primal–dual methods provide a natural algorithmic interface for solving the stochastic constrained optimization problems (Xu, 2020; Jin & Wang, 2022), which update the primal decision variable while maintaining nonnegative multipliers that encode the pressure of constraint violations (Bertsekas, 1982; Nocedal & Wright, 2006). Unfortunately, the multiplier update is fragile in the stochastic case because the algorithm only observes noisy mini-batch estimates of the constraints (Lan & Zhou, 2020; Zhang et al., 2022). The major difficulty is how the stored multiplier memory should respond to noisy and varying constraint information.

Raw constraint violation signals are inherently insufficient for stochastic dual adaptation, as they fail to simultaneously capture constraint activation and multiplier dissipation. Specifically, while a signed-violation signal can actively decrease the multiplier when a constraint becomes feasible, it invariably propagates stochastic noise from inactive constraints into the dual dynamics. A positive-violation signal successfully suppresses inactive noise channels, but it loses the capacity to actively reduce the multiplier upon feasibility recovery, causing accumulated multiplier memory to become permanently stale. Although projected augmented-Lagrangian methods reinforce feasibility via a projected multiplier (Rockafellar, 1976; Murtagh & Saunders, 1982), their standard full-replacement update introduces severe multiplier fluctuations when the underlying projected pressure is estimated under noisy mini-batch feedback (Li et al., 2024; Na et al., 2023).

To address this issue, we propose Residual-Controlled Multiplier Learning (RCML), a projected-pressure feedback framework for stochastic constrained decision-making. The central idea is to separate the projected-

pressure mechanism into two coupled signals: 1) an effective pressure signal for the primal update; 2) a pressure-memory residual for multiplier adaptation. In this way, the multiplier is no longer updated by directly accumulating raw constraint violations or by fully replacing the stored state with the projected pressure. Instead, RCML lets the stored multiplier track the projected pressure through a finite-gain residual feedback law, which naturally supports violation activation, stale-memory release, and inactive-constraint dead-zone behaviors.

The proposed method is developed in three steps. 1) we identify the pressure-memory residual induced by the inequality augmented Lagrangian and show that its zero set exactly captures feasibility and complementarity; 2) we use this residual as the input of a stochastic primal–dual learning rule, where the projected pressure drives the primal step and the residual controls multiplier memory; 3) we equip this residual-integral backbone with stabilization modules, including constraint filtering, adaptive coordinate-wise scaling, and residual-νPI correction, to handle noisy and heterogeneous mini-batch constraint feedback. This control-oriented view is related to primal–dual dynamics and feedback interpretations of Lagrange multipliers (Feijer & Paganini, 2010; Cherukuri et al., 2016), as well as recent I/PI multiplier updates (Stooke et al., 2020; Sohrabi et al., 2024).

We evaluate RCML on stochastic constrained optimization problems, stochastic energy-reserve allocation, nonconvex pricing-inventory allocation, and neural fair ranking, which follows exposure-based ranking fairness and cumulative-gain ranking evaluation (Singh & Joachims, 2018; Järvelin & Kekäläinen, 2002). The experiments examine not only feasibility and objective quality, but also multiplier-memory behavior under noisy constraint feedback.

Our main contributions are as follows.

- We propose a pressure–memory feedback mechanism for multiplier updates. By driving the primal step with the projected pressure and controlling the multiplier dynamics through the induced residual, the mechanism enables a single signal to naturally combine violation activation, stale-memory release, and an inactive-constraint dead-zone.
- We design a stochastic primal–dual algorithm that integrates finite-gain multiplier tracking with modular stabilization components. Under noisy and heterogeneous feedback, RCML achieves substantially lower multiplier fluctuation and smaller expected constraint violation than standard primal–dual updates.
- We establish finite-gain convergence of the deterministic convex-affine backbone and derive stochastic residual bounds for the mini-batch setting. These results reveal how the residual tracking law preserves primal–dual Lyapunov cancellation, and they quantify the roles of filtering, adaptive scaling, and dynamic correction inside the projected-pressure feedback loop.

2 Related Work

This section reviews prior work along the three ingredients that motivate RCML. First, constrained learning explains why multiplier states are needed: statistical requirements such as safety and fairness cannot usually be enforced by a fixed penalty weight, so the tradeoff between objective descent and constraint satisfaction must be adjusted dynamically. Second, stochastic constrained optimization explains why this adjustment is difficult: mini-batch constraint feedback is noisy, and therefore multiplier memory can become unstable when it is driven by raw violation signals or direct projected replacement. Third, feedback-control interpretations of multiplier dynamics explain where RCML intervenes: once the multiplier is viewed as a memory state in a feedback loop, the central design question becomes what signal should drive this memory.

Constrained learning and dynamic multiplier states Many data-driven decision systems must optimize a prediction or control objective while satisfying statistical requirements. Safety and reliability requirements are often formulated as expectation constraints on learned policies or decisions (Goh et al., 2016; Chamon & Ribeiro, 2020), while fairness-constrained learning imposes group-level constraints during classification or ranking (Agarwal et al., 2018; Cotter et al., 2019b; Singh & Joachims, 2018). These constraints

depend on the current model, data distribution, and sampling process, so they cannot always be replaced by a fixed regularization weight. Constrained learning methods therefore introduce dual variables, proxy losses, or game-theoretic players to adapt the tradeoff between objective descent and constraint satisfaction during training (Cotter et al., 2019a; Agarwal et al., 2018; Chamon & Ribeiro, 2020). Related constrained reinforcement learning methods use Lagrangian or primal–dual mechanisms to control expected cumulative constraint violations during policy optimization (Tessler et al., 2019; Stooke et al., 2020). In these methods, the multiplier is not only a static penalty coefficient, but also a dynamic memory state that records past constraint pressure and affects future primal updates. This makes multiplier-memory updating a central algorithmic component, especially when the constraint signal is observed only through stochastic mini-batches.

Stochastic constrained optimization under noisy feedback Stochastic constrained optimization studies how to update primal and dual variables using sampled objective and constraint information. Stochastic approximation methods provide foundational mechanisms for handling expectation constraints from noisy samples (Wang & Bertsekas, 2016; Lan & Zhou, 2020). Stochastic primal–dual methods extend this idea by coupling primal descent with multiplier ascent or correction under sampled feedback (Xu, 2020; Jin & Wang, 2022). When constraints are nonlinear or functional, constraint extrapolation and linearization-based methods improve feasibility control without requiring exact full-batch projections (Boob et al., 2023). For nonconvex constrained problems, single-loop perturbed ascent and related first-order schemes reduce the need for nested constrained subproblem solves (Lu, 2022; Alacaoglu & Wright, 2024). Another line of work replaces hard constraint handling with smoothed penalties or relaxed feasibility surrogates to obtain tractable stochastic updates (Huang et al., 2025; Yang et al., 2026). Stochastic augmented Lagrangian methods combine multiplier updates with penalty-based correction to stabilize expectation-constrained optimization under sampling noise (Zhang et al., 2022; Li et al., 2024). Stochastic SQP methods instead build local quadratic models and use active-set or proximal corrections to obtain more structured constrained steps (Na et al., 2023; Cui et al., 2025). These approaches improve feasibility control by changing the primal–dual update, the local model, or the correction mechanism. However, the multiplier memory is still commonly driven by 1) raw signed violations, 2) positive violations, or 3) direct projected multiplier replacement. Thus, existing stochastic constrained optimization mainly stabilizes the optimization loop around the constraint signal, whereas the design of the multiplier-memory input itself remains less explicit.

Feedback-control views and multiplier signal design To systematically re-engineer this memory input, a control-theoretic perspective proves essential, interpreting multiplier updates directly as closed-loop feedback laws. Classical augmented Lagrangian and projected multiplier methods already use the projected quantity to handle inequality constraints (Rockafellar, 1976; Murtagh & Saunders, 1982). Standard nonlinear programming treatments further connect projected or augmented multiplier updates with feasibility and complementarity enforcement (Bertsekas, 1982; Nocedal & Wright, 2006). Dynamical analyses of primal–dual algorithms show that multiplier evolution can be understood as a feedback process coupled with primal motion (Feijer & Paganini, 2010; Cherukuri et al., 2016). In safe reinforcement learning, the same idea appears as integral multiplier control, where accumulated constraint errors regulate policy updates (Tessler et al., 2019). PID-style Lagrangian methods make this control interpretation more explicit by adding proportional or derivative components to improve transient constraint behavior (Stooke et al., 2020; Sohrabi et al., 2024). Recent continuous-time and nonsmooth analyses further formalize the feedback nature of Lagrangian dynamics (Cerone et al., 2025; 2026). These works clarify that the multiplier is a feedback state, but the feedback input is usually built from raw constraint errors or their filtered variants. Our proposed RCML follows the same control viewpoint but changes the input to the multiplier-memory loop. Instead of feeding the memory update directly with raw violations or fully replacing the memory by the projected multiplier, RCML first forms the projected effective multiplier and then updates memory through the induced pressure-memory residual. This residual input combines violation activation, stale-memory release, and inactive-constraint dead-zone behavior within one projected-pressure feedback interface.

3 Problem Statement and Preliminaries

3.1 Stochastic Constrained Decision-Making Problem

We consider the stochastic inequality-constrained problem (Lan & Zhou, 2020; Zhang et al., 2022)

$$\min_{x\in\mathcal{X}} f(x) := \mathbb{E}_\xi[\ell(x;\xi)] \quad \text{s.t.} \quad c_i(x) := \mathbb{E}_\xi[h_i(x;\xi)] \le 0, \quad i = 1,\dots,m, \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ is the decision variable, ξ denotes a random sample, $\ell(x;\xi)$ is the sample-wise objective loss, and $h_i(x;\xi)$ is the sample-wise contribution to the i-th constraint. We write $c(x) = [c_1(x),\dots,c_m(x)]^\top$, $h(x;\xi) = [h_1(x;\xi),\dots,h_m(x;\xi)]^\top$, and denote the constraint Jacobian by $J_c(x) \in \mathbb{R}^{m\times d}$. In learning applications, the mini-batch constraint Jacobian $\widehat{J}_{c,k}$ is obtained by differentiating the mini-batch constraint estimate with respect to the model parameters.

The expectation-form Lagrangian of equation 1 is (Bertsekas, 1982; Nocedal & Wright, 2006)

$$\mathcal{L}(x,u) = f(x) + u^\top c(x) = \mathbb{E}_\xi\big[\ell(x;\xi) + u^\top h(x;\xi)\big], \quad u \in \mathbb{R}^m_+. \tag{2}$$

Its primal descent direction is $\nabla_x\mathcal{L}(x,u) = \nabla f(x) + J_c(x)^\top u$, and its multiplier-side ascent direction is $c(x)$. Under exact information, a projected primal–dual method updates the primal variable along $\nabla f(x_k) + J_c(x_k)^\top u_k$ and updates the multiplier along $c(x_k)$, with projection onto $\mathcal{X}$ and $\mathbb{R}^m_+$.

In stochastic learning or data-driven decision-making, the exact quantities $\nabla f(x_k)$, $c(x_k)$, and $J_c(x_k)$ are usually unavailable at each iteration. The algorithm instead observes mini-batch quantities $\hat{g}_k$, $\hat{c}_k$, and $\widehat{J}_{c,k}$, which are used as stochastic estimates of $\nabla f(x_k)$, $c(x_k)$, and $J_c(x_k)$, respectively. The key issue is not only that these estimates are noisy, but also that the noise can be transmitted into the stored multiplier memory if the memory is updated directly from raw constraint feedback.

3.2 A Generalized Primal–Dual Interface

To cover both classical stochastic primal–dual updates and the proposed RCML update, we use the following generalized interface:

$$x_{k+1} = \Pi_{\mathcal{X}}\big(x_k - \alpha_k(\hat{g}_k + \widehat{J}_{c,k}^\top \mu_k)\big), \quad u_{k+1} = \Pi_{\mathbb{R}^m_+}(u_k + \eta_k s_k). \tag{3}$$

Here $\Pi_{\mathcal{S}}$ denotes Euclidean projection onto a set $\mathcal{S}$, $u_k \in \mathbb{R}^m_+$ is the stored multiplier memory, $\mu_k \in \mathbb{R}^m_+$ is the effective multiplier pressure applied to the primal update, and $s_k \in \mathbb{R}^m$ is the feedback signal used to update the multiplier memory. The stepsizes $\alpha_k > 0$ and $\eta_k > 0$ control the primal and multiplier-memory updates.

The classical stochastic primal–dual update is recovered from equation 3 by choosing $\mu_k = u_k, s_k = \hat{c}_k$, or by replacing s_k with the positive violation signal $[\hat{c}_k]_+$ (Bertsekas, 1982). This choice ties two different roles to the same raw constraint-feedback loop: the stored multiplier u_k is used as the pressure in the primal update, and the mini-batch constraint estimate is directly integrated into the multiplier memory. Consequently, stochastic fluctuations in $\hat{c}_k$ can accumulate in u_k, especially when the constraint signal is highly variable across mini-batches.

The central design question is therefore whether the pressure applied to the primal update and the signal used to update multiplier memory should be tied to the same feedback signal. RCML answers this question by separating these two roles. It constructs a projected effective pressure for the primal update and a pressure-memory residual for updating the stored multiplier memory.

3.3 Projected Pressure and Memory Residual from the Augmented Lagrangian

We next introduce the projected-pressure signal induced by the inequality augmented Lagrangian. Let $\boldsymbol{\rho} = (\rho_1,\dots,\rho_m)^\top \in \mathbb{R}^m_{++}$ be a coordinate-wise pressure scale. The inequality augmented Lagrangian is (Rockafellar, 1976; Bertsekas, 1982)

$$\mathcal{L}_{\boldsymbol{\rho}}(x,u) = f(x) + \frac{1}{2}\sum_{i=1}^m \frac{[u_i + \rho_i c_i(x)]_+^2 - u_i^2}{\rho_i}, \tag{4}$$

where $[\cdot]_+$ denotes coordinate-wise projection onto the nonnegative orthant. This augmented-Lagrangian object induces two signals,

$$\lambda_{\boldsymbol{\rho}}(x,u) := [u + \boldsymbol{\rho} \odot c(x)]_+, \quad d_{\boldsymbol{\rho}}(x,u) := \lambda_{\boldsymbol{\rho}}(x,u) - u, \tag{5}$$

where $\odot$ denotes the Hadamard product. The vector $\lambda_{\boldsymbol{\rho}}(x,u)$ is the projected effective multiplier pressure, while $d_{\boldsymbol{\rho}}(x,u)$ is the pressure-memory residual that measures the gap between the projected pressure and the stored multiplier memory. Direct differentiation of equation 4 obtains $\nabla_x \mathcal{L}_{\boldsymbol{\rho}}(x,u) = \nabla f(x) + J_c(x)^\top \lambda_{\boldsymbol{\rho}}(x,u)$, $\nabla_u \mathcal{L}_{\boldsymbol{\rho}}(x,u) = \boldsymbol{\rho}^{-1} \odot d_{\boldsymbol{\rho}}(x,u)$, where $\boldsymbol{\rho}^{-1} = (1/\rho_1, \ldots, 1/\rho_m)^\top$. Therefore, $\lambda_{\boldsymbol{\rho}}$ is the multiplier pressure appearing in the primal derivative, and $d_{\boldsymbol{\rho}}$ is the multiplier-side residual generated by the same augmented-Lagrangian formulation.

In the isotropic case $\boldsymbol{\rho} = \rho_0 \mathbf{1}$, we write $\lambda_{\rho_0} := \lambda_{\rho_0 \mathbf{1}}$, $d_{\rho_0} := d_{\rho_0 \mathbf{1}}$. Then $\nabla_u \mathcal{L}_{\rho_0 \mathbf{1}}(x,u) = \rho_0^{-1} d_{\rho_0}(x,u)$. Classical projected augmented-Lagrangian methods use the projected pressure as the next multiplier memory. In an online update, this gives $u_{k+1} = \lambda_{\boldsymbol{\rho}}(x_k, u_k)$, whereas an offline augmented-Lagrangian convention may use $u_{k+1} = \lambda_{\boldsymbol{\rho}}(x_{k+1}, u_k)$. Thus, the memory is fully replaced by the current projected pressure. This unit-gain replacement is effective under exact deterministic information, but under stochastic mini-batch feedback it can transfer the full fluctuation of the projected pressure estimate into the multiplier memory.

3.4 Feedback-Control Interpretation

The generalized interface equation 3 can be interpreted as a closed-loop feedback system. The primal state x_k is regulated by the effective pressure μ_k, and the stored multiplier memory u_k is updated through the feedback signal s_k.

For the classical stochastic primal–dual update with $s_k = \hat{c}_k$, ignoring the nonnegative projection gives the formal recursion $u_k = u_0 + \sum_{t=0}^{k-1} \eta_t \hat{c}_t$. Thus, the multiplier memory behaves as an integral controller that accumulates the measured constraint signal. This interpretation explains its sensitivity to high-frequency stochastic noise: mini-batch fluctuations in $\hat{c}_t$ are directly accumulated into u_k. A signed violation signal can release multiplier mass when a constraint becomes feasible, but it also transmits fluctuations from inactive constraints. A positive violation signal suppresses negative inactive fluctuations, but it cannot actively remove stale positive multiplier memory through the constraint signal itself.

The projected augmented-Lagrangian pressure in equation 5 provides a richer feedback signal. In the active region $u_i + \rho_i c_i(x) > 0$, the i-th projected pressure satisfies $\lambda_{\boldsymbol{\rho},i}(x,u) = u_i + \rho_i c_i(x)$. Hence, the pressure combines an integral component u_i with a proportional correction $\rho_i c_i(x)$. In the inactive region $u_i + \rho_i c_i(x) \le 0$, the projection sets $\lambda_{\boldsymbol{\rho},i}(x,u) = 0$, introducing the saturation and dead-zone behavior required for inequality constraints.

From this viewpoint, projected augmented-Lagrangian methods already provide a useful projected pressure for the primal update, but they couple this pressure with unit-gain memory replacement. RCML keeps the projected-pressure correction for the primal update and replaces unit-gain memory replacement with finite-gain residual tracking. The projected pressure $\lambda_{\boldsymbol{\rho}}$ is used as the effective pressure in the primal channel, while the induced residual $d_{\boldsymbol{\rho}}$ becomes the feedback signal that controls the multiplier memory.

4 Methodology

Building on the two signals identified in the previous section, we now construct a systematic framework that uses λ for the primal update and d for memory tracking. We first analyze the residual signal structure, then embed it into a unified algorithm with finite-gain memory control, and finally introduce optional stabilization modules for stochastic feedback.

4.1 Residual-Controlled Multiplier Signal

The key signal in RCML is the pressure-memory residual $d_{\boldsymbol{\rho}}$ defined in equation 5, which measures how far the stored multiplier memory u is from the projected pressure $\lambda_{\boldsymbol{\rho}}(x,u)$. From equation 5, the coordinate-wise

form is

$$d_{\boldsymbol{\rho},i}(x,u) = \begin{cases} \rho_i c_i(x), & u_i + \rho_i c_i(x) > 0, \\ -u_i, & u_i + \rho_i c_i(x) \le 0. \end{cases}$$

This piecewise structure gives the residual three feedback behaviors. First, it provides *violation activation*: when constraint i is violated and the projected pressure is active, $d_{\boldsymbol{\rho},i}(x,u) = \rho_i c_i(x) > 0$, which increases the multiplier memory in proportion to the violation. Second, it provides *stale-memory release*: if a previously active constraint becomes sufficiently feasible while $u_i > 0$, then $u_i + \rho_i c_i(x) \le 0$ and $d_{\boldsymbol{\rho},i}(x,u) = -u_i$, so the obsolete multiplier value is actively removed. Third, it provides an *inactive dead-zone*: if $u_i = 0$ and $c_i(x) < 0$, then $d_{\boldsymbol{\rho},i}(x,u) = 0$, so a strictly inactive constraint does not inject negative noise into the multiplier dynamics.

This explains why $d_{\boldsymbol{\rho}}$ is more informative than either the raw violation $c(x)$ or its positive part $[c(x)]_+$. The signed signal $c(x)$ can release memory but also transmits inactive negative fluctuations; the positive signal $[c(x)]_+$ suppresses negative inactive fluctuations but cannot release stale positive memory. The residual $d_{\boldsymbol{\rho}}$ unifies activation, release, and dead-zone behavior in a single projected-pressure feedback signal.

The residual also carries an exact optimality meaning. As shown in Proposition 5.2 (Section 5), $d_{\boldsymbol{\rho}}(x,u) = 0$ is equivalent to $u \ge 0$, $c(x) \le 0$, and $u_i c_i(x) = 0$ for all i. Together with projected stationarity of the primal step, this yields the KKT system. Hence, RCML employs a memory-control signal that is both operationally meaningful in the feedback loop and mathematically aligned with constrained optimality.

4.2 RCML Finite-Gain Memory Tracking and Unified Algorithm

RCML applies the projected pressure to the primal update but adjusts the multiplier memory through the residual. At iteration k, the algorithm first forms a constraint signal $\tilde{c}_k$, which may be the raw mini-batch estimate $\hat{c}_k$ or a filtered version introduced in Section 4.3. It also selects a pressure-scale vector $\boldsymbol{\rho}_k \in \mathbb{R}^m_{++}$, which can be fixed, coordinate-wise, or adaptively updated.

Using the current signal $\tilde{c}_k$ and scale $\boldsymbol{\rho}_k$, the online projected pressure and residual are obtained by analogy with equation 5:

$$\lambda_k = [u_k + \boldsymbol{\rho}_k \odot \tilde{c}_k]_+, \quad d_k = \lambda_k - u_k.$$

In the generalized primal–dual template equation 3, RCML sets the primal pressure to $\mu_k = \lambda_k$ and constructs the memory feedback signal as $s_k = \kappa_{\mathrm{I}} d_k$ with gain $\kappa_{\mathrm{I}} > 0$. The multiplier memory update then becomes

$$u_{k+1} = \Pi_{\mathbb{R}^m_+}\big(u_k + \eta_k \kappa_{\mathrm{I}} d_k\big). \tag{6}$$

Let $\beta_k := \eta_k \kappa_{\mathrm{I}}$. If $0 \le \beta_k \le 1$, then using $d_k = \lambda_k - u_k$ gives $u_{k+1} = (1-\beta_k)u_k + \beta_k \lambda_k$, so the projection in equation 6 can be dropped because both u_k and λ_k are nonnegative. Thus the multiplier memory moves only a fraction β_k toward the current projected pressure.

The classical projected ALM update is recovered as the unit-gain case $\beta_k = 1$. Smaller values of β_k produce a smoother multiplier trajectory and reduce sensitivity to instantaneous mini-batch noise. In this way, RCML retains the projected-pressure correction of ALM while avoiding its aggressive full-replacement memory update.

Algorithm 1 summarizes the unified RCML execution flow. The minimal RCML backbone consists of two choices: $\mu_k = \lambda_k$ and $s_k = \kappa_{\mathrm{I}} d_k$. All other components, including constraint filtering, adaptive scaling, and residual-νPI correction, are optional stochastic stabilization modules.

The algorithm contains three signal channels: 1) *Measurement channel*: converts the raw mini-batch constraint estimate $\hat{c}_k$ into the processed signal $\tilde{c}_k$ (the filtered option is detailed in Section 4.3); 2) *Pressure channel*: computes the projected pressure λ_k, which is then applied to the primal step through $\widehat{J}_{c,k}^{\top}\lambda_k$; 3) *Memory-feedback channel*: forms the residual-based signal s_k and updates the stored multiplier memory u_k.

Algorithm 1 Unified Residual-Controlled Multiplier Learning

Require: Choose stepsizes $\{\alpha_k, \eta_k\} > 0$, initialize $x_0 \in \mathcal{X}$ and $u_0 \in \mathbb{R}^m_+$. If filtering, adaptive scaling, or residual-νPI correction is used, initialize their auxiliary states $\bar{c}_{-1} = 0$, $v_0 = 0$, $\xi_{-1} = 0$ and choose memory signal s_k (see Table 1).

1: **for** $k = 0, 1, \ldots, T-1$ **do**
2: Observe mini-batch estimates $\hat{g}_k$, $\hat{c}_k$, and $\widehat{J}_{c,k}$.
3: Form the constraint signal $\tilde{c}_k$ from $\hat{c}_k$ (e.g., via filtering or using the raw estimate).
4: Compute the projected pressure and residual: $\lambda_k = [u_k + \boldsymbol{\rho}_k \odot \tilde{c}_k]_+$, $d_k = \lambda_k - u_k$.
5: Update the primal variable: $x_{k+1} = \Pi_{\mathcal{X}}\big(x_k - \alpha_k[\hat{g}_k + \widehat{J}_{c,k}^\top \lambda_k]\big)$.
6: Update the multiplier memory: $u_{k+1} = \Pi_{\mathbb{R}^m_+}(u_k + \eta_k s_k)$.
7: **end for**

Table 1: RCML variants and classical baselines under the unified projected-pressure execution flow. λ_k and d_k denote the projected pressure and pressure-memory residual obtained from the selected constraint signal $\tilde{c}_k$ and pressure-scale vector $\boldsymbol{\rho}_k$. Setting $\eta_k = 1$ and $s_k = d_k$ recovers the classical projected augmented-Lagrangian replacement update (Rockafellar, 1976).

Method	Constraint signal	Pressure scale	Memory signal	Memory update	Main role
SGDA-Signed (Jin & Wang, 2022)	$\tilde{c}_k = \hat{c}_k$	Not used	$s_k = \hat{c}_k$	$\Pi_{\mathbb{R}^m_+}(u_k + \eta_k s_k)$	Signed integral
SGDA-Positive (Xu, 2020)	$\tilde{c}_k = \hat{c}_k$	Not used	$s_k = [\hat{c}_k]_+$	$\Pi_{\mathbb{R}^m_+}(u_k + \eta_k s_k)$	Positive integral
Projected-ALM (Bertsekas, 1982)	$\tilde{c}_k = \hat{c}_k$	$\boldsymbol{\rho}_k = \rho_0 \mathbf{1}$	$s_k = d_k$	$\eta_k = 1$ $u_{k+1} = \lambda_k$	Unit-gain tracking
Residual-I	$\tilde{c}_k = \hat{c}_k$	$\boldsymbol{\rho}_k = \rho_0 \mathbf{1}$	$s_k = \kappa_{\mathrm{I}} d_k$	$u_{k+1} = u_k + \eta_k s_k$ $0 \le \eta_k \kappa_{\mathrm{I}} \le 1$	Finite-gain tracking
RCML-Core	$\tilde{c}_k = \bar{c}_k$	$\boldsymbol{\rho}_k = \rho_0 \mathbf{1}$	$s_k = \kappa_{\mathrm{I}} d_k$	$u_{k+1} = u_k + \eta_k s_k$ $0 \le \eta_k \kappa_{\mathrm{I}} \le 1$	Filtered tracking
RCML-Adaptive	$\tilde{c}_k = \bar{c}_k$	$\boldsymbol{\rho}_k = [\rho_{k,1}, \ldots, \rho_{k,m}]^\top$	$s_k = \kappa_{\mathrm{I}} d_k$	$u_{k+1} = u_k + \eta_k s_k$ $0 \le \eta_k \kappa_{\mathrm{I}} \le 1$	Adaptive scaling
RCML-Robust	$\tilde{c}_k = \bar{c}_k$	$\boldsymbol{\rho}_k = [\rho_{k,1}, \ldots, \rho_{k,m}]^\top$	$s_k = \kappa_{\mathrm{I}} d_k + \kappa_{\mathrm{P}}(\xi_k - \xi_{k-1})$	$\Pi_{\mathbb{R}^m_+}(u_k + \eta_k s_k)$	Dynamic correction

Notes. The filtered signal $\bar{c}_k$, adaptive coordinates $\rho_{k,i}$, and residual smoothing state ξ_k are defined in Section 4.3. For residual-integral variants, the projection in the multiplier update is inactive when $0 \le \eta_k \kappa_{\mathrm{I}} \le 1$ because u_{k+1} becomes a convex combination of u_k and λ_k. For RCML-Robust, the additional dynamic correction may break this convex combination, so the non-negative projection is retained as a safeguard.

4.3 Stochastic Stabilization Modules

In stochastic constrained learning, mini-batch feedback can be very noisy. We therefore equip RCML with three optional stabilization modules. These modules do not alter the meaning of the projected-pressure residual; they only modify the measurement, pressure, or memory-feedback channel.

Constraint filtering The first module acts on the measurement channel. Instead of forming the pressure directly from the raw mini-batch estimate $\hat{c}_k$, RCML can use an exponential moving average:

$$\tilde{c}_k = (1 - \gamma_k)\, \tilde{c}_{k-1} + \gamma_k\, \hat{c}_k, \quad \gamma_k \in (0, 1]. \tag{7}$$

The parameter γ_k controls the noise–lag trade-off. When $\gamma_k = 1$, the module is inactive and $\tilde{c}_k = \hat{c}_k$. Smaller values reduce high-frequency stochastic fluctuations but introduce delay in detecting rapid constraint changes.

Adaptive coordinate-wise pressure scaling The second module acts on the pressure channel. A single isotropic pressure scale $\rho_0 \mathbf{1}$ can be unsuitable when constraints have very different magnitudes: a large-scale

constraint may dominate the projected pressure, while a small-scale one may receive insufficient correction. To reduce this imbalance, RCML can adapt the coordinate-wise scale $\boldsymbol{\rho}_k = [\rho_{k,1}, \ldots, \rho_{k,m}]^\top$.

Using the filtered constraint signal, we maintain a bias-corrected second-moment estimate:

$$v_{k+1} = (1-\eta_v)v_k + \eta_v\, \tilde{c}_k^2, \quad \widehat{v}_{k+1} = \frac{v_{k+1}}{1-(1-\eta_v)^{k+1}}.$$

The adaptive coordinate-wise pressure scale is then

$$\rho_{k,i} = \mathrm{clip}\left(\frac{\kappa_\rho}{\sqrt{\widehat{v}_{k+1,i}+\epsilon}},\ \rho_{\min},\ \rho_{\max}\right), \quad i = 1, \ldots, m. \tag{8}$$

Here $\kappa_\rho > 0$ is the base scale, $\epsilon > 0$ prevents division by zero, and $\rho_{\min}, \rho_{\max}$ keep the pressure scale within a stable range. The bias correction in $\widehat{v}_{k+1}$ is exact when the second moment of the constraint signal is stationary; under distribution shift or rapidly moving iterates it should be viewed as a pragmatic re-scaling rather than an exact moment correction.

Residual-νPI dynamic correction The third module acts on the memory-feedback channel. The pure residual-integral update $s_k = \kappa_{\mathrm{I}} d_k$ gives a conservative finite-gain controller. To accelerate the transient response when constraint pressure changes rapidly, RCML can add a dynamic correction based on a smoothed residual state:

$$\xi_k = \nu\, \xi_{k-1} + (1-\nu)\, d_k, \quad \nu \in [0, 1). \tag{9}$$

The difference $\xi_k - \xi_{k-1}$ captures the recent trend of the pressure-memory mismatch. The complete residual-νPI memory signal is

$$s_k = \kappa_{\mathrm{I}} d_k + \kappa_{\mathrm{P}}(\xi_k - \xi_{k-1}), \quad \kappa_{\mathrm{I}} > 0, \quad \kappa_{\mathrm{P}} \geq 0. \tag{10}$$

The term $\kappa_{\mathrm{I}} d_k$ provides integral action on the mismatch, while $\kappa_{\mathrm{P}}(\xi_k - \xi_{k-1})$ adds a proportional correction on its filtered change. Setting $\kappa_{\mathrm{P}} = 0$ recovers the pure finite-gain residual-integral backbone. Because the dynamic term may cause the memory update to deviate from a convex combination of u_k and λ_k, the projection onto $\mathbb{R}^m_+$ is retained for this variant.

5 Theoretical Analysis

The results are organized as follows. Section 5.1 shows that $d_{\boldsymbol{\rho}}$ is an exact feasibility–complementarity residual and that, together with projected stationarity, it characterizes the KKT system. Section 5.2 analyzes the deterministic convex-affine residual-integral backbone corresponding to the finite-gain update in Section 4.2. Section 5.3 derives a stopped finite-time stochastic residual bound under mini-batch feedback. Sections 5.4 and 5.5 then explain how constraint filtering, adaptive pressure scaling, and residual-νPI correction enter the measurement, pressure, and memory-feedback channels.

Scope The residual identities in Section 5.1 hold for general smooth nonlinear constraints $c(x)$. The dynamical convergence and stochastic residual bounds in Sections 5.2 and 5.3 are established for the convex-affine backbone $c(x) = Ax - b$ and $\mathcal{X} = \mathbb{R}^d$. In these two subsections, the vector scale is specialized to the isotropic form $\boldsymbol{\rho} = \rho_0 \mathbf{1}$ with $\rho_0 > 0$, and the scalar symbol ρ_0 is used only for this isotropic specialization. The nonlinear formulation in Section 4 defines the implementable RCML mechanism, while the convex-affine analysis isolates the finite-gain residual-memory dynamics in a setting where the Lyapunov cancellation can be proved exactly. The final nonconvex discussion is used to interpret the residual diagnostics near regular KKT points.

Definition 5.1. For $x \in \mathcal{X}$, the projected stationarity residual is

$$\mathcal{G}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) := \alpha^{-1} \left\| x - \Pi_{\mathcal{X}}\big(x - \alpha[\nabla f(x) + J_c(x)^\top \lambda_{\boldsymbol{\rho}}(x,u)]\big) \right\|,$$

and the combined residual is

$$\mathcal{R}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) := \big(\mathcal{G}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u)\big)^2 + \|d_{\boldsymbol{\rho}}(x,u)\|^2.$$

The projected stationarity residual follows the standard gradient-mapping and normal-cone characterization of constrained stationarity (Bertsekas, 1982; Nocedal & Wright, 2006; Bauschke & Combettes, 2017). When $\mathcal{X} = \mathbb{R}^d$, the projected stationarity term reduces to the gradient norm: $\mathcal{R}^{\mathbb{R}^d}_{\alpha,\boldsymbol{\rho}}(x,u) = \|\nabla f(x) + J_c(x)^\top \lambda_{\boldsymbol{\rho}}(x,u)\|^2 + \|d_{\boldsymbol{\rho}}(x,u)\|^2$.

5.1 Residual Geometry and KKT Characterization

We first prove that the pressure-memory residual $d_{\boldsymbol{\rho}}$ defined in equation 5 is an exact optimality residual. Throughout this subsection, $\boldsymbol{\rho} \in \mathbb{R}^m_{++}$ is allowed to be coordinate-wise.

Proposition 5.2 (Residual zero set)**.** *For any $\boldsymbol{\rho} \in \mathbb{R}^m_{++}$,*

$$d_{\boldsymbol{\rho}}(x,u) = 0 \iff u \geq 0, \quad c(x) \leq 0, \quad u_i c_i(x) = 0, \quad i = 1, \ldots, m.$$

Proof. (i) Necessity. Let $d_{\boldsymbol{\rho}}(x,u) = 0$. From equation 5, this is equivalent to $u = [u + \boldsymbol{\rho} \odot c(x)]_+$. Since $[\cdot]_+$ is the projection onto $\mathbb{R}^m_+$, for each coordinate i: if $u_i > 0$, then $u_i = [u_i + \rho_i c_i(x)]_+$ implies $u_i + \rho_i c_i(x) > 0$, hence $u_i = u_i + \rho_i c_i(x)$ and $c_i(x) = 0$; if $u_i = 0$, then $0 = [\rho_i c_i(x)]_+$ gives $c_i(x) \leq 0$. Therefore $c(x) \leq 0$ and $u_i c_i(x) = 0$ for all i.

(ii) Sufficiency. Let $u \geq 0$, $c(x) \leq 0$, and $u_i c_i(x) = 0$. If $u_i > 0$, complementarity implies $c_i(x) = 0$, hence $[u_i + \rho_i c_i(x)]_+ = u_i$. If $u_i = 0$, feasibility implies $c_i(x) \leq 0$, hence $[u_i + \rho_i c_i(x)]_+ = [\rho_i c_i(x)]_+ = 0 = u_i$. Thus $[u + \boldsymbol{\rho} \odot c(x)]_+ = u$, i.e., $d_{\boldsymbol{\rho}}(x,u) = 0$. □

Proposition 5.3 (KKT characterization by the combined residual)**.** *For any $\alpha > 0$, $\mathcal{R}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) = 0$ if and only if (x,u) satisfies the KKT system of $\{\min_{x \in \mathcal{X}} f(x) \text{ s.t. } c(x) \leq 0\}$.*

Proof. (i) Necessity. If $\mathcal{R}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) = 0$, then $d_{\boldsymbol{\rho}}(x,u) = 0$. Proposition 5.2 gives $c(x) \leq 0$, $u \geq 0$, and $u_i c_i(x) = 0$. Moreover, from equation 5, $d_{\boldsymbol{\rho}}(x,u) = 0$ implies $\lambda_{\boldsymbol{\rho}}(x,u) = u$. The condition $\mathcal{G}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) = 0$ yields $x = \Pi_{\mathcal{X}}\big(x - \alpha[\nabla f(x) + J_c(x)^\top u]\big)$. By the optimality condition of Euclidean projection onto a convex set (Bertsekas, 1982; Bauschke & Combettes, 2017), $z = \Pi_{\mathcal{X}}(y)$ is equivalent to $y - z \in N_{\mathcal{X}}(z)$, where $N_{\mathcal{X}}(z)$ denotes the normal cone to $\mathcal{X}$ at z. Taking $z = x$ and $y = x - \alpha[\nabla f(x) + J_c(x)^\top u]$ gives $-\alpha[\nabla f(x) + J_c(x)^\top u] \in N_{\mathcal{X}}(x)$, i.e., $0 \in \nabla f(x) + J_c(x)^\top u + N_{\mathcal{X}}(x)$. Thus the full KKT system holds.

(ii) Sufficiency. If (x,u) satisfies the KKT system, then Proposition 5.2 gives $d_{\boldsymbol{\rho}}(x,u) = 0$, and from equation 5 we have $\lambda_{\boldsymbol{\rho}}(x,u) = u$. The stationarity condition is equivalent to $x = \Pi_{\mathcal{X}}\big(x - \alpha[\nabla f(x) + J_c(x)^\top \lambda_{\boldsymbol{\rho}}(x,u)]\big)$. Hence $\mathcal{G}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) = 0$ and $\mathcal{R}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}(x,u) = 0$. □

These results identify the object controlled by the feedback loop: $d_{\boldsymbol{\rho}}$ controls feasibility and complementarity, while $\mathcal{G}^{\mathcal{X}}_{\alpha,\boldsymbol{\rho}}$ controls primal stationarity. This is exactly the residual-level interpretation used by RCML: the multiplier memory is not driven by an arbitrary error signal, but by a projected-pressure residual whose zero set is aligned with the KKT conditions.

5.2 Finite-Gain Convex-Affine Backbone

We now analyze the deterministic residual-integral backbone behind RESIDUAL-I and the finite-gain RCML update in Table 1. For the remainder of Sections 5.2 and 5.3, we analyze the isotropic specialization of the coordinate-wise pressure scale: $\boldsymbol{\rho} = \rho_0 \mathbf{1}$ with $\rho_0 > 0$. The symbol ρ_0 is used only in this isotropic analysis. This avoids mixing the vector-valued pressure scale used in Algorithm 1 with the scalar weight needed in the Lyapunov function. Under this specialization, $\boldsymbol{\rho} \odot (Ax - b) = \rho_0(Ax - b)$, and the projected pressure becomes $[u + \rho_0(Ax - b)]_+$.

Consider

$$\min_{x \in \mathbb{R}^d} f(x) \text{ s.t. } Ax - b \leq 0,$$

where f is continuously differentiable, μ-strongly convex, and has an L-Lipschitz gradient. Assume Slater's condition holds (Nocedal & Wright, 2006) and $u_0 \geq 0$. Let $x^\star$ be the unique primal optimizer and let $\mathcal{U}^\star$ be the set of KKT multipliers. For any vector v and closed set $\mathcal{S}$, we write $\operatorname{dist}(v, \mathcal{S}) = \inf_{w \in \mathcal{S}} \|v - w\|$.

Applying the template equation 3 with the signals from equation 5 yields the deterministic finite-gain backbone

$$\lambda_k = [u_k + \rho_0(Ax_k - b)]_+,\ d_k = \lambda_k - u_k,\ x_{k+1} = x_k - \alpha_k[\nabla f(x_k) + A^\top \lambda_k],\ u_{k+1} = u_k + \beta_k d_k. \tag{11}$$

Here λ_k and d_k are deterministic instances of equation 5 with $c(x) = Ax - b$ and $\boldsymbol{\rho} = \rho_0 \mathbf{1}$. The effective memory-tracking gain β_k corresponds to the product $\eta_k \kappa_{\mathrm{I}}$ in equation 6. We fix a gain ratio $\theta > 0$ and set $\beta_k = \theta \alpha_k$. Then $u_{k+1} = u_k + \theta \alpha_k d_k$, and the finite-gain condition $0 < \beta_k \leq 1$ becomes $\theta \alpha_k \leq 1$. Under this condition, the multiplier update can be rewritten as $u_{k+1} = (1 - \beta_k)u_k + \beta_k \lambda_k \in \mathbb{R}^m_+$.

Lemma 5.4 (Weighted summability and slow variation)**.** *Let $\{a_k\}$ be a nonnegative sequence, $\alpha_k > 0$, $\sum_k \alpha_k = \infty$, and $\alpha_k \to 0$. If $\sum_k \alpha_k a_k < \infty$ and $|a_{k+1} - a_k| \leq C\alpha_k$, then $a_k \to 0$.*

Proof. Suppose, for contradiction, that $a_k \not\to 0$. Then there exist $\varepsilon > 0$ and a subsequence $\{k_j\}$ with $a_{k_j} \geq \varepsilon$ for all j. From $|a_{k+1} - a_k| \leq C\alpha_k$, for any $k > k_j$ we have $|a_k - a_{k_j}| \leq C \sum_{t=k_j}^{k-1} \alpha_t$. Let $\delta = \varepsilon/(2C)$. Because $\sum_k \alpha_k = \infty$ and $\alpha_k \to 0$, for each sufficiently large k_j we can find a forward index $\ell_j > k_j$ such that

$$\sum_{t=k_j}^{\ell_j} \alpha_t \geq \delta, \quad \sum_{t=k_j}^{\ell_j - 1} \alpha_t < \delta.$$

For every $k \in [k_j, \ell_j]$, the total step mass is at most δ, hence $|a_k - a_{k_j}| \leq C\delta = \varepsilon/2$, and therefore $a_k \geq \varepsilon/2$. The α-mass of the window $[k_j, \ell_j]$ is at least δ. Since $\sum_k \alpha_k = \infty$ and $\alpha_k \to 0$, we can select a subsequence of disjoint such windows whose total α-mass is infinite. Consequently,

$$\sum_k \alpha_k a_k \geq \sum_j \frac{\varepsilon}{2} \sum_{t=k_j}^{\ell_j} \alpha_t \geq \sum_j \frac{\varepsilon}{2}\, \delta = \infty,$$

contradicting the hypothesis $\sum_k \alpha_k a_k < \infty$. Hence $a_k \to 0$. □

Theorem 5.5 (Finite-gain convergence of the residual-integral backbone)**.** *Under the assumptions $0 < \alpha_k \leq \bar{\alpha}$, $0 < \beta_k \leq 1$, $\sum_k \alpha_k = \infty$, and $\sum_k \alpha_k^2 < \infty$, for every $u^\star \in \mathcal{U}^\star$ the sequence generated by equation 11 satisfies*

$$x_k \to x^\star, \quad d_k \to 0, \quad \nabla f(x_k) + A^\top \lambda_k \to 0,$$

and consequently

$$\operatorname{dist}(u_k, \mathcal{U}^\star) \to 0, \quad \operatorname{dist}(\lambda_k, \mathcal{U}^\star) \to 0.$$

If $\mathcal{U}^\star = \{u^\star\}$ is a singleton, then $u_k \to u^\star$ and $\lambda_k \to u^\star$.

Proof. Step 1: weighted Lyapunov function. For any $u^\star \in \mathcal{U}^\star$, define

$$V_k^\theta(u^\star) := \|x_k - x^\star\|^2 + (\theta\rho_0)^{-1}\|u_k - u^\star\|^2.$$

The weight $(\theta\rho_0)^{-1}$ matches the gain ratio $\beta_k/\alpha_k = \theta$. Let $G_k = \nabla f(x_k) + A^\top \lambda_k$.

Step 2: primal descent expansion. The primal update in equation 11 gives

$$\|x_{k+1} - x^\star\|^2 = \|x_k - x^\star - \alpha_k G_k\|^2 = \|x_k - x^\star\|^2 - 2\alpha_k \langle x_k - x^\star, G_k\rangle + \alpha_k^2 \|G_k\|^2.$$

Since $u^\star \in \mathcal{U}^\star$, KKT stationarity gives $\nabla f(x^\star) + A^\top u^\star = 0$. Hence

$$\langle x_k - x^\star, G_k\rangle = \langle x_k - x^\star, \nabla f(x_k) - \nabla f(x^\star)\rangle + \langle A(x_k - x^\star), \lambda_k - u^\star\rangle.$$

By μ-strong convexity of f, $\langle x_k - x^\star, \nabla f(x_k) - \nabla f(x^\star)\rangle \geq \mu\|x_k - x^\star\|^2$. Therefore

$$\|x_{k+1} - x^\star\|^2 \leq \ \|x_k - x^\star\|^2 - 2\alpha_k\mu\|x_k - x^\star\|^2 - 2\alpha_k\langle A(x_k - x^\star), \lambda_k - u^\star\rangle + \alpha_k^2\|G_k\|^2.$$

Step 3: multiplier expansion and projection inequality. From equation 11, the multiplier update is $u_{k+1} = u_k + \beta_k d_k$. Hence

$$\|u_{k+1} - u^\star\|^2 = \|u_k - u^\star\|^2 + 2\beta_k\langle u_k - u^\star, d_k\rangle + \beta_k^2\|d_k\|^2.$$

Using $d_k = \lambda_k - u_k$, we have $u_k - u^\star = \lambda_k - u^\star - d_k$, so that

$$\langle u_k - u^\star, d_k\rangle = \langle \lambda_k - u^\star, d_k\rangle - \|d_k\|^2.$$

To bound $\langle \lambda_k - u^\star, d_k\rangle$, we use the firm nonexpansiveness of Euclidean projection onto a closed convex set (Bauschke & Combettes, 2017). Applying this property to $\lambda_k = [u_k + \rho_0(Ax_k - b)]_+$ and the KKT identity $u^\star = [u^\star + \rho_0(Ax^\star - b)]_+$ (which follows from Proposition 5.2) yields

$$\|\lambda_k - u^\star\|^2 \le \big\langle \lambda_k - u^\star,\ (u_k + \rho_0(Ax_k - b)) - (u^\star + \rho_0(Ax^\star - b))\big\rangle.$$

Expanding the right-hand side via $u_k - u^\star = \lambda_k - u^\star - d_k$ gives

$$\|\lambda_k - u^\star\|^2 \le \|\lambda_k - u^\star\|^2 - \langle \lambda_k - u^\star, d_k\rangle + \rho_0\langle \lambda_k - u^\star, A(x_k - x^\star)\rangle.$$

Canceling $\|\lambda_k - u^\star\|^2$ from both sides, we obtain

$$\langle \lambda_k - u^\star, d_k\rangle \le \rho_0\langle \lambda_k - u^\star, A(x_k - x^\star)\rangle.$$

Substituting this into the expression for $\langle u_k - u^\star, d_k\rangle$ gives

$$\langle u_k - u^\star, d_k\rangle \le \rho_0\langle \lambda_k - u^\star, A(x_k - x^\star)\rangle - \|d_k\|^2.$$

Step 4: cross-term cancellation. Multiply the multiplier expansion by $(\theta\rho_0)^{-1}$ and use $\beta_k = \theta\alpha_k$:

$$(\theta\rho_0)^{-1}\|u_{k+1} - u^\star\|^2 \le\ (\theta\rho_0)^{-1}\|u_k - u^\star\|^2 + 2\alpha_k\langle \lambda_k - u^\star, A(x_k - x^\star)\rangle - \frac{2\alpha_k}{\rho_0}\|d_k\|^2 + \frac{\theta\alpha_k^2}{\rho_0}\|d_k\|^2.$$

Adding this to the primal descent inequality cancels the cross term $\langle A(x_k - x^\star), \lambda_k - u^\star\rangle$ and yields

$$V_{k+1}^\theta(u^\star) \le\ V_k^\theta(u^\star) - 2\alpha_k\mu\|x_k - x^\star\|^2 - \frac{2\alpha_k}{\rho_0}\|d_k\|^2 + \alpha_k^2\|G_k\|^2 + \frac{\theta\alpha_k^2}{\rho_0}\|d_k\|^2.$$

Step 5: bounding the quadratic terms. From the definition of λ_k and the firm nonexpansiveness of the projection,

$$\|\lambda_k - u^\star\| \le \|u_k - u^\star + \rho_0 A(x_k - x^\star)\| \le \|u_k - u^\star\| + \rho_0\|A\|\|x_k - x^\star\|.$$

Using $G_k = \nabla f(x_k) - \nabla f(x^\star) + A^\top(\lambda_k - u^\star)$ and the Lipschitz continuity of ∇f,

$$\|G_k\| \le L\|x_k - x^\star\| + \|A\|\|\lambda_k - u^\star\| \le (L + \rho_0\|A\|^2)\|x_k - x^\star\| + \|A\|\|u_k - u^\star\|.$$

Thus

$$\|G_k\|^2 \le 2(L + \rho_0\|A\|^2)^2\|x_k - x^\star\|^2 + 2\|A\|^2\|u_k - u^\star\|^2.$$

From the definition of $V_k^\theta(u^\star)$, $\|x_k - x^\star\|^2 \le V_k^\theta(u^\star)$ and $\|u_k - u^\star\|^2 \le \theta\rho_0 V_k^\theta(u^\star)$. Hence

$$\|G_k\|^2 \le C_G V_k^\theta(u^\star),$$

with $C_G = 2(L + \rho_0\|A\|^2)^2 + 2\theta\rho_0\|A\|^2$.

For the $\|d_k\|^2$ terms, note that $\theta\alpha_k \le 1$ implies $\theta\alpha_k^2 \le \alpha_k$. Therefore

$$-\frac{2\alpha_k}{\rho_0}\|d_k\|^2 + \frac{\theta\alpha_k^2}{\rho_0}\|d_k\|^2 = -\frac{\alpha_k}{\rho_0}\|d_k\|^2 - \frac{\alpha_k(1 - \theta\alpha_k)}{\rho_0}\|d_k\|^2 \le -\frac{\alpha_k}{\rho_0}\|d_k\|^2.$$

Substituting these bounds into the Lyapunov recursion gives

$$V_{k+1}^{\theta}(u^\star) \le \big(1 + C_G\alpha_k^2\big)V_k^{\theta}(u^\star) - 2\alpha_k\mu\|x_k - x^\star\|^2 - \rho_0^{-1}\alpha_k\|d_k\|^2. \tag{12}$$

Step 6: convergence of x_k, d_k, and G_k. Since $\sum_k \alpha_k^2 < \infty$, the Robbins–Siegmund almost-supermartingale lemma (Robbins & Siegmund, 1971) applied to equation 12 guarantees that $V_k^{\theta}(u^\star)$ remains bounded and

$$\sum_{k=0}^{\infty} \alpha_k\|x_k - x^\star\|^2 < \infty, \quad \sum_{k=0}^{\infty} \alpha_k\|d_k\|^2 < \infty.$$

Boundedness of V_k^{θ} implies boundedness of $\{x_k\}$, $\{u_k\}$, $\{\lambda_k\}$, $\{d_k\}$, and $\{G_k\}$. The increments satisfy

$$\|x_{k+1} - x_k\| = \alpha_k\|G_k\| = O(\alpha_k), \quad \|u_{k+1} - u_k\| = \theta\alpha_k\|d_k\| = O(\alpha_k).$$

To bound $\|d_{k+1} - d_k\|$, observe from equation 5 that $d_{\rho_0}(x, u) = [u + \rho_0(Ax - b)]_+ - u$. The projection $[\cdot]_+$ is firmly nonexpansive, hence 1-Lipschitz. For any (x, u) and (x', u'),

$$\|d(x, u) - d(x', u')\| \le 2\|u - u'\| + \rho_0\|A\|\|x - x'\|.$$

Thus $\|d_{k+1} - d_k\| = O(\alpha_k)$ on the bounded trajectory. Similarly,

$$\big|\|x_{k+1} - x^\star\|^2 - \|x_k - x^\star\|^2\big| \le \big(\|x_{k+1} - x^\star\| + \|x_k - x^\star\|\big)\,\|x_{k+1} - x_k\| = O(\alpha_k).$$

Applying Lemma 5.4 first to $a_k = \|x_k - x^\star\|^2$ and then to $a_k = \|d_k\|^2$, together with the slow-variation bounds above, yields

$$\|x_k - x^\star\|^2 \to 0, \quad \|d_k\|^2 \to 0.$$

Therefore $x_k \to x^\star$ and $d_k \to 0$.

It remains to prove $G_k \to 0$. Since ∇f is L-Lipschitz and the trajectory is bounded, there exists $C_G' > 0$ such that

$$\|G_{k+1} - G_k\| \le C_G'\big(\|x_{k+1} - x_k\| + \|u_{k+1} - u_k\|\big) \le C_G'\alpha_k.$$

Suppose $G_k \not\to 0$. Then there exist $\varepsilon > 0$ and infinitely many indices k_j with $\|G_{k_j}\| \ge 4\varepsilon$. Set $v_j = G_{k_j}/\|G_{k_j}\|$ and $\Delta = \varepsilon/(2C_G')$. Since $\sum_k \alpha_k = \infty$ and $\alpha_k \to 0$, for each sufficiently large k_j we can find $\ell_j \ge k_j$ such that $\sum_{k=k_j}^{\ell_j} \alpha_k \ge \Delta$ and $\sum_{k=k_j}^{\ell_j-1} \alpha_k < \Delta$. For $k \in [k_j, \ell_j]$, the slow variation of G_k yields $\|G_k - G_{k_j}\| \le C_G' \sum_{t=k_j}^{k-1} \alpha_t \le \varepsilon$, hence $\langle G_k, v_j\rangle \ge 3\varepsilon$. Using $x_{k+1} = x_k - \alpha_k G_k$, we get $\langle x_{\ell_j+1} - x_{k_j}, v_j\rangle = -\sum_{k=k_j}^{\ell_j} \alpha_k\langle G_k, v_j\rangle \le -3\varepsilon\Delta$. Thus $\|x_{\ell_j+1} - x_{k_j}\| \ge 3\varepsilon\Delta$, contradicting $x_k \to x^\star$. Therefore $G_k \to 0$.

Finally, let $u_{k_j} \to \bar{u}$ be any cluster point of $\{u_k\}$. Since $d_k = \lambda_k - u_k \to 0$, we also have $\lambda_{k_j} \to \bar{u}$. Taking limits in $G_{k_j} = \nabla f(x_{k_j}) + A^\top\lambda_{k_j} \to 0$ gives $\nabla f(x^\star) + A^\top\bar{u} = 0$. From $d_k \to 0$ and $x_k \to x^\star$ we obtain $\bar{u} = [\bar{u} + \rho_0(Ax^\star - b)]_+$. By Proposition 5.2, $\bar{u} \ge 0$, $Ax^\star - b \le 0$, and $\bar{u}_i(Ax^\star - b)_i = 0$ for all i. Thus every cluster point belongs to $\mathcal{U}^\star$. Since $\{u_k\}$ is bounded and all its cluster points lie in the closed set $\mathcal{U}^\star$, we conclude $\operatorname{dist}(u_k, \mathcal{U}^\star) \to 0$. Because $\lambda_k - u_k = d_k \to 0$, the same holds for λ_k. If $\mathcal{U}^\star = \{u^\star\}$, pointwise convergence follows. □

The theorem shows that full projected multiplier replacement is not required for convergence. Finite-gain residual tracking preserves the primal–dual cancellation once the multiplier energy is weighted by $(\theta\rho_0)^{-1}$. This explains why the gain ratio $\theta = \beta_k/\alpha_k$ is not merely a tuning parameter; it determines the correct relative scale between primal descent and multiplier memory tracking.

Remark 5.6 (Time-varying gain ratios)**.** The proof above uses a fixed gain ratio $\theta = \beta_k/\alpha_k$. The same argument can be extended to a slowly varying ratio $\theta_k = \beta_k/\alpha_k$, provided that

$$0 < \underline{\theta} \le \theta_k \le \overline{\theta} < \infty, \quad 0 < \beta_k \le 1, \quad \sum_{k=0}^{\infty} |\theta_{k+1}^{-1} - \theta_k^{-1}| < \infty.$$

In that case, one uses the time-varying Lyapunov function $V_k^{\theta_k} = \|x_k - x^\star\|^2 + (\theta_k\rho_0)^{-1}\|u_k - u^\star\|^2$. After the same primal–dual cancellation, the only additional term is $\rho_0^{-1}(\theta_{k+1}^{-1} - \theta_k^{-1})\|u_{k+1} - u^\star\|^2$. Taking the positive part and using the uniform bounds on θ_k, this term is controlled by a summable multiplicative perturbation of the Lyapunov recursion. Hence the Robbins–Siegmund almost-supermartingale argument (Robbins & Siegmund, 1971) remains valid. The summability condition on $|\theta_{k+1}^{-1} - \theta_k^{-1}|$ is essential; without it, the variation of the Lyapunov weight may accumulate and destroy the quasi-Fejér descent structure (Combettes, 2001).

5.3 Stochastic Convex-Affine Residual Bound

We now quantify how stochastic mini-batch feedback perturbs the finite-gain backbone. Throughout this subsection, the pressure scale is the isotropic scalar $\rho_0 > 0$ and the constraint is $c(x) = Ax - b$. The analysis follows the standard stochastic-approximation convention of representing mini-batch quantities as conditionally unbiased estimates plus martingale-difference noise (Robbins & Monro, 1951; Kushner & Yin, 2003; Borkar, 2008; Bottou et al., 2018). Specifically, assume $\hat{g}_k = \nabla f(x_k) + \zeta_k,\ \hat{c}_k = Ax_k - b + \varepsilon_k$, where $\mathcal{F}_k$ is the sigma-field generated by all randomness before sampling the mini-batch at iteration k. The noise variables satisfy $\mathbb{E}[\zeta_k \mid \mathcal{F}_k] = 0,\ \mathbb{E}[\|\zeta_k\|^2 \mid \mathcal{F}_k] \le \sigma_g^2/B_k,\ \mathbb{E}[\varepsilon_k \mid \mathcal{F}_k] = 0,\ \mathbb{E}[\|\varepsilon_k\|^2 \mid \mathcal{F}_k] \le \sigma_c^2/B_k$. Here $\zeta_k \in \mathbb{R}^d$ denotes the mini-batch gradient noise, $\varepsilon_k \in \mathbb{R}^m$ denotes the mini-batch constraint-estimation noise, B_k is the batch size at iteration k, and σ_g, σ_c are uniform conditional second-moment constants. The $1/B_k$ scaling corresponds to mini-batch averaging under bounded single-sample second moments; the theorem only uses the displayed conditional moment bounds, not independence across different iterations. The stochastic pressure and residual are $\hat{\lambda}_k = [u_k + \rho_0\hat{c}_k]_+,\ \hat{d}_k = \hat{\lambda}_k - u_k$, and the stochastic update is $x_{k+1} = x_k - \alpha_k[\hat{g}_k + A^\top\hat{\lambda}_k],\ u_{k+1} = u_k + \theta\alpha_k\hat{d}_k$.

Let $\lambda_k = [u_k + \rho_0(Ax_k - b)]_+,\ d_k = \lambda_k - u_k,\ p_k = \hat{\lambda}_k - \lambda_k$. By firm nonexpansiveness of the projection, $\|p_k\| \le \rho_0\|\varepsilon_k\|,\ \ \mathbb{E}[\|p_k\|^2 \mid \mathcal{F}_k] \le \rho_0^2\sigma_c^2/B_k$.

Throughout the proof, $C, C_0, C_1, \ldots$ denote finite positive constants whose values may change from line to line. Constants denoted by $K_1, \ldots, K_4$ in the theorem statement are horizon-independent constants after summation. Their allowed dependence is stated explicitly in the theorem. To avoid imposing global boundedness as an unverified assumption, we state the result for a stopped process. For $R > 0$, define $\tau_R = \inf\{k \ge 0 : \|x_k - x^\star\| + \|u_k - u^\star\| > R\}$. The stopped iterates are $x_k^R := x_{k\wedge\tau_R}$ and $u_k^R := u_{k\wedge\tau_R}$. Equivalently, the stopped process follows the original stochastic update on $\{k < \tau_R\}$ and has zero increment on $\{k \ge \tau_R\}$. In the following, we omit the superscript R for readability, but all conditional recursions are understood in this stopped sense. If the implementation uses compact primal projection and bounded dual clipping, then $\tau_R = \infty$ for a sufficiently large R. The following stopped finite-horizon bound is a residual version of the standard almost-supermartingale and stochastic-approximation argument used for noisy gradient-type recursions (Robbins & Siegmund, 1971; Kushner & Yin, 2003; Borkar, 2008).

Theorem 5.7 (Stopped stochastic weighted residual bound)**.** *Suppose the problem assumptions of Theorem 5.5 hold for the convex-affine backbone, and let $u^\star \in \mathcal{U}^\star$ be fixed. For a finite horizon T, assume $0 < \theta\alpha_k \le 1$ for $k = 0, \ldots, T-1$. Then there exist constants $K_1, K_2, K_3, K_4 > 0$, depending on μ, L, $\|A\|$, ρ_0, θ, and R, but not on T, B_k, or the noise realization, such that*

$$\begin{aligned}\frac{\sum_{k=0}^{T-1}\alpha_k\,\mathbb{E}\big[\mathbf{1}_{\{k<\tau_R\}}(\|x_k - x^\star\|^2 + \|d_k\|^2)\big]}{\sum_{k=0}^{T-1}\alpha_k} &\le \frac{K_1V_0^\theta + K_2\sum_{k=0}^{T-1}\alpha_k^2}{\sum_{k=0}^{T-1}\alpha_k}\\ &+ K_3\,\frac{\sum_{k=0}^{T-1}\alpha_k\,\rho_0\sigma_c/\sqrt{B_k}}{\sum_{k=0}^{T-1}\alpha_k} + K_4\,\frac{\sum_{k=0}^{T-1}\alpha_k^2\,(\sigma_g^2 + \rho_0^2\sigma_c^2)/B_k}{\sum_{k=0}^{T-1}\alpha_k}.\end{aligned} \tag{13}$$

In particular, if $\alpha_k = T^{-1/2}$ and $B_k \equiv B$, then

$$T^{-1}\sum_{k=0}^{T-1}\mathbb{E}\big[\mathbf{1}_{\{k<\tau_R\}}(\|x_k - x^\star\|^2 + \|d_k\|^2)\big] \le O(T^{-1/2}) + O(\rho_0\sigma_c/\sqrt{B}) + O\big((\sigma_g^2 + \rho_0^2\sigma_c^2)/(B\sqrt{T})\big).$$

Proof. Step 1: stochastic primal expansion. Let $\hat{G}_k = \hat{g}_k + A^\top \hat{\lambda}_k$. Since $\hat{g}_k = \nabla f(x_k) + \zeta_k$ and $\hat{\lambda}_k = \lambda_k + p_k$, we have $\hat{G}_k = G_k + \zeta_k + A^\top p_k$ with $G_k = \nabla f(x_k) + A^\top \lambda_k$. The primal update gives

$$\begin{aligned}\|x_{k+1} - x^\star\|^2 &= \|x_k - x^\star - \alpha_k \hat{G}_k\|^2 \\ &= \|x_k - x^\star\|^2 - 2\alpha_k \langle x_k - x^\star, G_k\rangle - 2\alpha_k \langle x_k - x^\star, \zeta_k + A^\top p_k\rangle + \alpha_k^2 \|\hat{G}_k\|^2.\end{aligned}$$

The G_k-term yields the same deterministic descent and cross term as in Theorem 5.5. The gradient noise satisfies $\mathbb{E}[\langle x_k - x^\star, \zeta_k\rangle \mid \mathcal{F}_k] = 0$. On the stopped region, $\|x_k - x^\star\| \le R$, so $2\alpha_k |\langle x_k - x^\star, A^\top p_k\rangle| \le 2\alpha_k R \|A\| \|p_k\|$.

Step 2: stochastic multiplier expansion. The stochastic multiplier update is $u_{k+1} = u_k + \theta\alpha_k(d_k + p_k)$. Expanding the weighted multiplier energy gives

$$(\theta\rho_0)^{-1}\|u_{k+1} - u^\star\|^2 = (\theta\rho_0)^{-1}\|u_k - u^\star\|^2 + \frac{2\alpha_k}{\rho_0}\langle u_k - u^\star, d_k\rangle + \frac{2\alpha_k}{\rho_0}\langle u_k - u^\star, p_k\rangle + \frac{\theta\alpha_k^2}{\rho_0}\|d_k + p_k\|^2.$$

The d_k-term is handled by the same projection inequality as in Theorem 5.5, which cancels the primal cross term. The pressure-noise term is bounded on the stopped region by $2\alpha_k\rho_0^{-1}|\langle u_k - u^\star, p_k\rangle| \le 2\alpha_k R\rho_0^{-1}\|p_k\|$.

Step 3: conditional Lyapunov recursion. Adding the primal and multiplier expansions, the deterministic cross terms cancel. The remaining quadratic terms are bounded as follows. Using $\hat{G}_k = G_k + \zeta_k + A^\top p_k$,

$$\|\hat{G}_k\|^2 \le 3\|G_k\|^2 + 3\|\zeta_k\|^2 + 3\|A\|^2\|p_k\|^2,$$

and $\|d_k + p_k\|^2 \le 2\|d_k\|^2 + 2\|p_k\|^2$. On the stopped region, V_k^θ is bounded by a constant depending on R, and hence $\|G_k\|$ and $\|d_k\|$ are bounded. Taking conditional expectations and using Young's inequality together with the martingale-difference moment bounds on ζ_k and p_k yields

$$\mathbb{E}\big[\|\hat{G}_k\|^2 + \|d_k + p_k\|^2 \mid \mathcal{F}_k\big] \le C(1 + V_k^\theta) + C(\sigma_g^2 + \rho_0^2\sigma_c^2)/B_k$$

for some constant $C > 0$. Therefore

$$\begin{aligned}\mathbb{E}[V_{k+1}^\theta \mid \mathcal{F}_k] \le{}& (1 + C_0\alpha_k^2)V_k^\theta - C_1\alpha_k(\|x_k - x^\star\|^2 + \|d_k\|^2) \\ &+ C_2\alpha_k\,\mathbb{E}[\|p_k\| \mid \mathcal{F}_k] + C_3\alpha_k^2(\sigma_g^2 + \rho_0^2\sigma_c^2)/B_k + C_4\alpha_k^2.\end{aligned}$$

By firm nonexpansiveness of Euclidean projection (Bauschke & Combettes, 2017) and Jensen's inequality, $\mathbb{E}[\|p_k\| \mid \mathcal{F}_k] \le (\mathbb{E}[\|p_k\|^2 \mid \mathcal{F}_k])^{1/2} \le \rho_0\sigma_c/\sqrt{B_k}$.

Step 4: stopped summation. By the stopped-process construction, the update is applied only on $\{k < \tau_R\}$ and the increment is zero on $\{k \ge \tau_R\}$. Hence the conditional recursion becomes

$$\begin{aligned}\mathbb{E}[V_{k+1}^\theta \mid \mathcal{F}_k] \le V_k^\theta + \mathbf{1}_{\{k<\tau_R\}}\Big[&C_0\alpha_k^2 V_k^\theta - C_1\alpha_k(\|x_k - x^\star\|^2 + \|d_k\|^2) \\ &+ C_2\alpha_k\rho_0\sigma_c/\sqrt{B_k} + C_3\alpha_k^2(\sigma_g^2 + \rho_0^2\sigma_c^2)/B_k + C_4\alpha_k^2\Big].\end{aligned}$$

Taking unconditional expectations,

$$\begin{aligned}\mathbb{E}[V_{k+1}^\theta] \le{}& \mathbb{E}[V_k^\theta] - C_1\alpha_k\,\mathbb{E}\big[\mathbf{1}_{\{k<\tau_R\}}(\|x_k - x^\star\|^2 + \|d_k\|^2)\big] \\ &+ C_0\alpha_k^2\,\mathbb{E}[\mathbf{1}_{\{k<\tau_R\}}V_k^\theta] + C_2\alpha_k\rho_0\sigma_c/\sqrt{B_k} + C_3\alpha_k^2(\sigma_g^2 + \rho_0^2\sigma_c^2)/B_k + C_4\alpha_k^2.\end{aligned}$$

On the stopped region, V_k^θ is bounded, so the term $C_0\alpha_k^2\,\mathbb{E}[\mathbf{1}_{\{k<\tau_R\}}V_k^\theta]$ is absorbed into $C\alpha_k^2$. Summing from $k = 0$ to $T - 1$ gives

$$\begin{aligned}\sum_{k=0}^{T-1}\alpha_k\,\mathbb{E}\big[\mathbf{1}_{\{k<\tau_R\}}(\|x_k - x^\star\|^2 + \|d_k\|^2)\big] \le{}& K_1V_0^\theta + K_2\sum_{k=0}^{T-1}\alpha_k^2 + K_3\sum_{k=0}^{T-1}\alpha_k\rho_0\sigma_c/\sqrt{B_k} \\ &+ K_4\sum_{k=0}^{T-1}\alpha_k^2(\sigma_g^2 + \rho_0^2\sigma_c^2)/B_k.\end{aligned}$$

Dividing by $\sum_{k=0}^{T-1} \alpha_k$ yields equation 13. This is the usual finite-horizon weighted-average consequence of a one-step almost-supermartingale recursion (Robbins & Siegmund, 1971; Kushner & Yin, 2003). The stated rate follows by substituting $\alpha_k = T^{-1/2}$ and $B_k \equiv B$. □

The bound contains a non-vanishing pressure-noise floor $O(\rho_0 \sigma_c / \sqrt{B})$ when the batch size B is fixed. Thus, under only second-moment mini-batch noise assumptions, the stochastic iterates are controlled in a residual neighborhood rather than guaranteed to converge exactly to the KKT set. Exact residual convergence can be recovered by increasing B_k so that $\rho_0 \sigma_c / \sqrt{B_k} \to 0$, or by using a vanishing effective pressure scale when such a schedule is compatible with the desired constraint accuracy. In the present algorithm, ρ_0 is kept bounded away from zero to maintain feasibility pressure, so the practical mechanisms for reducing this neighborhood are variance reduction, larger batches, and filtering.

Even if $\hat{c}_k$ is an unbiased estimate of $c(x_k)$, the projection $[\cdot]_+$ is nonlinear:

$$\mathbb{E}[\hat{c}_k \mid \mathcal{F}_k] = c(x_k) \quad \not\Rightarrow \quad \mathbb{E}\big[[u_k + \rho_0 \hat{c}_k]_+ \mid \mathcal{F}_k\big] = [u_k + \rho_0 c(x_k)]_+.$$

Therefore the main stochastic theorem uses the conservative first-moment pressure bound. A sharper $O(1/B)$ pressure-noise floor is available only when the trajectory is locally away from the nonsmooth projection boundary.

Proposition 5.8 (Local second-moment refinement away from projection kinks)**.** *Suppose that, on the stopped region, there exists a margin $\delta > 0$ such that $|u_{k,i} + \rho_0 c_i(x_k)| \geq \delta$ for every coordinate i. Assume also that $\|\rho_0 \varepsilon_k\|_\infty \leq \delta/2$ almost surely. Then the active pattern of the projection is unchanged by the stochastic perturbation, and there exists an $\mathcal{F}_k$-measurable diagonal matrix D_k with diagonal entries in $\{0, 1\}$ such that $p_k = D_k \rho_0 \varepsilon_k$. Consequently,*

$$\mathbb{E}[p_k \mid \mathcal{F}_k] = 0, \quad \mathbb{E}[\|p_k\|^2 \mid \mathcal{F}_k] \leq \rho_0^2 \sigma_c^2 / B_k.$$

Under this local active-pattern stability condition, the first-moment pressure term in equation 13 disappears, and the pressure-noise contribution is of order $O(\rho_0^2 \sigma_c^2 / B)$ for fixed B.

Proof. For each coordinate i, define $a_{k,i} = u_{k,i} + \rho_0 c_i(x_k)$. The margin assumption gives $|a_{k,i}| \geq \delta$, and the bounded-noise assumption gives $|\rho_0 \varepsilon_{k,i}| \leq \delta/2$. Therefore $a_{k,i}$ and $a_{k,i} + \rho_0 \varepsilon_{k,i}$ have the same sign. If $a_{k,i} > 0$, then $p_{k,i} = [a_{k,i} + \rho_0 \varepsilon_{k,i}]_+ - [a_{k,i}]_+ = \rho_0 \varepsilon_{k,i}$. If $a_{k,i} < 0$, then $p_{k,i} = 0$. Thus $p_k = D_k \rho_0 \varepsilon_k$, where D_k selects the locally active pressure coordinates. Since D_k is $\mathcal{F}_k$-measurable and $\mathbb{E}[\varepsilon_k \mid \mathcal{F}_k] = 0$, we obtain $\mathbb{E}[p_k \mid \mathcal{F}_k] = 0$ and $\mathbb{E}[\|p_k\|^2 \mid \mathcal{F}_k] = \rho_0^2 \, \mathbb{E}[\|D_k \varepsilon_k\|^2 \mid \mathcal{F}_k] \leq \rho_0^2 \sigma_c^2 / B_k$. Substituting this into the proof of Theorem 5.7 removes the first-moment pressure-bias term and leaves only the second-moment pressure term. □

5.4 Filtering as a Noise–Lag Tradeoff

The stochastic bound above shows that constraint noise enters mainly through the pressure channel. Filtering acts before pressure formation and trades variance reduction for tracking lag. We continue to work under the convex-affine setting $c(x) = Ax - b$ with isotropic pressure scale ρ_0. Using the filtered signal defined in equation 7, write $\hat{c}_k = c(x_k) + \varepsilon_k$, $e_k = \bar{c}_k - c(x_k)$. Where e_k is the filtering error, $\bar{c}_k$ is the exponentially smoothed constraint signal, and γ is the smoothing coefficient. We also use $\alpha_{\max} := \sup_k \alpha_k$ and a stopped-region bound M_G on the second moment of the stochastic primal direction. Assume c is L_c-Lipschitz with $L_c = \|A\|$ (since $c(x) = Ax - b$), $\mathbb{E}[\varepsilon_k \mid \mathcal{F}_k] = 0$, $\mathbb{E}[\|\varepsilon_k\|^2 \mid \mathcal{F}_k] \leq \sigma_c^2 / B$. For the stopped process inSection 5.3, the stochastic primal direction satisfies $\mathbb{E}[\|\hat{G}_k\|^2 \mid \mathcal{F}_k] \leq M_G^2$. This holds because $\hat{G}_k = G_k + \zeta_k + A^\top p_k$, and on the stopped region every component has bounded second moment by the definitions of ζ_k, p_k, and the boundedness of G_k (which follows from the boundedness of V_k^θ as shown in Theorem 5.5). The following bound is a direct EMA bias–variance decomposition under a martingale-difference noise model, as commonly used in stochastic approximation and stochastic-gradient analyses (Kushner & Yin, 2003; Borkar, 2008; Bottou et al., 2018).

Proposition 5.9 (Filtering error bound in expectation)**.** *For constant $\gamma \in (0, 1]$, the filtering error satisfies*

$$\mathbb{E}\|e_k\|^2 \leq C(1-\gamma)^{2k} \|e_0\|^2 + C L_c^2 M_G^2 \alpha_{\max}^2 / \gamma^2 + C \gamma \sigma_c^2 / B. \tag{14}$$

Moreover,

$$\|[u_k + \rho_0 \bar{c}_k]_+ - [u_k + \rho_0 c(x_k)]_+\| \le \rho_0 \|e_k\|,$$

so the filtering error enters the stochastic residual bound through the pressure channel.

Proof. From equation 7, $\bar{c}_k = (1-\gamma)\bar{c}_{k-1} + \gamma(c(x_k) + \varepsilon_k)$. Subtracting $c(x_k)$ and adding and subtracting $c(x_{k-1})$ gives

$$e_k = (1-\gamma)e_{k-1} + (1-\gamma)(c(x_{k-1}) - c(x_k)) + \gamma\varepsilon_k.$$

Unrolling the recursion,

$$e_k = (1-\gamma)^k e_0 + \sum_{t=1}^{k}(1-\gamma)^{k-t+1}(c(x_{t-1}) - c(x_t)) + \gamma\sum_{t=1}^{k}(1-\gamma)^{k-t}\varepsilon_t.$$

The initial term contributes $(1-\gamma)^{2k}\|e_0\|^2$. For the lag term, Minkowski's inequality gives

$$\big(\mathbb{E}\|\text{lag}\|^2\big)^{1/2} \le \sum_{t=1}^{k}(1-\gamma)^{k-t+1}\big(\mathbb{E}\|c(x_{t-1}) - c(x_t)\|^2\big)^{1/2}.$$

By Lipschitz continuity and the bound $\|x_t - x_{t-1}\| \le \alpha_{t-1}\|\hat{G}_{t-1}\|$ (from the primal update with nonexpansive projection), we have $\big(\mathbb{E}\|x_t - x_{t-1}\|^2\big)^{1/2} \le \alpha_{t-1}M_G \le \alpha_{\max}M_G$. Hence

$$\big(\mathbb{E}\|\text{lag}\|^2\big)^{1/2} \le L_c M_G \alpha_{\max}\sum_{j=0}^{k-1}(1-\gamma)^{j+1} \le L_c M_G \alpha_{\max}/\gamma,$$

and the squared lag contribution is bounded by $L_c^2 M_G^2 \alpha_{\max}^2/\gamma^2$.

For the stochastic noise term, the martingale-difference property gives

$$\mathbb{E}\Big\|\gamma\sum_{t=1}^{k}(1-\gamma)^{k-t}\varepsilon_t\Big\|^2 = \gamma^2\sum_{t=1}^{k}(1-\gamma)^{2(k-t)}\mathbb{E}\|\varepsilon_t\|^2 \le \frac{\gamma}{2-\gamma}\,\frac{\sigma_c^2}{B}.$$

Combining the three bounds and absorbing numerical constants yields equation 14. The last inequality follows directly from the firm nonexpansiveness of the projection $[\cdot]_+$. □

The bound has a direct interpretation. A smaller γ reduces the variance term $O(\gamma\sigma_c^2/B)$, but increases the lag term $O(\alpha_{\max}^2/\gamma^2)$. Thus filtering is a noise–lag tradeoff rather than a uniformly monotone feasibility improvement. This matches the algorithmic role described in Section 4.3: filtering stabilizes the measurement channel before pressure formation, but excessive smoothing can delay the response to changing constraints.

5.5 Adaptive Scaling and Dynamic Correction as Feedback-Loop Perturbations

Adaptive scaling and dynamic correction modify different channels of the residual-controlled feedback loop. Adaptive scaling changes the map from the constraint signal to the projected pressure, while residual-νPI correction changes the memory-feedback signal after the residual has been formed. This subsection does not claim a separate convergence theorem for these optional modules. Instead, it gives a perturbation accounting result showing how such modules enter the Lyapunov recursion of the convex-affine backbone.

Adaptive scaling as a pressure and residual perturbation In the implementation, the adaptive scale is computed from the clipped second-moment rule in equation 8. For the present perturbation analysis, it is enough to regard $\boldsymbol{\rho}_k = (\rho_{k,1}, \ldots, \rho_{k,m})^\top \in \mathbb{R}^m_{++}$ as a clipped coordinatewise scale. Let $\rho_0\mathbf{1}$ be the reference isotropic scale used in the convex-affine backbone. For a filtered constraint signal $\bar{c}_k$, define

$$\lambda_k^{\rho_0} = [u_k + \rho_0\bar{c}_k]_+, \quad d_k^{\rho_0} = \lambda_k^{\rho_0} - u_k, \quad \lambda_k^{\boldsymbol{\rho}_k} = [u_k + \boldsymbol{\rho}_k \odot \bar{c}_k]_+, \quad d_k^{\boldsymbol{\rho}_k} = \lambda_k^{\boldsymbol{\rho}_k} - u_k.$$

The scale-induced perturbation is $p_k^\rho := \lambda_k^{\boldsymbol{\rho}_k} - \lambda_k^{\rho_0}$. Since both residuals are formed by subtracting the same stored multiplier u_k, the same perturbation also appears in the memory residual: $d_k^{\boldsymbol{\rho}_k} = d_k^{\rho_0} + p_k^\rho$. Thus adaptive scaling should be understood as a perturbation of both the primal pressure channel and the memory-residual channel.

Proposition 5.10 (Adaptive scaling as a projected-pressure perturbation)**.** *The scale-induced perturbation satisfies*

$$\|p_k^\rho\| \le \|(\boldsymbol{\rho}_k - \rho_0 \mathbf{1}) \odot \bar{c}_k\|.$$

Consequently, if $\boldsymbol{\rho}_k$ is clipped and $\bar{c}_k$ is bounded, then adaptive scaling contributes a bounded perturbation to both the projected pressure and the residual. If

$$\frac{\sum_{k=0}^{T-1} \alpha_k \, \|(\boldsymbol{\rho}_k - \rho_0 \mathbf{1}) \odot \bar{c}_k\|^2}{\sum_{k=0}^{T-1} \alpha_k} \to 0 \quad \text{as } T \to \infty,$$

then the adaptive-scaling perturbation becomes negligible in the averaged residual bound.

Proof. By firm nonexpansiveness of the projection $[\cdot]_+$,

$$\|p_k^\rho\| = \big\|[u_k + \boldsymbol{\rho}_k \odot \bar{c}_k]_+ - [u_k + \rho_0 \bar{c}_k]_+\big\| \le \|(\boldsymbol{\rho}_k - \rho_0 \mathbf{1}) \odot \bar{c}_k\|.$$

If $\boldsymbol{\rho}_k \in [\rho_{\min}, \rho_{\max}]^m$ and $\bar{c}_k$ is bounded, the right-hand side is bounded. The last statement follows directly from the same inequality after multiplying by α_k, summing, and normalizing by $\sum_{k=0}^{T-1} \alpha_k$. □

Dynamic residual correction as a memory-feedback perturbation Residual-νPI correction acts after the residual has been computed. Using equation 9 and equation 10, the memory signal can be written as

$$s_k = \kappa_{\mathrm{I}} d_k + \kappa_{\mathrm{P}}(\xi_k - \xi_{k-1}) = \kappa_{\mathrm{I}}(d_k + q_k), \quad q_k := (\kappa_{\mathrm{P}}/\kappa_{\mathrm{I}})(\xi_k - \xi_{k-1}).$$

The correction q_k is a memory-feedback perturbation. It does not change the target residual equation $d_k = 0$, but the instantaneous condition $d_k = 0$ alone does not imply $s_k = 0$ unless the auxiliary state has also settled. The preserved equilibrium is the augmented one: $d_k = 0$, $\xi_{k-1} = 0$, under which $\xi_k = 0$, $q_k = 0$, and $s_k = 0$. Therefore, the dynamic correction should be interpreted as a transient feedback modification rather than as a change of the target KKT residual.

Generic inexact residual loop To account for stochastic gradients, pressure perturbations, residual perturbations, and dynamic correction in a unified way, let λ_k and d_k denote the reference isotropic backbone signals with scale ρ_0. Consider the inexact loop

$$\tilde{\lambda}_k = \lambda_k + p_k^\lambda, \quad \tilde{d}_k = d_k + p_k^d + q_k,$$
$$x_{k+1} = x_k - \alpha_k\big[\nabla f(x_k) + A^\top \tilde{\lambda}_k + r_k\big], \quad u_{k+1} = u_k + \theta \alpha_k \tilde{d}_k.$$

Here p_k^λ is the perturbation entering the primal pressure channel, p_k^d is the perturbation entering the residual used by the memory update, q_k is the additional memory-feedback correction induced by the residual-νPI term, and r_k is the primal-gradient perturbation. For perturbations generated by stochastic projection noise or adaptive scaling, one typically has $p_k^d = p_k^\lambda$, because the residual is defined as the projected pressure minus u_k. We keep the two symbols separate to allow more general inexact implementations. This separation follows the standard inexact fixed-point/perturbed stochastic approximation viewpoint (Kushner & Yin, 2003; Borkar, 2008; Bauschke & Combettes, 2017).

The next statement is a quasi-Fejér perturbation accounting result: summable first-order errors and summable second-order errors preserve the residual limit of the reference recursion (Robbins & Siegmund, 1971; Bauschke & Combettes, 2017).

Proposition 5.11 (Perturbation accounting for the inexact residual loop)**.** *On a stopped Lyapunov region, the inexact residual loop satisfies*

$$\begin{aligned} V_{k+1}^\theta \le (1 + C\alpha_k^2) V_k^\theta - C_1 \alpha_k (\|x_k - x^\star\|^2 + \|d_k\|^2) + C_2 \alpha_k \big(\|p_k^\lambda\| + \|p_k^d\| + \|q_k\| + \|r_k\|\big) \\ + C_3 \alpha_k^2 \big(\|p_k^\lambda\|^2 + \|p_k^d\|^2 + \|q_k\|^2 + \|r_k\|^2\big) + C_4 \alpha_k^2, \end{aligned}$$

where the constants depend on the stopped-region radius and the problem parameters, but not on k. Consequently, if

$$\sum_k \alpha_k\big(\|p_k^\lambda\| + \|p_k^d\| + \|q_k\| + \|r_k\|\big) < \infty$$

and the corresponding second-order perturbation terms are summable, then the residuals still satisfy $x_k \to x^\star$, $d_k \to 0$, and $G_k \to 0$ as in Theorem 5.5.

Proof. Repeat the proof of Theorem 5.5 with λ_k replaced by $\tilde{\lambda}_k = \lambda_k + p_k^\lambda$, d_k replaced by $\tilde{d}_k = d_k + p_k^d + q_k$, and G_k replaced by $G_k + A^\top p_k^\lambda + r_k$, where $G_k = \nabla f(x_k) + A^\top \lambda_k$. The deterministic part gives the same primal descent, multiplier expansion, projection inequality, and cross-term cancellation as in the reference backbone.

The additional primal perturbation terms are

$$-2\alpha_k\langle x_k - x^\star, A^\top p_k^\lambda + r_k\rangle + \alpha_k^2\|G_k + A^\top p_k^\lambda + r_k\|^2 - \alpha_k^2\|G_k\|^2.$$

On the stopped region, $\|x_k - x^\star\| \le R$, so $2\alpha_k|\langle x_k - x^\star, A^\top p_k^\lambda + r_k\rangle| \le 2\alpha_k R(\|A\|\|p_k^\lambda\| + \|r_k\|)$. The difference of the quadratic terms is bounded by $C\alpha_k^2(V_k^\theta + \|p_k^\lambda\|^2 + \|r_k\|^2)$, where the V_k^θ term is absorbed into the usual $(1 + C\alpha_k^2)V_k^\theta$ factor.

For the multiplier update, set $e_k := p_k^d + q_k$. Then $u_{k+1} = u_k + \theta\alpha_k(d_k + e_k)$. Compared with the deterministic multiplier expansion, the additional terms in the weighted multiplier energy are

$$\frac{2\alpha_k}{\rho_0}\langle u_k - u^\star, e_k\rangle + \frac{\theta\alpha_k^2}{\rho_0}\big(\|d_k + e_k\|^2 - \|d_k\|^2\big).$$

The first term cannot be absorbed into the deterministic descent because the Lyapunov recursion contains no negative term proportional to $\|u_k - u^\star\|^2$. Using the stopped-region bound $\|u_k - u^\star\| \le R$, $2\alpha_k\rho_0^{-1}|\langle u_k - u^\star, e_k\rangle| \le 2\alpha_k R\rho_0^{-1}(\|p_k^d\| + \|q_k\|)$. For the second term, $\|d_k + e_k\|^2 - \|d_k\|^2 = 2\langle d_k, e_k\rangle + \|e_k\|^2$. Since d_k is bounded on the stopped region, $\alpha_k^2|\|d_k + e_k\|^2 - \|d_k\|^2| \le C\alpha_k^2(\|p_k^d\| + \|q_k\| + \|p_k^d\|^2 + \|q_k\|^2)$. Because $\alpha_k \le \bar{\alpha}$, the linear $\alpha_k^2(\|p_k^d\| + \|q_k\|)$ term is dominated by the first-order perturbation term $C\alpha_k(\|p_k^d\| + \|q_k\|)$.

Combining all bounds with the deterministic descent recursion yields the stated inequality. If the first-order perturbations are α_k-summable and the second-order perturbations are summable, the same quasi-Fejér argument (Bauschke & Combettes, 2017) used in Theorem 5.5 implies that the additional inexact channels do not alter the asymptotic convergence of the residuals. □

The proposition is an accounting result, not a standalone convergence theorem for adaptive scaling or residual-νPI correction. It identifies the type of perturbation control that would be sufficient to preserve the deterministic finite-gain residual limit. If the perturbations are merely bounded but not summable, the same inequality should instead be interpreted as predicting a residual neighborhood whose size depends on the accumulated perturbation level.

Remark on local nonconvex geometry and convergence The analysis above focuses on the convex-affine backbone, where the projected-pressure residual dynamics admit a clean Lyapunov cancellation. For general nonlinear or nonconvex constrained problems, the same residual map

$$F_{\boldsymbol{\rho}}(x, u) = [\nabla f(x) + J_c(x)^\top \lambda_{\boldsymbol{\rho}}(x, u), \quad d_{\boldsymbol{\rho}}(x, u)]^\top$$

still has a meaningful local interpretation. Near a regular KKT point satisfying LICQ, strict complementarity, and the second-order sufficient condition (Bonnans & Shapiro, 2000; Facchinei & Pang, 2003), the active set is locally stable and the projected pressure map becomes piecewise smooth with a fixed local pattern. In such a neighborhood, $d_{\boldsymbol{\rho}}(x, u)$ is locally equivalent to the feasibility–complementarity residual, and $F_{\boldsymbol{\rho}}(x, u)$ is locally Lipschitz equivalent to the standard KKT residual after active-set identification. Thus, under a local error bound or metric subregularity condition (Rockafellar & Wets, 1998; Dontchev & Rockafellar, 2014), small $\|F_{\boldsymbol{\rho}}(x, u)\|$ implies a small local KKT residual.

A full nonconvex convergence theory would require additional ingredients beyond the scope of this paper. One possible route is to establish metric subregularity of the KKT mapping around a regular solution and then prove that the residual-controlled dynamics enter and remain in the corresponding neighborhood. Another route is to combine the residual feedback law with a merit function, trust-region mechanism, or line-search globalization scheme so that descent and feasibility control can be enforced before local regularity becomes valid. For stochastic nonconvex problems, one would additionally need to control mini-batch noise, active-set switching, and the bias introduced by the projected pressure near the kink set. These requirements are substantially stronger than the convex-affine assumptions used in Theorem 5.5 and Theorem 5.7.

Therefore, we do not claim global convergence for general nonconvex constrained learning problems. Instead, the role of the nonconvex experiments is empirical: they test whether the residual-controlled multiplier dynamics provide stable feasibility control and useful multiplier-memory behavior in settings where the convex-affine theory no longer applies directly. The reported residual, feasibility, and objective trajectories should be interpreted as evidence of practical stability rather than as a substitute for a full nonconvex convergence theorem.

5.6 Summary of the Theoretical Findings

The analysis establishes four conclusions. First, $d_{\boldsymbol{\rho}}(x, u)$ is an exact feasibility–complementarity residual induced by the inequality augmented Lagrangian equation 4, and together with projected stationarity it characterizes the KKT system. Second, the convex-affine backbone equation 11 converges under finite-gain tracking $\beta_k = \theta\alpha_k$, with

$$x_k \to x^\star, \quad d_k \to 0, \quad \nabla f(x_k) + A^\top \lambda_k \to 0, \quad \operatorname{dist}(u_k, \mathcal{U}^\star) \to 0.$$

Third, the stopped stochastic analysis gives the weighted residual bound equation 13. For $\alpha_k = T^{-1/2}$ and fixed batch size B, the conservative finite-time behavior is

$$O(T^{-1/2}) + O(\rho_0 \sigma_c / \sqrt{B}),$$

which indicates a residual neighborhood under fixed mini-batch noise. A sharper $O(\rho_0^2 \sigma_c^2 / B)$ pressure-noise floor is obtained only under a local active-pattern stability condition that keeps the projected pressure away from its kink set. Fourth, filtering, adaptive pressure scaling, and residual-νPI correction act as perturbations of the measurement, pressure, and memory-feedback channels, respectively. In particular, filtering reduces pressure variance at the cost of lag, adaptive scaling changes the pressure gain, and dynamic residual correction modifies the transient memory feedback.

6 Experiments

6.1 Experimental roadmap

The experiments are designed to evaluate the residual-controlled multiplier framework from four complementary perspectives. First, we test whether the pressure-memory residual provides the intended signal-level behavior and whether finite-gain memory tracking stabilizes multiplier dynamics. Second, we examine whether the framework improves tolerance-aware decision quality under stochastic reserve constraints. Third, we evaluate reliable solution discovery in a nonconvex stochastic pricing-inventory problem with multiple random initializations. Fourth, we test whether the same multiplier-control mechanism can be integrated into neural fair-ranking training under exposure constraints.

Each experiment isolates a specific role of the framework: projected pressure formation, finite-gain residual memory, measurement filtering, adaptive pressure scaling, and dynamic memory-feedback correction. Detailed data generation procedures, hardware, hyperparameters, validation grids, sample sizes, and evaluation windows are reported in B.

Unless otherwise stated, all reported values are averaged over random seeds, and tail metrics are computed over the final evaluation window of each run. All multiplier-based methods are implemented under the stochastic primal–dual interface in equation 3. They differ in how they construct the effective pressure, the pressure-memory residual, and the memory feedback signal.

6.2 Compared Methods and Metrics

We compare RCML variants with unconstrained training and fixed penalty baselines (Nocedal & Wright, 2006), raw violation-based stochastic primal–dual updates including SGDA-Signed and SGDA-Positive (Xu, 2020; Jin & Wang, 2022), projected augmented-Lagrangian replacement Projected-ALM (Bertsekas, 1982), constraint extrapolation ConEx (Boob et al., 2023), perturbed primal–dual ascent GDPA (Lu, 2022; Alacaoglu & Wright, 2024), stochastic SQP StoSQP (Curtis et al., 2024; Fang et al., 2024), and stochastic barrier baselines StochBarrier (Dimitrieski et al., 2025). The main residual-controlled methods are Residual-I, RCML-Core, RCML-Adaptive, and RCML-Robust. Diagnostic variants such as Adaptive-Proj., RCML-Adaptive-τ, and Residual-νPI are used only in ablation studies.

Objective, cost, profit, and NDCG measure primal decision quality, where NDCG is used for ranking evaluation (Järvelin & Kekäläinen, 2002). Constraint violation, tolerance excess, feasibility rate, exposure violation, and reliable rate measure constraint control, where exposure violation is used for exposure-based fair-ranking constraints (Singh & Joachims, 2018). Dual total variation measures multiplier-memory movement, and residual total variation measures fluctuation of the pressure-memory residual. Runtime measures the computational overhead of the residual-controlled update.

6.3 Experiment 1: Signal Diagnostics and Module Ablation

Experiment 1 verifies the residual-controlled multiplier interface through signal-level diagnostics and module-level ablations. The signal-level diagnostic uses a scalar four-phase constraint sequence to test whether different multiplier signals distinguish violation activation, stale-memory release, and inactive dead-zone behavior. The module-level ablation then evaluates the contribution of projected pressure formation, finite-gain residual memory, constraint filtering, and adaptive coordinatewise pressure scaling.

The ablation benchmark contains stochastic LP, convex QP, and mildly nonconvex QP tasks. We evaluate stationary feedback, high-noise feedback, and heterogeneous-scale constraints in the main text. The active-set switching diagnostic is reported in B.5 because it is used only to evaluate optional transient extensions.

Signal-level validation We first use a four-phase scalar constraint sequence with inactive, violated, release, and inactive phases. The compared signals are

$$s_k^{\text{signed}} = \hat{c}_k, \quad s_k^{\text{positive}} = [\hat{c}_k]_+, \quad s_k^{\text{residual}} = [u_k + \rho_0 \hat{c}_k]_+ - u_k,$$

and a filtered residual signal $\bar{c}_k = (1-\gamma)\bar{c}_{k-1} + \gamma \hat{c}_k$, $s_k^{\text{filtered}} = [u_k + \rho_0 \bar{c}_k]_+ - u_k$. All signals are inserted into the same multiplier recursion $u_{k+1} = [u_k + \eta s_k]_+$. Figure 1 shows that the signed signal can decrease obsolete multiplier mass but also leaks negative control in inactive regions. The positive signal avoids negative leakage but does not actively release obsolete multiplier memory. The residual signal releases stored multiplier mass when $u_k > 0$ and becomes zero after the multiplier reaches the inactive dead zone. The filtered residual further suppresses near-boundary noise, with a visible response delay.

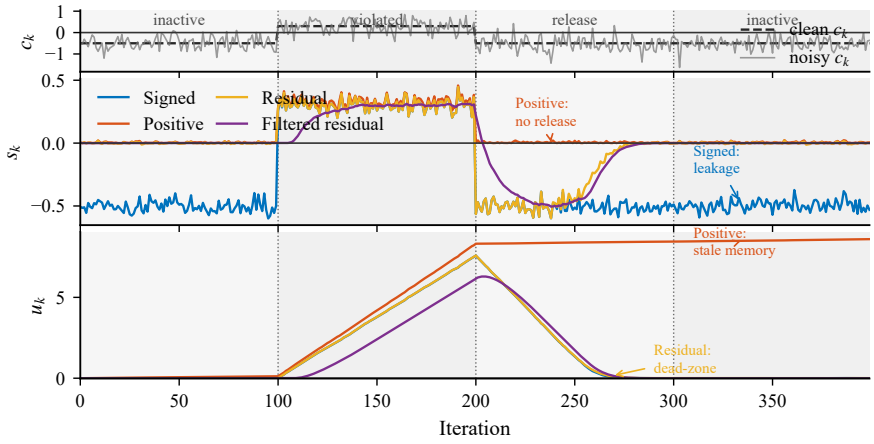

Figure 1: Signal-level validation of the residual multiplier interface. The top panel shows the scalar constraint state c_k, the middle panel shows the control signal s_k, and the bottom panel shows the multiplier state u_k. The residual signal distinguishes feasible-but-releasing states from genuinely inactive dead-zone states.

Core ablation The core ablation compares raw violation-based updates, projected-pressure replacement, finite-gain residual memory, and adaptive pressure scaling.

Table 2: Core signal and memory ablation on stationary stochastic LP, QP, and mildly nonconvex QP tasks. RelRate denotes the fraction of tail iterates with violation below 5×10^{-2}. Lower is better for ObjTail, ViolTail, DualTV, ResidualTV, and runtime; higher is better for RelRate.

Problem	Method	ObjTail	ViolTail	RelRate	DualTV	ResidualTV	Runtime
LP	SGDA-Signed	-1.040×10^{1}	5.989×10^{-1}	0.000	2.261×10^{-3}	2.188×10^{0}	6.39×10^{-2}
LP	SGDA-Positive	-7.166×10^{0}	2.201×10^{0}	0.000	2.994×10^{-3}	9.535×10^{-1}	6.04×10^{-2}
LP	Projected-ALM	-1.045×10^{1}	3.715×10^{-2}	0.644	6.909×10^{-1}	1.027×10^{0}	6.00×10^{-2}
LP	Residual-I	-1.052×10^{1}	4.339×10^{-2}	0.610	3.533×10^{-3}	9.712×10^{-1}	6.21×10^{-2}
LP	RCML-Adaptive	-1.035×10^{1}	3.489×10^{-2}	0.704	6.469×10^{-3}	1.265×10^{0}	6.89×10^{-2}
QP	SGDA-Signed	7.161×10^{-1}	4.844×10^{-1}	0.000	1.299×10^{-3}	6.999×10^{-1}	7.24×10^{-2}
QP	SGDA-Positive	1.578×10^{1}	1.315×10^{-1}	0.166	6.881×10^{-4}	3.642×10^{-1}	7.23×10^{-2}
QP	Projected-ALM	7.893×10^{-1}	2.330×10^{-2}	0.832	1.862×10^{-1}	2.717×10^{-1}	7.25×10^{-2}
QP	Residual-I	7.734×10^{-1}	1.547×10^{-2}	0.982	1.717×10^{-3}	2.715×10^{-1}	7.39×10^{-2}
QP	RCML-Adaptive	7.824×10^{-1}	2.094×10^{-2}	0.886	3.515×10^{-3}	4.134×10^{-1}	8.01×10^{-2}
NCVQP	SGDA-Signed	-8.062×10^{-2}	2.447×10^{-1}	0.000	4.765×10^{-4}	5.588×10^{-1}	7.15×10^{-2}
NCVQP	SGDA-Positive	3.105×10^{0}	9.024×10^{-1}	0.000	1.375×10^{-3}	4.746×10^{-1}	7.26×10^{-2}
NCVQP	Projected-ALM	-8.840×10^{-2}	1.442×10^{-3}	1.000	1.042×10^{-1}	1.561×10^{-1}	7.44×10^{-2}
NCVQP	Residual-I	-4.810×10^{-2}	2.269×10^{-3}	1.000	5.898×10^{-4}	1.447×10^{-1}	7.34×10^{-2}
NCVQP	RCML-Adaptive	-6.371×10^{-2}	1.835×10^{-3}	1.000	1.077×10^{-3}	1.938×10^{-1}	7.87×10^{-2}

Table 2 gives three observations. First, raw violation-based updates have poor reliability in all three tasks. Second, projected replacement improves feasibility but produces much larger multiplier variation. Third, replacing full replacement with finite-gain residual memory keeps feasibility in the same range while substantially reducing DualTV. The adaptive variant improves reliability in LP, remains competitive in QP, and preserves full reliability in NCVQP.

High-noise feedback The high-noise setting evaluates how filtering changes the measurement channel.

Table 3: High-noise module ablation. High-noise uses sparse constraint mini-batches with additive observation noise. ViolP95 is the tail 95-th percentile violation.

Problem	Method	ViolTail	ViolP95	RelRate	DualTV	ResidualTV	Runtime
LP	Projected-ALM	0.000	0.000	1.000	1.909×10^{0}	3.022×10^{0}	1.39×10^{-1}
LP	Adaptive-Proj.	1.517×10^{-2}	1.236×10^{-1}	0.903	1.077×10^{0}	1.138×10^{0}	1.58×10^{-1}
LP	Residual-I	0.000	0.000	1.000	4.724×10^{-3}	2.422×10^{0}	1.43×10^{-1}
LP	RCML-Core	5.802×10^{-2}	1.967×10^{-1}	0.539	3.039×10^{-3}	9.806×10^{-1}	1.46×10^{-1}
LP	RCML-Adaptive	1.863×10^{-2}	1.143×10^{-1}	0.855	4.242×10^{-3}	1.438×10^{0}	1.60×10^{-1}
QP	Projected-ALM	1.231×10^{-4}	0.000	1.000	5.098×10^{-1}	8.075×10^{-1}	1.94×10^{-1}
QP	Adaptive-Proj.	4.949×10^{-3}	3.179×10^{-2}	0.965	4.004×10^{-1}	4.825×10^{-1}	2.20×10^{-1}
QP	Residual-I	1.044×10^{-4}	0.000	1.000	2.331×10^{-3}	6.703×10^{-1}	2.02×10^{-1}
QP	RCML-Core	2.982×10^{-2}	1.011×10^{-1}	0.740	1.380×10^{-3}	2.394×10^{-1}	1.93×10^{-1}
QP	RCML-Adaptive	5.812×10^{-3}	4.555×10^{-2}	0.959	2.893×10^{-3}	5.785×10^{-1}	2.15×10^{-1}
NCVQP	Projected-ALM	0.000	0.000	1.000	3.739×10^{-1}	6.179×10^{-1}	2.67×10^{-1}
NCVQP	Adaptive-Proj.	0.000	0.000	1.000	3.017×10^{-1}	3.982×10^{-1}	2.86×10^{-1}
NCVQP	Residual-I	0.000	0.000	1.000	9.441×10^{-4}	4.416×10^{-1}	2.51×10^{-1}
NCVQP	RCML-Core	1.229×10^{-5}	0.000	1.000	5.771×10^{-4}	1.725×10^{-1}	2.50×10^{-1}
NCVQP	RCML-Adaptive	0.000	0.000	1.000	1.026×10^{-3}	3.495×10^{-1}	2.72×10^{-1}

Table 3 shows that filtering reduces residual variation but may increase violation because of response lag. For instance, in high-noise QP, RCML-Core has lower ResidualTV than Residual-I, while its violation is larger. In NCVQP, all projected-pressure and residual variants satisfy the reliability criterion, and the main difference is variation in multiplier and residual signals.

Heterogeneous-scale constraints The heterogeneous-scale setting evaluates adaptive coordinatewise pressure scaling.

Table 4: Heterogeneous-scale module ablation. Constraint channels are multiplied by unequal positive scales. ViolP95 is the tail 95-th percentile violation.

Problem	Method	ViolTail	ViolP95	RelRate	DualTV	ResidualTV	Runtime
LP	Adaptive-Proj.	2.524×10^{-2}	1.335×10^{-1}	0.884	6.307×10^{-1}	7.454×10^{-1}	1.20×10^{-1}
LP	Residual-I	4.158×10^{-2}	6.722×10^{-2}	0.653	3.817×10^{-3}	1.264×10^{0}	1.15×10^{-1}
LP	RCML-Core	1.233×10^{-1}	2.556×10^{-1}	0.006	2.693×10^{-3}	5.692×10^{-1}	1.09×10^{-1}
LP	RCML-Adaptive	3.183×10^{-2}	1.436×10^{-1}	0.810	4.393×10^{-3}	9.805×10^{-1}	1.20×10^{-1}
QP	Adaptive-Proj.	1.239×10^{-2}	4.622×10^{-2}	0.953	3.203×10^{-1}	3.390×10^{-1}	1.68×10^{-1}
QP	Residual-I	2.817×10^{-2}	7.310×10^{-2}	0.910	1.554×10^{-3}	2.864×10^{-1}	1.58×10^{-1}
QP	RCML-Core	4.139×10^{-2}	8.928×10^{-2}	0.703	9.301×10^{-4}	1.085×10^{-1}	1.59×10^{-1}
QP	RCML-Adaptive	1.613×10^{-2}	5.855×10^{-2}	0.937	3.283×10^{-3}	4.106×10^{-1}	1.59×10^{-1}
NCVQP	Adaptive-Proj.	1.039×10^{-3}	7.364×10^{-3}	1.000	2.331×10^{-1}	2.454×10^{-1}	1.51×10^{-1}
NCVQP	Residual-I	4.244×10^{-3}	2.325×10^{-2}	1.000	1.026×10^{-3}	1.722×10^{-1}	1.51×10^{-1}
NCVQP	RCML-Core	5.913×10^{-3}	2.918×10^{-2}	1.000	8.341×10^{-4}	5.835×10^{-2}	1.55×10^{-1}
NCVQP	RCML-Adaptive	2.988×10^{-3}	1.887×10^{-2}	1.000	1.689×10^{-3}	3.051×10^{-1}	1.59×10^{-1}

Table 4 shows that adaptive scaling improves constraint handling when channels have unequal scales. On LP and QP, RCML-Adaptive improves feasibility relative to RCML-Core. On NCVQP, all residual and projected variants reach full reliability, and the differences are mainly in violation magnitude and signal variation. These results indicate that adaptive scaling is useful under heterogeneous constraints, but it is not uniformly dominant across all task geometries.

6.4 Experiment 2: Tolerance-Aware Stochastic Energy-Reserve Allocation

Experiment 2 evaluates RCML on a tolerance-aware stochastic energy-reserve allocation task. The decision variable represents reserve allocation, and the objective is to reduce operating cost while satisfying stochastic reserve adequacy constraints. Unlike strict feasibility benchmarks, this task allows small reserve inadequacy within a prescribed engineering tolerance. The relevant goal is therefore to balance cost, tolerance feasibility, reserve over-allocation, and multiplier stability.

For each method, we report tail-averaged operating cost, maximum reserve inadequacy violation, tolerance excess, tolerance-feasible rate, tolerance-aware score, reserve over-allocation, and runtime. The tolerance-aware score is defined as

$$\mathrm{TA} = \mathrm{Cost} + \omega_{\mathrm{tol}}[\mathrm{Viol} - \delta_{\mathrm{tol}}]_+.$$

The RCML-Adaptive configuration is selected by a small validation sweep and then fixed for the main comparison. The sweep is reported in Appendix B.5.

Table 5: Tolerance-aware stochastic energy-reserve allocation results. Normal uses $B = 256$ and $\delta_{\mathrm{tol}} = 10^{-2}$; Stress uses $B = 96$ and $\delta_{\mathrm{tol}} = 2 \times 10^{-2}$.

Setting	Method	Cost	Viol.	TolExc	TolFeas	TA	OverRes.	Time
Normal	Primal-only	0.000	1.805	1.795	0.000	179.466	0.000	0.11
Normal	SGDA-Signed	5.454	3.018×10^{-2}	2.018×10^{-2}	0.000	7.472	1.133×10^{-1}	0.10
Normal	SGDA-Positive	17.808	0.000	0.000	1.000	17.808	2.176	0.11
Normal	Projected-ALM	5.445	1.684×10^{-2}	7.400×10^{-3}	0.187	6.185	1.073×10^{-1}	0.11
Normal	Residual-I	5.453	8.182×10^{-3}	6.659×10^{-4}	0.727	5.519	1.111×10^{-1}	0.11
Normal	StoSQP	5.669	8.523×10^{-3}	4.321×10^{-3}	0.680	6.102	1.640×10^{-1}	0.40
Normal	GDPA	**5.294**	4.680×10^{-2}	3.680×10^{-2}	0.000	8.975	8.270×10^{-2}	0.11
Normal	ConEx	5.457	1.833×10^{-2}	8.350×10^{-3}	0.013	6.292	1.130×10^{-1}	0.11
Normal	StochBarrier	5.853	1.126×10^{-3}	2.246×10^{-4}	0.967	5.875	2.097×10^{-1}	0.11
Normal	RCML-Adaptive	5.444	7.335×10^{-3}	$\mathbf{6.496\times10^{-4}}$	0.760	**5.509**	1.064×10^{-1}	0.13
Stress	Primal-only	0.000	1.856	1.836	0.000	183.608	0.000	0.08
Stress	SGDA-Signed	5.701	3.254×10^{-2}	1.423×10^{-2}	0.227	7.124	1.167×10^{-1}	0.08
Stress	SGDA-Positive	17.808	0.000	0.000	1.000	17.808	2.132	0.08
Stress	Projected-ALM	5.713	3.090×10^{-2}	1.323×10^{-2}	0.280	7.035	1.189×10^{-1}	0.08
Stress	Residual-I	5.692	$\mathbf{1.335\times10^{-2}}$	$\mathbf{3.887\times10^{-4}}$	0.867	**5.731**	1.137×10^{-1}	0.09
Stress	StoSQP	6.107	1.839×10^{-2}	8.186×10^{-3}	0.607	6.926	2.175×10^{-1}	0.38
Stress	GDPA	**5.539**	4.883×10^{-2}	2.883×10^{-2}	0.000	8.422	8.731×10^{-2}	0.09
Stress	ConEx	5.701	2.765×10^{-2}	1.013×10^{-2}	0.367	6.714	1.166×10^{-1}	0.09
Stress	StochBarrier	6.702	0.000	0.000	1.000	6.702	3.477×10^{-1}	0.09
Stress	RCML-Adaptive	5.695	1.447×10^{-2}	1.430×10^{-3}	0.727	5.838	1.138×10^{-1}	0.11

Table 5 shows distinct tolerance-aware behaviors. Primal-only has the lowest cost but violates reserve constraints. SGDA-Positive and StochBarrier achieve high feasibility but with larger over-reserve and higher cost. GDPA obtains the lowest cost in both regimes, but its tolerance-feasible rate is zero. In the normal regime, RCML-Adaptive gives the lowest tolerance-aware score among the nontrivial methods. In the stress regime, Residual-I gives the lowest tolerance-aware score, while RCML-Adaptive remains close with lower cost than conservative feasible baselines.

6.5 Experiment 3: Reliable Solution Discovery in Nonconvex Pricing-Inventory Allocation

Experiment 3 evaluates reliable solution discovery in a nonconvex stochastic pricing-inventory allocation problem. The decision variables are the inventory allocation x_i and price r_i for each item. Demand depends nonlinearly on price and stochastic demand shocks, and the constraint requires expected shortage to remain below a prescribed tolerance level.

The purpose of this experiment is to test whether each method can reliably discover admissible high-profit solutions under stochastic feedback and random initialization. For each method and scenario, we run multiple

seed–initialization pairs. A run is called reliable if its tail tolerance-feasible rate is at least 0.9. We report reliable rate, number of reliable runs, mean and best profit over reliable runs, reliable violation, reliable overstock, and all-run average profit.

Table 6: Reliable solution discovery in nonconvex stochastic pricing-inventory allocation. ReliableRate is the fraction of seed–initialization runs whose tail tolerance-feasible rate is at least 0.9. MeanRelProfit, BestRelProfit, RelViol., and RelOverstock are computed only over reliable runs. AllProfit is averaged over all runs, including unreliable ones.

Setting	Method	ReliableRate	Runs	MeanRelProfit	BestRelProfit	RelViol.	RelOverstock	AllProfit
Normal	Primal-only	0.000	0/25	–	–	–	–	50.0297
Normal	SGDA-Signed	0.920	23/25	50.0897	52.2016	7.865×10^{-3}	8.619×10^{-3}	50.0144
Normal	SGDA-Positive	0.920	23/25	50.0817	52.1975	7.480×10^{-3}	9.900×10^{-3}	50.0067
Normal	Projected-ALM	1.000	25/25	50.0060	52.1763	3.886×10^{-3}	8.155×10^{-3}	50.0060
Normal	Residual-I	1.000	25/25	50.0049	52.1782	2.011×10^{-3}	8.102×10^{-3}	50.0049
Normal	RCML-Adaptive	1.000	25/25	50.0058	52.1790	2.337×10^{-3}	$\mathbf{7.976\times10^{-3}}$	50.0058
Normal	StoSQP	0.200	5/25	49.7684	50.1769	7.085×10^{-3}	1.017×10^{-2}	49.9650
Normal	GDPA	0.600	15/25	50.1948	52.1967	9.589×10^{-3}	8.144×10^{-3}	50.0210
Normal	ConEx	0.920	23/25	50.0908	52.2007	7.899×10^{-3}	8.444×10^{-3}	50.0155
Normal	StochBarrier	1.000	25/25	46.7575	50.2614	2.228×10^{-3}	2.341×10^{-2}	46.7575
Stress	Primal-only	0.000	0/25	–	–	–	–	50.6661
Stress	SGDA-Signed	0.000	0/25	–	–	–	–	50.4946
Stress	SGDA-Positive	0.000	0/25	–	–	–	–	50.4849
Stress	Projected-ALM	1.000	25/25	48.3833	50.7371	8.883×10^{-3}	1.246×10^{-3}	48.3833
Stress	Residual-I	1.000	25/25	49.4066	51.7168	1.621×10^{-2}	5.550×10^{-5}	49.4066
Stress	RCML-Adaptive	1.000	25/25	**49.6109**	**51.8196**	1.833×10^{-2}	$\mathbf{2.222\times10^{-5}}$	49.6109
Stress	StoSQP	0.560	14/25	47.5305	49.4934	1.355×10^{-2}	5.530×10^{-3}	47.3098
Stress	GDPA	0.000	0/25	–	–	–	–	50.5476
Stress	ConEx	0.000	0/25	–	–	–	–	50.5015
Stress	StochBarrier	1.000	25/25	44.9962	47.5309	3.066×10^{-3}	1.187×10^{-2}	44.9962

Table 6 shows that the normal regime is less discriminative: several methods obtain high reliable rates. In the stress regime, raw violation-based methods, GDPA, and ConEx have zero reliable runs, whereas projected-pressure and residual-controlled methods remain reliable. Among the fully reliable methods, RCML-Adaptive has the highest mean reliable profit under stress, while StochBarrier is more conservative. These observations are empirical and do not imply global nonconvex convergence.

6.6 Experiment 4: Large-Scale Fair Ranking under Exposure Constraints

Experiment 4 evaluates whether residual-controlled multiplier updates can be used as an in-processing constrained training method for neural fair ranking. Each query contains candidate items with features and binary group attributes. A neural ranker assigns a score to each item and is trained with a ListNet-type ranking loss. The constraint controls the soft exposure gap between the two groups.

Each query contains a set of candidate items with feature vectors and group attributes. A neural ranker $s_\theta(q, i)$ assigns a score to each item. The ranking loss is denoted by $\mathcal{L}_{\text{rank}}(\theta)$. Let $E_0(\theta)$ and $E_1(\theta)$ denote the average soft exposure assigned to the two groups under a differentiable softmax ranking distribution. The soft exposure gap is $g(\theta) = |E_0(\theta) - E_1(\theta)|$. The fairness constraint is $c(\theta) = g(\theta) - \epsilon \leq 0$. In the reported experiment, evaluation uses an admissible tolerance band $g(\theta) \leq \epsilon_{\text{eval}} + \delta_{\text{tol}} = 0.015$. For residual-controlled methods, a stricter internal training threshold is used to absorb mini-batch noise, while all reported metrics are evaluated using the common admissible threshold.

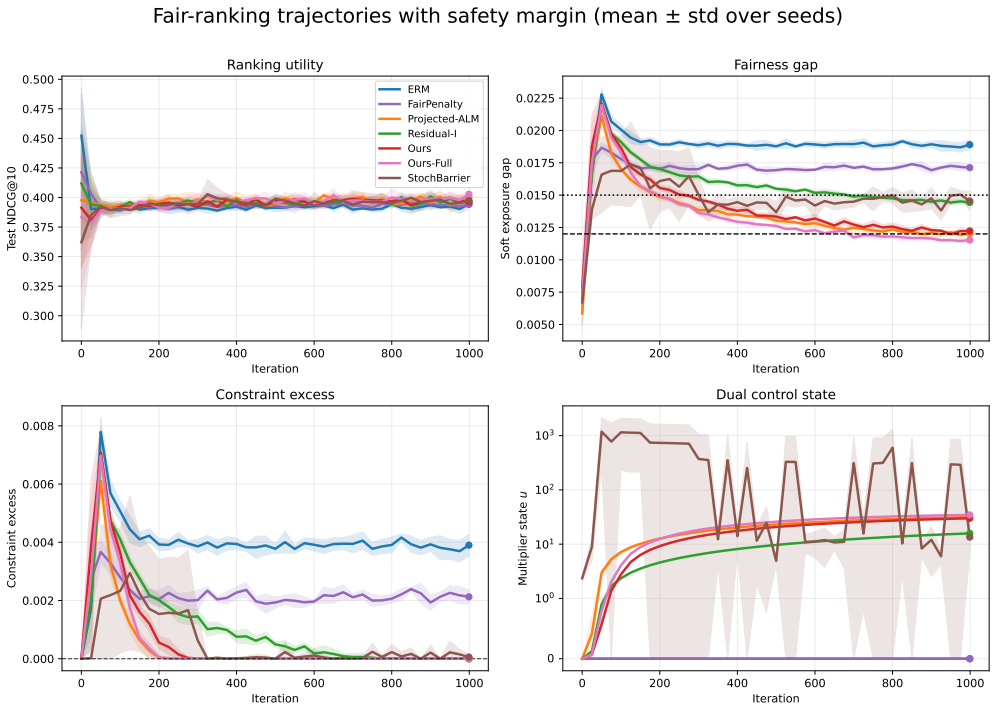

Figure 2: Large-scale fair-ranking training trajectories. Curves show mean ± standard deviation over random seeds. The dashed line in the fairness-gap panel denotes the target exposure threshold, and the dotted line denotes the admissible evaluation threshold 0.015. SGDA-Signed and SGDA-Positive are omitted from the trajectory plot for readability but are included in Table 7.

Table 7: Large-scale fair ranking under exposure constraints. Evaluation uses admissible soft exposure gap 0.015. All statistics are computed over random seeds at the final evaluation point.

Method	NDCG@10	Soft gap	TolExcess	Dual state u	Runtime (s)
ERM	0.3944 ± 0.0039	0.0189 ± 0.0004	$3.901\times10^{-3} \pm 3.672\times10^{-4}$	0.0000 ± 0.0000	7.64 ± 2.17
FairPenalty	0.3942 ± 0.0020	0.0171 ± 0.0002	$2.126\times10^{-3} \pm 2.398\times10^{-4}$	0.0000 ± 0.0000	8.67 ± 1.81
SGDA-Signed	0.3948 ± 0.0037	0.0183 ± 0.0001	$3.304\times10^{-3} \pm 1.263\times10^{-4}$	1.3558 ± 0.0100	7.98 ± 1.54
SGDA-Positive	0.3944 ± 0.0036	0.0182 ± 0.0001	$3.211\times10^{-3} \pm 1.308\times10^{-4}$	1.3435 ± 0.0076	7.96 ± 1.61
Projected-ALM	0.4002 ± 0.0025	0.0120 ± 0.0001	$0.000\times10^{0} \pm 0.000\times10^{0}$	31.5923 ± 0.1286	8.29 ± 1.76
Residual-I	0.3971 ± 0.0020	0.0144 ± 0.0001	$0.000\times10^{0} \pm 0.000\times10^{0}$	15.8298 ± 0.1247	8.66 ± 2.27
RCML-Adaptive	0.3962 ± 0.0043	0.0122 ± 0.0001	$0.000\times10^{0} \pm 0.000\times10^{0}$	19.9738 ± 0.1614	8.25 ± 2.72
RCML-Robust	$\mathbf{0.4028 \pm 0.0024}$	0.0115 ± 0.0003	$0.000\times10^{0} \pm 0.000\times10^{0}$	14.6091 ± 0.1905	8.29 ± 2.20
StochBarrier	0.3973 ± 0.0030	0.0145 ± 0.0007	$5.660\times10^{-5} \pm 9.761\times10^{-5}$	13.4958 ± 6.2623	8.94 ± 2.29

Figure 2 and Table 7 show that unconstrained and penalty-based training retain positive tolerance excess. The SGDA variants also remain outside the admissible exposure band. Projected-pressure and residual-controlled methods satisfy the admissible exposure tolerance. Among feasible methods, RCML-Robust gives the highest NDCG@10, while Residual-I uses a smaller dual state than Projected-ALM.

6.7 Summary of Empirical Findings

The experiments give four main observations. First, projected pressure formation improves feasibility relative to raw violation-based multiplier updates. Second, finite-gain residual memory substantially reduces multiplier variation compared with full projected replacement. Third, filtering and adaptive scaling act on different channels: filtering reduces measurement noise at the cost of lag, whereas adaptive scaling is useful under

heterogeneous constraint magnitudes. Fourth, in allocation, nonconvex discovery, and fair-ranking tasks, the residual-controlled variants avoid either infeasible low-cost solutions or overly conservative feasible solutions. These findings are empirical and should be read together with the theoretical scope in Section 5: the strict convergence result applies to the convex-affine backbone, while the optional stabilization modules are analyzed as perturbations of the backbone feedback loop.

7 Conclusion

This paper addresses the multiplier updating bottleneck in stochastic constrained optimization by proposing a residual control signal that separates instantaneous projected pressure from internal memory updates. Based on this signal, we construct the RCML algorithmic framework, establish its finite-gain convergence and stochastic bounds through theoretical analysis, and empirically validate its effectiveness. However, the proposed methodology has some limitations. First, the stochastic stabilization modules introduce additional hyperparameters, and the constraint filtering mechanism fundamentally trades tracking lag for variance reduction, which may lead to transient constraint breaches under rapid distribution shifts. Second, the algorithm lacks an active mechanism to avoid nonsmooth projection boundaries (kinks); thus, achieving the sharper $O(1/B)$ noise floor passively relies on the local stability of the active set. Future work could improve upon this by developing hyperparameter-free adaptive residual controllers or incorporating margin-aware mechanisms that actively stabilize the trajectory near nonsmooth boundaries without sacrificing response speed.

References

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A Reductions Approach to Fair Classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 60–69, Stockholm, Sweden, 2018.

Ahmet Alacaoglu and Stephen J. Wright. Complexity of Single Loop Algorithms for Nonlinear Programming with Stochastic Objective and Constraints. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pp. 4627–4635, Valencia, Spain, 2024.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

J. Frédéric Bonnans and Alexander Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, 2000.

Digvijay Boob, Qi Deng, and Guanghui Lan. Stochastic first-order methods for convex and nonconvex functional constrained optimization. *Mathematical Programming*, 197(1):215–279, 2023.

Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, Cambridge, 2008.

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

Vito Cerone, Sophie M. Fosson, Simone Pirrera, and Diego Regruto. A New Framework for Constrained Optimization via Feedback Control of Lagrange Multipliers. *IEEE Transactions on Automatic Control*, 70 (11):7141–7156, 2025.

Vito Cerone, Sophie M. Fosson, Simone Pirrera, Alessandro Re, and Diego Regruto. Feedback control of Lagrange multipliers for non-smooth constrained optimization. *arXiv preprint arXiv:2604.06511*, 2026.

Luiz F. O. Chamon and Alejandro Ribeiro. Probably Approximately Correct Constrained Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2020.

Ashish Cherukuri, Enrique Mallada, and Jorge Cortés. Asymptotic convergence of constrained primal-dual dynamics. *Systems & Control Letters*, 87:10–15, 2016.

Patrick L. Combettes. Quasi-Fejérian Analysis of Some Optimization algorithms. *Studies in Computational Mathematics*, 8:115–152, 2001.

Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training Well-Generalizing Classifiers for Fairness Metrics and Other Data-Dependent Constraints. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 1397–1405, Long Beach, California, USA, 2019a.

Andrew Cotter, Heinrich Jiang, Maya Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals. *Journal of Machine Learning Research*, 20(172):1–59, 2019b.

Yawen Cui, Xiao Wang, and Xiantao Xiao. A Two-phase Stochastic Momentum-Based Algorithm for Nonconvex Expectation-Constrained Optimization. *Journal of Scientific Computing*, 104(1):16, 2025.

Frank E. Curtis, Daniel P. Robinson, and Baoyu Zhou. Sequential quadratic optimization for stochastic optimization with deterministic nonlinear inequality and equality constraints. *SIAM Journal on Optimization*, 34(4), 2024.

Naum Dimitrieski, Jing Cao, and Christian Ebenbauer. Stochastic gradient descent for constrained optimization based on adaptive relaxed barrier functions. *IEEE Control Systems Letters*, 9:829–834, 2025.

Asen L. Dontchev and R. Tyrrell Rockafellar. *Implicit Functions and Solution Mappings: A View from Variational Analysis*. Springer Series in Operations Research and Financial Engineering. Springer, 2014.

Francisco Facchinei and Jong-Shi Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2003.

Yuchen Fang, Sen Na, Michael W. Mahoney, and Mladen Kolar. Fully stochastic trust-region sequential quadratic programming for equality-constrained optimization problems. *SIAM Journal on Optimization*, 34(2):2007–2037, 2024.

Diego Feijer and Fernando Paganini. Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica*, 46:1974–1981, 2010.

Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael Friedlander. Satisfying Real-world Goals with Dataset Constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, 2016.

Ruichuan Huang, Jiawei Zhang, and Ahmet Alacaoglu. Stochastic Smoothed Primal-Dual Algorithms for Nonconvex Optimization with Linear Inequality Constraints. *arXiv preprint arXiv:2504.07607*, 2025.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

Lingzi Jin and Xiao Wang. A stochastic primal-dual method for a class of nonconvex constrained optimization. *Computational Optimization and Applications*, 83(1):143–180, 2022.

Harold J. Kushner and G. George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*, volume 35 of *Stochastic Modelling and Applied Probability*. Springer, 2003.

Guanghui Lan and Zhiqiang Zhou. Algorithms for stochastic optimization with function or expectation constraints. *Computational Optimization and Applications*, 76(2):461–498, 2020.

Zichong Li, Pin-Yu Chen, Sijia Liu, Songtao Lu, and Yangyang Xu. Stochastic inexact augmented Lagrangian method for nonconvex expectation constrained optimization. *Computational Optimization and Applications*, 87(1):117–147, 2024.

Songtao Lu. A Single-Loop Gradient Descent and Perturbed Ascent Algorithm for Nonconvex Functional Constrained Optimization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 14315–14357, Baltimore, Maryland, USA, 2022.

Bruce A. Murtagh and Michael A. Saunders. A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Mathematical Programming Study*, 16:84–117, 1982.

Sen Na, Mihai Anitescu, and Mladen Kolar. Inequality constrained stochastic nonlinear optimization via active-set sequential quadratic programming. *Mathematical Programming*, 202(1–2):279–353, 2023.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Herbert Robbins and David Siegmund. A Convergence Theorem for Non Negative Almost Supermartingales and Some Applications. In *Optimizing Methods in Statistics*, pp. 233–257. Academic Press, 1971.

R. T. Rockafellar. Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.

R. Tyrrell Rockafellar and Roger J. B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften*. Springer, 1998.

Ashudeep Singh and Thorsten Joachims. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, London, UK, 2018.

Motahareh Sohrabi, Juan Ramirez, Tianyue H. Zhang, Simon Lacoste-Julien, and Jose Gallego-Posada. On PI Controllers for Updating Lagrange Multipliers in Constrained Optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, Vienna, Austria, 2024.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive Safety in Reinforcement Learning by PID Lagrangian Methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9133–9143, Vienna, Austria, 2020.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward Constrained Policy Optimization. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, USA, 2019.

Mengdi Wang and Dimitri P. Bertsekas. Stochastic First-Order Methods with Random Constraint Projection. *SIAM Journal on Optimization*, 26(1):681–717, 2016.

Yangyang Xu. Primal-Dual Stochastic Gradient Method for Convex Programs with Many Functional Constraints. *SIAM Journal on Optimization*, 30(2):1664–1692, 2020.

Ming Yang, Gang Li, Quanqi Hu, Qihang Lin, and Tianbao Yang. Single-loop Algorithms for Stochastic Non-Convex Optimization with Weakly-Convex Constraints. *Transactions on Machine Learning Research*, 2026.

Liwei Zhang, Yule Zhang, Jia Wu, and Xiantao Xiao. Solving Stochastic Optimization with Expectation Constraints Efficiently by a Stochastic Augmented Lagrangian-Type Algorithm. *INFORMS Journal on Computing*, 34(6):2989–3006, 2022.

A Hyperparameter Roles and Tuning Guidelines for RCML

Table 8 summarizes the practical hyperparameter tuning guidelines. Some are standard in stochastic primal–dual optimization, some belong to the projected-pressure residual mechanism, and some are only used when optional stabilization modules are activated.

Table 8: Roles and tuning guidelines of the main hyperparameters. "Origin" indicates whether a parameter belongs to the stochastic primal–dual interface, the projected-pressure residual mechanism, or an optional RCML stabilization module.

Parameter	Origin	Role	Tune?	Practical guideline
α_k	Primal–dual interface	Primal stepsize for updating x_k. It controls objective descent and the sensitivity of the primal step to multiplier pressure.	Yes	Tune as in standard stochastic optimization. Use smaller values when constraint gradients are noisy or when the projected pressure λ_k is large.
η_k	Primal–dual interface	Multiplier-memory stepsize in equation 3. It controls how strongly the feedback signal s_k changes u_k.	Yes	Tune jointly with κ_{I}. For residual-integral variants, the effective gain $\beta_k = \eta_k \kappa_{\mathrm{I}}$ should satisfy $0 \leq \beta_k \leq 1$.
u_0	Primal–dual interface	Initial multiplier memory.	Usually no	Set $u_0 = 0$ unless prior knowledge indicates that some constraints should be active at initialization.
$\boldsymbol{\rho}_k$	Projected-pressure mechanism	Pressure-scale vector used in equation 5. It controls how strongly each constraint signal is amplified before projection.	Yes	Use either a fixed isotropic scale $\boldsymbol{\rho}_k = \rho_0 \mathbf{1}$ or an adaptive coordinatewise scale. Larger values enforce constraints more aggressively but may amplify noisy mini-batch estimates.
ρ_0	Fixed pressure scale	Scalar scale used when the pressure scale is isotropic, $\boldsymbol{\rho}_k = \rho_0 \mathbf{1}$.	Yes	Tune together with α_k. Larger ρ_0 increases feasibility pressure but also increases sensitivity to constraint noise.
$\rho_{k,i}$	Adaptive pressure scaling	The i-th coordinate of the adaptive pressure-scale vector $\boldsymbol{\rho}_k$.	Usually no after rule is fixed	Computed by equation 8. Tune the base scale and clipping bounds rather than each coordinate separately.
κ_{I}	Residual memory tracking	Integral gain from the residual d_k to the memory signal s_k. It controls finite-gain tracking speed.	Yes	Core RCML parameter. Small values produce smoother memory; large values approach projected ALM replacement. Tune through $\beta_k = \eta_k \kappa_{\mathrm{I}}$.
γ_k	Constraint filtering	Smoothing gain in equation 7.	Yes	Use $\gamma_k = 1$ for unfiltered feedback. Decrease it under high mini-batch noise, but avoid excessive lag.
v_0	Adaptive scaling state	Initial second-moment state for constraint magnitudes.	No	Set $v_0 = 0$. Bias correction in $\widehat{v}_{k+1}$ reduces initialization effects.
η_v	Adaptive pressure scaling	Second-moment update rate for the adaptive scale.	Low priority	Use a small to moderate value. Larger values adapt faster but may make $\boldsymbol{\rho}_k$ fluctuate.
κ_ρ	Adaptive pressure scaling	Base scale in equation 8.	Yes for adaptive variants	Controls the average level of $\rho_{k,i}$. Tune together with $\rho_{\min}$ and $\rho_{\max}$.

Continued on next page

Table 8: Roles and tuning guidelines of the main hyperparameters. Continued.

Parameter	Origin	Role	Tune?	Practical guideline
$\rho_{\min}, \rho_{\max}$	Adaptive pressure scaling	Lower and upper clipping bounds for coordinatewise pressure scales.	Yes for adaptive variants	Use $\rho_{\min} > 0$ to avoid vanishing pressure scaling and $\rho_{\max}$ to limit noise amplification.
ϵ	Adaptive pressure scaling	Numerical stabilizer in $\sqrt{\widehat{v}_{k+1,i} + \epsilon}$.	Usually no	Use a small constant, such as 10^{-8} or 10^{-6}, depending on numerical precision.
ν	Residual-νPI correction	Residual smoothing factor in equation 9.	Yes for robust variant	Larger values produce smoother but slower dynamic correction. Smaller values respond faster to changing residual trends.
κ_{P}	Residual-νPI correction	Gain multiplying $\xi_k - \xi_{k-1}$ in equation 10.	Yes for robust variant	Start from $\kappa_{\mathrm{P}} = 0$ and increase gradually. Large values may accelerate response but can also amplify residual oscillations.
$\bar{c}_{-1}, \xi_{-1}$	Auxiliary states	Initial states for constraint filtering and residual smoothing.	No	Set to zero unless warm-start information is available.
B_k	Stochastic estimation	Mini-batch size for estimating $\hat{g}_k$, $\hat{c}_k$, and $\widehat{J}_{c,k}$.	Task-dependent	Larger batches reduce feedback noise but increase computation. RCML is designed for settings where full-batch constraints are expensive.

In practice, we tune RCML hierarchically. First, choose the primal stepsize α_k and a pressure scale. For the basic projected-pressure update, this is usually a fixed isotropic scale $\boldsymbol{\rho}_k = \rho_0 \mathbf{1}$. Second, tune the effective residual tracking gain $\beta_k = \eta_k \kappa_{\mathrm{I}}$, which determines how close the method is to full projected ALM replacement. Third, enable constraint filtering by decreasing γ_k when mini-batch constraint estimates are visibly noisy. Finally, activate adaptive pressure scaling when constraints have heterogeneous magnitudes, and activate residual-νPI correction when multiplier oscillations remain after filtering.

This hierarchy also clarifies which parameters are intrinsic to RCML. The primal stepsize α_k, multiplier-memory stepsize η_k, initial multiplier u_0, and batch size B_k are standard components of stochastic primal–dual learning. The projected-pressure scale $\boldsymbol{\rho}_k$, residual signal d_k, and tracking gain κ_{I} define the RCML backbone. The parameters γ_k, η_v, κ_ρ, $\rho_{\min}$, $\rho_{\max}$, ϵ, ν, and κ_{P} belong to optional stabilization modules. Therefore, the minimal RCML implementation only requires tuning α_k, ρ_0, and $\beta_k = \eta_k \kappa_{\mathrm{I}}$, while the additional parameters are introduced only when filtering, adaptive scaling, or dynamic correction is used.

B Additional Experimental Details

This appendix summarizes the implementation details needed to reproduce the experiments. All experiments were conducted on a workstation equipped with an NVIDIA RTX 4060 Ti GPU and an Intel Core i5-13490F CPU. The fair-ranking experiment uses PyTorch with CUDA when available, while the other experiments are implemented using NumPy. The public repository contains one main script for each experiment.

B.1 Experiment 1 Details

Experiment 1 uses synthetic stochastic LP, convex QP, and mildly nonconvex QP tasks with $d = 30$ decision variables and $m = 30$ inequality constraints. For each task, we generate 2048 objective samples and 2048 constraint samples. The gradient mini-batch size is 32, and each method is evaluated over 10 random seeds. The reliability tolerance is 5×10^{-2}.

The stationary regime uses 500 iterations, a tail window of 50, constraint batch size 32, and no additional observation noise. The high-noise regime uses 1500 iterations, a tail window of 150, constraint batch size

4, and additive constraint noise with standard deviation 0.25. The heterogeneous-scale regime uses 700 iterations, a tail window of 70, constraint batch size 16, additive noise with standard deviation 0.05, and log-uniform positive rescaling of constraint channels. The active-set switching regime uses 500 iterations, a switch at iteration 250, a tail window of 50, constraint batch size 16, and additive constraint noise with standard deviation 0.07. Before the switch, constraints are tightened to accumulate multiplier memory; after the switch, they are relaxed to test stale-memory release. The optional residual-νPI and τ-gated variants are used only in the active-set switching ablation and are not treated as default RCML methods.

B.2 Experiment 2 Details

Experiment 2 uses a tolerance-aware stochastic energy-reserve allocation problem with $d = 20$ decision variables and $m = 30$ constraints. The training and testing sample sizes are 30000 and 60000, respectively. Each method is evaluated over 5 random seeds for 1600 iterations. Tail metrics are computed over the last 300 iterations, and evaluation is performed every 10 iterations.

The normal-noise regime uses batch size $B = 256$, lognormal noise scale 0.25, and tolerance $\delta_{\mathrm{tol}} = 10^{-2}$. The moderate-noise stress regime uses batch size $B = 96$, lognormal noise scale 0.30, spike probability 0.05, spike scale 2.2, spike fraction 0.20, and tolerance $\delta_{\mathrm{tol}} = 2 \times 10^{-2}$. The tolerance-aware score uses $\omega_{\mathrm{tol}} = 100$.

The common optimization parameters are primal learning rate 5×10^{-2}, multiplier-memory learning rate $\eta_u = 4 \times 10^{-2}$, fixed isotropic pressure scale $\rho_0 = 1.0$, residual smoothing factor $\nu = 0.65$, integral gain $\kappa_{\mathrm{I}} = 1.0$, and residual-$\nu$PI gain $\kappa_{\mathrm{P}} = 0.05$. For the selected RCML-Adaptive configuration, we use $\gamma = 0.70$, $\eta_v = 0.05$, $\kappa_\rho = 0.80$, $\rho_{\min} = 0.15$, and $\rho_{\max} = 4.0$. The configuration sweep in Table 9 varies γ, κ_ρ, $\rho_{\min}$, and the multiplier-memory learning rate η_u.

B.3 Experiment 3 Details

Experiment 3 uses a nonconvex stochastic pricing-inventory allocation problem with $d = 30$ items. For item i, the decision variables are inventory x_i and price r_i. Demand follows an exponential price-response model with stochastic shocks,

$$D_i(r_i, \xi_i) = \bar{D}_i \exp[-a_i(r_i - r_i^0)]\xi_i + \mathrm{spike}_i.$$

The objective is to maximize expected profit, equivalently to minimize negative profit. The constraint requires expected shortage to remain below an item-dependent tolerance.

The training and testing sample sizes are 20000 and 25000, respectively. Each method is evaluated with 5 random seeds and 5 random initializations, resulting in 25 seed–initialization runs per scenario. Each run uses 650 iterations, with tail metrics computed over the last 120 iterations. A run is considered reliable if its tail tolerance-feasible rate is at least 0.9.

The normal-noise regime uses batch size $B = 256$ and tolerance $\delta_{\mathrm{tol}} = 1.5 \times 10^{-2}$. The moderate-noise stress regime uses batch size $B = 128$ and tolerance $\delta_{\mathrm{tol}} = 3.0 \times 10^{-2}$. The primal learning rates are 4.0×10^{-2} for inventory and 2.5×10^{-2} for price. The multiplier-memory learning rate is $\eta_u = 4.0 \times 10^{-2}$, and the fixed isotropic pressure scale is $\rho_0 = 10.0$. For RCML-Adaptive, we use $\gamma = 0.70$, $\eta_v = 0.05$, $\kappa_\rho = 0.80$, $\rho_{\min} = 0.15$, and $\rho_{\max} = 10.0$.

B.4 Experiment 4 Details

Experiment 4 uses a large-scale learning-to-rank dataset. The training set contains 10000 queries and the test set contains 3000 queries. Each query contains 30 candidate items with 24-dimensional features and a binary group attribute. The neural ranker is a two-hidden-layer multilayer perceptron with hidden dimension 64 and ReLU activations. The model is trained with AdamW for 1000 iterations using batch size 128, learning rate 10^{-3}, weight decay 10^{-5}, and gradient clipping threshold 5.0. Each method is evaluated over 5 random seeds.

The ranking loss is a ListNet-type loss. Soft exposure is computed using softmax temperature 0.75, while the target distribution uses temperature 1.0. The evaluation fairness threshold is $\epsilon_{\mathrm{eval}} = 0.012$, and the tolerance band is $\delta_{\mathrm{tol}} = 0.003$, giving the admissible exposure-gap threshold 0.015. For residual-controlled methods, the internal training threshold is $\epsilon_{\mathrm{train}} = 0.010$. The reported utility metric is NDCG@10.

The multiplier-control parameters are multiplier-memory learning rate $\eta_u = 0.16$, fixed isotropic pressure scale $\rho_0 = 18.0$, projected-ALM fixed isotropic pressure scale $\rho_{0,\mathrm{ALM}} = 10.0$, filtering parameter $\gamma = 0.85$, adaptive scaling rate $\eta_v = 0.05$, adaptive base scale $\kappa_\rho = 4.0$, clipping bounds $\rho_{\min} = 2.0$, $\rho_{\max} = 60.0$, residual smoothing factor $\nu = 0.70$, integral gain $\kappa_{\mathrm{I}} = 1.50$, and residual-$\nu$PI gain $\kappa_{\mathrm{P}} = 0.30$. The fixed fairness penalty baseline uses penalty weight 300.0, and the stochastic barrier baseline uses barrier parameter 0.020, barrier shift 0.003, and numerical barrier constant 10^{-5}. The synthetic ranking data are generated with group-dependent relevance bias, group-dependent feature shift, query-level group skew, and additive noise, so that the unconstrained ERM solution violates the admissible exposure band and activates multiplier-based constrained training.

B.5 Additional Ablation Tables

Table 9: Configuration selection for RCML-Adaptive. The screening score is averaged over the two stochastic feedback regimes; lower is better.

Variant	γ	κ_ρ	$\rho_{\min}$	η_u	Score	TolFeas	TA	Cost
RCML-Adaptive-lowlag	0.70	0.80	0.15	0.04	**6.098**	**0.743**	**5.673**	5.569
RCML-Adaptive-strong	0.50	1.00	0.15	0.04	6.298	0.707	5.707	5.567
RCML-Adaptive-mild	0.50	0.80	0.10	0.04	6.298	0.707	5.707	5.567
RCML-Adaptive-fastdual	0.60	0.80	0.20	0.05	6.511	0.683	5.719	5.568
RCML-Adaptive-default	0.35	0.65	0.10	0.04	6.561	0.677	5.736	5.566

Table 10: Optional dynamic extensions under active-set switching. Constraints are tightened before the switch and relaxed after the switch. PostDualTV is the post-switch average of $\|u_{k+1} - u_k\|_2$, and ExcessMem is the post-switch average of $\|u_k\|_2/(1 + \|u_{k_s}\|_2)$.

Problem	Method	PostObj	PostViol	PostDualTV	ExcessMem	Runtime
LP	RCML-Adaptive	-1.061×10^{1}	2.430×10^{-2}	4.232×10^{-3}	2.785×10^{-1}	1.09×10^{-1}
LP	RCML-Adaptive-τ	-1.049×10^{1}	2.437×10^{-2}	4.104×10^{-3}	2.661×10^{-1}	1.04×10^{-1}
LP	Residual-νPI	-1.062×10^{1}	2.423×10^{-2}	5.099×10^{-3}	2.786×10^{-1}	1.10×10^{-1}
QP	RCML-Adaptive	5.662×10^{-1}	3.745×10^{-3}	1.793×10^{-3}	9.574×10^{-2}	1.18×10^{-1}
QP	RCML-Adaptive-τ	6.238×10^{-1}	3.236×10^{-3}	1.684×10^{-3}	9.237×10^{-2}	1.17×10^{-1}
QP	Residual-νPI	5.664×10^{-1}	3.731×10^{-3}	2.090×10^{-3}	9.570×10^{-2}	1.18×10^{-1}
NCVQP	RCML-Adaptive	-1.110×10^{-1}	0.000	5.149×10^{-4}	6.328×10^{-2}	8.53×10^{-2}
NCVQP	RCML-Adaptive-τ	-1.454×10^{-1}	0.000	5.244×10^{-4}	6.428×10^{-2}	8.39×10^{-2}
NCVQP	Residual-νPI	-1.283×10^{-1}	0.000	5.202×10^{-4}	6.362×10^{-2}	8.73×10^{-2}