IMPLICIT GRADIENT TRANSPORT FOR DELAYED DECISION-FOCUSED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision-focused learning (DFL) trains prediction models for better decisions, rather than prediction accuracy alone. In the online setting, predictions can guide successive decisions; a learning algorithm (the learner) updates a model as decision outcomes arrive. When those outcomes arrive late, both the model and the decisions it would produce have changed, making old learning gradients stale. How can the learner make better use of this feedback? We introduce implicit gradient transport for online mirror descent (IGT-OMD) to answer this question and combine gradients from new feedback with changes in past gradients caused by model updates. One version of this algorithm reuses stored decisions and gradient information (frozen transport); another recomputes them at the current model (full re-evaluation). For stationary problems, we bound the frozen gradient-descent variant's excess cumulative loss over the best fixed model chosen in hindsight (i.e., regret), and derive a sufficient number of iterations to solve the decision problem. For a shared strongly convex quadratic model, continuous and discrete stability analyses give an explicit learning-rate rule. Experiments cover linear–quadratic regulator (LQR) stability, Warcraft paths, and Sinkhorn transport. At delay 50 on Sinkhorn, full re-evaluation with Adam reduces loss by 10.81% versus learning from newly arrived feedback alone and by 26.6% versus the best tested delayed optimistic mirror descent (DOOMD) baseline. The transport gain costs $40.28\times$ the arrival-only control's training-update time. Independent random realizations confirm the transport benefit; frozen transport offers a lower-cost option in the tradeoff between decision quality and computation.

1 INTRODUCTION

A route planner predicts travel times on road segments and chooses the route with the smallest predicted total time. A small prediction error on an uncompetitive route may be harmless; between nearly tied routes, it can switch the choice to a slower route. Training the predictor for the quality of the selected route, rather than prediction accuracy alone, is our learning goal (*decision-focused learning*; DFL) (Wilder et al., 2019; Elmachtoub & Grigas, 2022). Learning the predictions and optimizing the resulting route form a two-level construction (*bilevel optimization*): the outer learning algorithm updates the predictor; the inner solver chooses a route using those predictions.

In *online learning*, decisions are made successively and the prediction model is updated using only feedback available so far. Suppose travel times arrive only after several more routes have been selected. Other feedback may meanwhile update the predictor and the route it would choose. A learning gradient that accounts for how predictions change the downstream decision (*hypergradient*) can therefore become stale through both dependencies. This potential worsening through the inner decision (*staleness amplification*) motivates our method.

How much decision computation should the learner repeat when feedback arrives? The learner can evaluate the gradient at the model that chose the route, reuse the stored decision and its gradient information while updating the predictor dependence, or recompute the decision at the current model. We study the last two choices with one memory rule: add the gradients of newly revealed decision losses and the *changes* in gradients of previously revealed examples. This difference correction (*gradient transport*) updates old information rather than adding another copy of an old gradient.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Inspired by implicit gradient transport (Arnold et al., 2019), we develop this rule for gradients that pass through a decision problem. We call it *implicit gradient transport for online mirror descent* (IGT-OMD). The analyzed mirror-descent update is ordinary gradient descent with the constructed gradient. Reducing the rate according to the largest observed number of outstanding outcomes (*queue damping*) controls the step size of the algorithm.

Contributions. (1) Reuse delayed decisions at two costs. One memory rule supports cheap frozen transport and full decision re-evaluation. **(2) Quantify delay, inner accuracy, and stable steps.** For the stationary frozen update, we prove a regret bound, derive a logarithmic sufficient inner-solver budget, and link continuous stability to an exact discrete learning-rate interval under common curvature. **(3) Measure where reuse helps.** Full re-evaluation improves Sinkhorn loss by 6.31–10.81% over matched arrival-only learning on new random realizations and by 18.2–26.6% over the best tested delayed optimistic mirror descent (DOOMD) baseline after a rate-grid extension. Matched comparisons separate optimizer, gradient freshness, and transport effects, with their computational price.

1.1 Related Work

Decision-focused learning and differentiable decisions. Mandi et al. (2024) survey decision-focused methods and benchmarks. Optimization layers differentiate continuous solution maps (Amos & Kolter, 2017; Agrawal et al., 2019), whereas blackbox differentiation and randomized perturbations supply learning signals for discrete solvers (Vlastelica et al., 2020; Berthet et al., 2020). These supply gradient oracles; we study their reuse after delayed feedback. **Bilevel optimization and inner accuracy.** Approximate hypergradients avoid exact inner solves (Pedregosa, 2016; Ghadimi & Wang, 2018), using optimization dynamics or optimality equations (Franceschi et al., 2018; Grazzi et al., 2020; Ji et al., 2021). Online bilevel methods study changing inner and outer objectives through alternating updates (Ataee Tarzanagh et al., 2024), windowed hypergradient estimates (Lin et al., 2023), and stochastic first- or zeroth-order oracles (Nazari et al., 2025). Online decision-focused learning also has decision-specific analyses (Capitaine et al., 2026). These address how to compute or track a bilevel gradient; our stationary analysis isolates what feedback delay and cached inner states add to its error. **Delayed online learning and adaptive steps.** Mirror descent and follow-the-regularized-leader (FTRL) differ (Beck & Teboulle, 2003; McMahan, 2017). Delay reductions (Joulani et al., 2013), online gradient descent (OGD) (Quanrud & Khashabi, 2015), D-FTRL (Qiu et al., 2025), and delayed optimism (Flaspohler et al., 2021) address delay without bilevel structure; Banker-OMD treats bandit feedback (Huang et al., 2023). Their delay-aware updates complement our decision-aware gradient reuse. **Gradient transport and continuous-time analysis.** Arnold et al. (2019) transport gradients to control bias and variance; we derive a bilevel construction inspired by that principle. Yu et al. (2025) study delayed stochastic gradient descent (SGD) via delay differential equations. Our continuous/discrete results share one restricted recurrence.

2 Problem Formulation

2.1 Online Bilevel Optimization with Delayed Feedback

Each round $t = 1, \dots, T$ is a decision followed by available feedback and an update; T is the finite horizon. The learner observes instance features (the *context*). A predictor with p parameters $\theta_t \in \mathbb{R}^p$ maps these features to inputs of the decision problem $f_t(w; \theta_t)$, where $w \in \mathbb{R}^q$ is a q-dimensional decision. In the routing example, predicted travel times define f_t, while the outer loss ℓ_t measures the route's true cost. The exact optimal decision and its true loss are

$$w_t^\star(\theta) = \arg\min_w f_t(w; \theta), \qquad F_t(\theta) = \ell_t(w_t^\star(\theta), \theta). \tag{1}$$

Appendix A collects notation (Table A.1); Appendix B explains bilevel optimization, mirror descent and delay equations. The learner executes w_t using θ_t, then processes available feedback to obtain θ_{t+1}. The solver may return $w_t \neq w_t^\star(\theta_t)$. Features and predicted inputs are known when acting; the true outcome arrives at the end of round $a_t = t + d_t$, with integer delay $d_t \geq 0$. Only then can the evaluator score the original decision as $\ell_t(w_t, \theta_t)$, not a decision recomputed after learning (*pre-update scoring*).

Let s denote a decision's *source round*: when it was made. Its *record* stores the context and executed decision. After registering the current decision, let A_t index arriving feedback, Q_t index pending feedback, and $\bar{q}_t$ be the largest pending count seen so far:

$$A_t = \{s \le t : s + d_s = t\}, \quad Q_t = \{s \le t : s + d_s > t\}, \quad \bar{q}_t = \max_{u \le t} |Q_u|. \tag{2}$$

On arrival, feedback and gradients are added; the record leaves the queue and becomes eligible for the *transport buffer* for later updates. The record's age $t - s \ge d_s$ keeps growing after arrival: delay does not bound the record's age, while buffer capacity bounds the number of retained records.

We compare learning with one fixed model chosen after seeing all T losses. Over a fixed comparison set $\Omega \subseteq \mathbb{R}^p$, assume some $\theta^\star$ attains $\min_{\theta\in\Omega} \sum_{t=1}^T F_t(\theta)$. We define *executed-decision regret* as cumulative loss in excess of this fixed comparator, separating learning error from execution error:

$$R_T^{\mathrm{dec}} = \sum_{t=1}^T \ell_t(w_t, \theta_t) - \min_{\theta\in\Omega} \sum_{t=1}^T F_t(\theta) = R_T^{\mathrm{env}} + E_T, \quad E_T = \sum_{t=1}^T [\ell_t(w_t, \theta_t) - F_t(\theta_t)]. \tag{3}$$

Here R_T^{env} uses exact inner solutions, and E_T accounts for executing approximate ones. Our empirical metrics—stability boundaries, decision losses, and gaps to the current optimal decision—answer different questions from this fixed-comparator regret.

2.2 Hypergradient Computation via Implicit Differentiation

While outcome s is pending, other feedback may change the model from its source parameters θ_s, even with stationary outcomes; changing outcomes add environmental drift. An exact hypergradient at current parameters θ requires $w_s^\star(\theta)$, a new optimum for the same source problem, rather than w_s, the decision actually executed with θ_s. For a smooth, unconstrained inner problem with nonsingular inner Hessian, define $H_s(w, \theta) = \nabla^2_{ww} f_s(w; \theta)$ and $J_s(w, \theta) = \nabla^2_{w\theta} f_s(w; \theta) \in \mathbb{R}^{q\times p}$. Here H_s describes inner curvature and J_s how the inner gradient changes with predictor parameters. Implicit differentiation gives

$$\nabla F_s(\theta) = \nabla_\theta \ell_s(w_s^\star(\theta), \theta) - J_s(w_s^\star(\theta), \theta)^\top H_s(w_s^\star(\theta), \theta)^{-1} \nabla_w \ell_s(w_s^\star(\theta), \theta). \tag{4}$$

The first term is the predictor's direct effect; the second carries its effect through the optimal decision. A linear solve for $H_s^{-1}\nabla_w \ell_s$ gives an auxiliary vector (the *adjoint*) without forming the inverse Hessian. Our algorithm caches quantities to avoid repeating both solves for every retained example.

3 IGT-OMD Algorithm

How should feedback be evaluated on arrival and reused afterward? Our main construction, *frozen implicit IGT-OMD*, caches a decision and adjoint to refresh gradients without repeated inner solves; its gradient-descent update is analyzed below. For other decision layers, *frozen task adapters* cache a task-specific decision-gradient direction instead. *Full re-evaluation* tests whether refreshing the decision calculation itself is worth the repeated inner computation. Table 1 maps these choices; all use the same gradient-difference memory rule.

Variant	What is reused or recomputed?	Role in this paper
Frozen implicit	Keep the source decision and arrival adjoint; update the parameter-dependent terms.	Euclidean theory; LQR implementation.
Frozen task adapter	Keep a task's decision-gradient direction; update the predictor Jacobian.	Warcraft and Sinkhorn factorials.
Full re-evaluation	Repeat the finite inner computation and differentiate it at the current model.	Sinkhorn with Adam; fresh confirmation.

Table 1: One memory rule, different gradient calculations (*oracles*). Frozen quantities stay fixed during reuse.

What is cached in the frozen variant? To reuse (4) without repeatedly solving the inner problem, we retain the executed decision w_s. When feedback from source round s arrives at a_s, we compute its adjoint v_s and define the reusable gradient $g_s(\theta)$:

$$v_s = H_s(w_s, \theta_{a_s})^{-1} \nabla_w \ell_s(w_s, \theta_{a_s}), \qquad g_s(\theta) = \nabla_\theta \ell_s(w_s, \theta) - J_s(w_s, \theta)^\top v_s. \tag{5}$$

The cached decision–adjoint pair (w_s, v_s) stays fixed; only the parameter-dependent terms in $g_s(\theta)$ change. At arrival this is the implicit formula at the executed decision; later it is a cheaper approximation to the current hypergradient (a *surrogate*).

Why transport changes? Full-gradient replay makes a record's accumulated contribution depend on its number of reuses. Our algorithm contributes its gradient on arrival, then only changes. For a continuously retained record, these unweighted contributions telescope to its latest surrogate gradient: $2.0 + 0.3 + 0.1 = 2.4$, not repeated copies of the full gradient. This is not generally a parameter displacement under varying rates or Adam. Let $\mathcal{B}_t$ be the indices of at most $B_{\max}$ previously arrived records retained before round t's new admissions, unlike the pending set Q_t. Their increments $u_{s,t}$ form the correction c_t and complete update direction h_t:

$$u_{s,t} = g_s(\theta_t) - g_s(\theta_{t-1}), \qquad c_t = \sum_{s \in \mathcal{B}_t} u_{s,t}, \qquad h_t = \sum_{s \in A_t} g_s(\theta_t) + c_t. \tag{6}$$

Per-round procedure. The pending queue sets the rate; the arrived buffer supplies corrections. We use $\eta_t = \eta_0 / \sqrt{1 + \beta \bar{q}_t}$: base rate $\eta_0 > 0$ and damping strength $\beta \geq 0$ reduce steps as the queue envelope grows. Algorithm 1 applies the unprojected gradient update $\theta_{t+1} = \theta_t - \eta_t h_t$, which is mirror descent with $\psi(\theta) = \|\theta\|^2/2$ (*Euclidean OMD*) (Beck & Teboulle, 2003). Figure 1 gives an example with $d = B_{\max} = 2$.

Algorithm 1 Frozen Euclidean IGT-OMD

Require: Rate $\eta_0 > 0$, damping $\beta \geq 0$, capacity $B_{\max}$.

1: Initialize $\theta_0 = \theta_1$; empty queue and buffer; $\bar{q}_0 = 0$.
2: **for** $t = 1, \ldots, T$ **do**
3: Execute w_t using θ_t; register source state and arrival $t + d_t$.
4: Collect A_t; update the pending set Q_t and envelope $\bar{q}_t$.
5: Snapshot old buffer $\mathcal{B}_t$; set $h_t = 0$.
6: For each $s \in A_t$ in source order, form (5); add $g_s(\theta_t)$ to h_t; stage its cache.
7: For each $s \in \mathcal{B}_t$ in source order, add $u_{s,t}$ to h_t; cache $g_s(\theta_t)$ in place of $g_s(\theta_{t-1})$.
8: Admit arrivals; keep $B_{\max}$ largest source indices.
9: Set $\eta_t = \eta_0 / \sqrt{1 + \beta \bar{q}_t}$; update $\theta_{t+1} = \theta_t - \eta_t h_t$.
10: **end for**

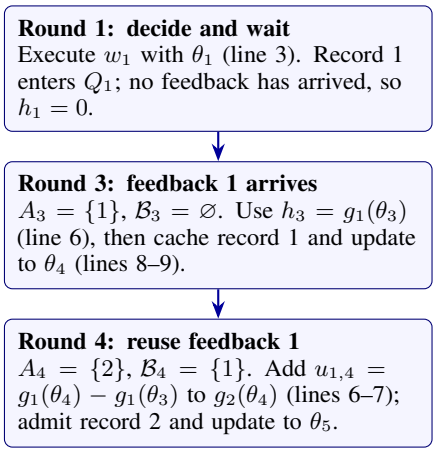


Figure 1: Example feedback flow for $d = B_{\max} = 2$, linked to algorithm lines. Decisions occur every round; rounds 1–2 have no updates. Record 1 leaves the pending queue on arrival in round 3 and first contributes a transport correction from $\mathcal{B}_4$ in round 4.

Each retained record is evaluated every round, caching its gradient. New arrivals enter after transport and receive no same-round correction; eviction ends further corrections. Arrivals and corrections are summed without clipping, averaging, or a moving-average gradient estimator. The Adam+IGT variation applies Adam to the same h_t and schedule (Kingma & Ba, 2015). Setting $B_{\max} = 0$ disables transport without changing arrival computation; $\beta = 0$ disables damping.

Cost and memory. Arrival builds a cache and solves for an adjoint (cost C_a); later reuse evaluates only the frozen gradient (cost C_g). These are different operations, so gradient work is $O(|A_t|C_a + |\mathcal{B}_t|C_g + p)$, excluding the executed inner solve and queue bookkeeping. Persistent storage is $O(p + B_{\max}(p + q) + |Q_t|q)$, plus task-specific feedback and diagnostic logs; arrival batches also need transient storage. With supplied dense derivatives, the mixed product costs $O(pq)$ and a direct $q \times q$ adjoint solve costs $O(q^3)$; constructing derivatives is additional. Structured products or solvers can be cheaper, so costs depend on the decision layer.

Full re-evaluation variant. To test the benefit of refreshing the decision dependence, retain source feedback and repeat the inner solver at the current model. If $w_s^K(\theta)$ is its output after K steps, use $g_s^{\mathrm{full,K}}(\theta) = \nabla_\theta \ell_s(w_s^K(\theta), \theta)$. This differentiates the entire finite computation, not an exact optimum; source data and the previous gradient stay cached. The arrival clock (A_t), correction (c_t), admission rule and chosen optimizer are unchanged. Each arrival and buffered evaluation repeats a solve; with forward-and-backward cost C_K, gradient work is $O((|A_t| + |\mathcal{B}_t|)C_K + p)$. *Full re-evaluation Adam+IGT* uses Adam. Its transport-off control also recomputes each arrival's inner solution and gradient, but keeps no buffer: the comparison isolates buffered corrections. Its benefits are tested empirically; the regret analysis concerns the frozen implicit Euclidean update.

4 Theoretical Analysis

We connect regret, hypergradient error, and inner accuracy for stationary smooth losses, $f_t = f$, $\ell_t = \ell$. Stability uses a common-curvature model. Regret requires constant delay; the algorithm accepts variable arrivals. Norms are Euclidean or induced operator norms.

Assumptions. We make the following assumptions about the inner problem, gradients, trajectory, and memory (Appendix C). **A1: A well-defined inner problem.** The inner loss is strongly convex with curvature $\mu_w > 0$, its mixed derivative is bounded by C_J, and $\|w_t - w^\star(\theta_t)\| \le \epsilon$, the inner-solver error tolerance. Let $\kappa_w = C_J/\mu_w$ bound the optimum's sensitivity to model changes. **A2: Regular hypergradients.** The implicit formula (4) is L_g-Lipschitz in the inner state; every frozen surrogate is L_S-Lipschitz in θ. Put $\kappa = L_g\kappa_w$ to combine inner-solution and hypergradient sensitivities. **A3: Controlled outer trajectories.** F is convex on the fixed comparison set Ω, which contains the trajectory and a minimizer $\theta^\star$. On that trajectory, $F(\theta) - F(\theta^\star) \ge (\mu_F/2)\|\theta - \theta^\star\|^2$, $\|\theta - \theta^\star\| \le D$, and $\|h_t\| \le G$. The gradient bound $\|\nabla F\| \le G_F$ holds along comparison segments. These are trajectory premises: D, G and G_F bound distance, update direction, and exact gradient, respectively. **A4: Memory geometry.** For the actual increments, $\|\sum_{s\in\mathcal{B}_t} u_{s,t}\|^2 \le \rho \sum_{s\in\mathcal{B}_t} \|u_{s,t}\|^2$. The coefficient bounds reinforcement: orthogonal increments allow $\rho = 1$, whereas equal nonzero increments require $\rho \ge |\mathcal{B}_t|$. One may always take $\rho = B_{\max}$ when $B_{\max} \ge 1$; a uniform bound independent of buffer size requires additional control of alignment. Here μ_F measures *quadratic growth* of loss away from the comparator. A fixed affine subspace containing all iterates and the comparator may suffice; curvature on changing Jacobian ranges does not. Bounds independent of horizon and delay are called *uniform constants*.

Can delayed learning achieve low regret? The next result bounds learning error relative to the fixed comparator and separates the cost of executing approximate decisions.

Theorem 1 (Delayed bilevel regret)**.** *Suppose the four assumptions hold, $0 \le d < T$, $B_{\max} \le d$, and the queue and buffer are initially empty. Run frozen Euclidean IGT-OMD (Algorithm 1) with constant delay d, fixed $\beta > 0$, and $\eta_0 = c/\sqrt{T}$ for $c > 0$. For $D, G, G_F, \mu_F^{-1}, L_g, L_S, \kappa_w$ uniformly bounded in T and d, regret with exact inner solutions and regret for executed decisions satisfy*

$$R_T^{\mathrm{env}} = O\Big(\sqrt{T(1+d)} + T\epsilon^2\Big), \qquad R_T^{\mathrm{dec}} = R_T^{\mathrm{env}} + E_T. \tag{7}$$

The bound holds even with the worst-case choice $\rho = B_{\max}$. If additionally $E_T \le C_{\mathrm{exec}} T\epsilon^2$ for a constant C_{exec} independent of T, d, the same order bounds decision regret. Thus $d = o(T)$ and $\epsilon^2 = o(1)$ suffice for a sublinear upper bound under this additional execution-gap condition.

Takeaway. Under the execution condition, vanishing inner error and sublinear delay give sublinear decision regret. Inner strong convexity alone does not ensure this condition: outer loss can vary linearly with inner error. A uniformly Lipschitz inner-state outer gradient that vanishes at the exact decision gives the quadratic gap. For fixed $\beta > 0$, damping replaces the drift factor d^2 by $d^2/(1+\beta d) = O(d)$ (Appendix C.4). With exact inner solves, the delay dependence matches the $O(\sqrt{T(1+d)})$ constant-delay order of delayed online gradient descent (Quanrud & Khashabi, 2015), under the stated bilevel assumptions.

How does delay affect the hypergradient? The next result separates inner approximation, model movement during delay, and memory corrections.

Theorem 2 (Hypergradient sensitivity to delay)**.** *Under assumptions 1, 2 and 4, constant delay d, $B_{\max} \le d$, and the cache clock of Algorithm 1, write the predictor's parameter change as $\Delta_k = \theta_{k+1} - \theta_k$. For $t > d$, the update-direction error relative to the exact hypergradient obeys*

$$\|h_t - \nabla F(\theta_t)\| \le L_g \epsilon + \kappa \sum_{k=t-d}^{t-1} \|\Delta_k\| + L_S \sqrt{\rho |\mathcal{B}_t|}\, \|\Delta_{t-1}\|. \tag{8}$$

Takeaway. Delay enters through model movement while feedback waits, amplified by inner-optimum sensitivity κ. This upper bound does not ensure that memory improves decisions; matched experiments test that question. Nonconvex outer losses are allowed under A1, A2 and A4; regret additionally needs A3's convexity and trajectory conditions.

How accurately must we solve decisions? To convert envelope regret to executed-decision regret, we bound the outer loss's sensitivity to inner error. Its inner-state gradient bound M controls the linear term; gradient Lipschitz constant L_ℓ controls the quadratic remainder.

Theorem 3 (Inner accuracy and decision regret)**.** *Under Theorem 1's conditions, suppose $\ell(\cdot, \theta)$ has L_ℓ-Lipschitz gradient on the inner comparison segments and $\|\nabla_w \ell(w^\star(\theta_t), \theta_t)\| \le M$, with M, L_ℓ uniform in T, d. The execution-gap magnitude and decision regret obey*

$$|E_T| \le T(M\epsilon + L_\ell \epsilon^2/2), \qquad R_T^{\mathrm{dec}} = O\Big(\sqrt{T(1+d)} + T\epsilon + T\epsilon^2\Big). \tag{9}$$

Suppose each inner solve starts within distance R_w of its exact optimum, and each iteration reduces this distance by a factor $\gamma \in (0,1)$. Assume R_w, γ are uniform in T, d. After K iterations the distance is at most $R_w \gamma^K$; choose K so that $R_w \gamma^K \le T^{-1/2}$. This requires at most $O(1 + \log T)$ iterations and yields $R_T^{\mathrm{dec}} = O(\sqrt{T(1+d)})$.

Takeaway. Accuracy $\epsilon \le T^{-1/2}$ suffices; uniform initialization and contraction yield a logarithmic solver budget. Appendix C.9 bounds warm-start error; Appendix C.5 gives a $T\epsilon^2/2$ floor for prescribed inexact decisions.

When are delayed updates stable? Accuracy controls approximation error; stability also depends on how repeated gradient corrections affect parameter updates. We isolate this memory effect under common curvature, then compare a continuous model with the discrete update. Suppose all frozen surrogates have the common form $g_s(\theta) = C(\theta - \theta^\star)$, where $C \in \mathbb{R}^{p \times p}$ is symmetric positive definite, there is one arrival per clock, and the old buffer has constant size B. The applied Euclidean rate η is constant. Let x_n be a coordinate of the parameter error $\theta - \theta^\star$ along an eigenvector of C, with curvature $\lambda > 0$; n counts active updates. Algorithm 1 gives $x_{n+1} = [1 - \eta\lambda(1+B)]x_n + \eta\lambda B x_{n-1}$ (10). This isolates memory and step size from source-state error. To examine the role of lag, first consider a continuous model depending on its past state (a *delay differential equation*, DDE) (Yu et al., 2025).

Proposition 1 (Continuous-time stability)**.** *For $\lambda > 0$, $B \ge 0$, and any fixed lag $h \ge 0$, $x'(u) = -\lambda(1+B)x(u) + \lambda B x(u-h)$ (11) is exponentially stable. In slow time $u = n\eta$, forward Euler with step η applied to the member $h = \eta$ of this DDE family gives* (10) *exactly.*

Takeaway. Stability holds for every finite lag, with a lag-dependent decay rate, because the current-state coefficient exceeds the delayed one by λ. Here $h = \eta$ is one-clock memory; d is feedback delay.

Continuous stability alone does not determine which finite learning rates are stable; the discrete recurrence imposes the following restriction.

Proposition 2 (Discrete stability and learning-rate scaling)**.** *Under the common-curvature conditions above, all solutions of the discrete recurrence converge to zero if and only if*

$$0 < \eta < \frac{2}{\lambda_{\max}(C)(1+2B)}. \tag{12}$$

If the steady queue size is $\bar{q}$ and $\eta = \eta_0/\sqrt{1+\beta\bar{q}}$, the equivalent nominal-rate condition is $0 < \eta_0 < 2\sqrt{1+\beta\bar{q}}/[\lambda_{\max}(C)(1+2B)]$.

Takeaway. Applied rates in (12) give convergence in this model. For $\bar{q} = B = d$, damping changes the nominal boundary's large-d scaling from d^{-1} to $d^{-1/2}$. The applied boundary remains d^{-1}. Thus continuous stability complements a finite-step restriction. Transport off has the wider applied interval $0 < \eta < 2/\lambda_{\max}(C)$; memory's practical benefit goes beyond enlarging this quadratic-model interval.

5 Experiments

LQR tests *stable learning rates*, Warcraft *reuse for discrete decisions*, and Sinkhorn *gradient reuse and its cost*. The neural tasks test the mechanism beyond the stationary theorem's assumptions. Decisions are scored before same-clock learning, with ten paired seeds per comparison. Configurations and full results are in Appendices D–F; separate post-update replay studies appear in Appendix G.

5.1 Linear Quadratic Regulator Stability Boundary

We test frozen implicit IGT-OMD in an LQR proxy whose predictor estimates matrices $(\widehat{A}, \widehat{B})$ for a ten-state, three-control linear system. Ten warm-started gradient steps select a one-step quadratic-control gain. For delays 1–500, the base-scale search expands to bracket acceptance and rejection within 5%. Acceptance requires bounded loss, tail slope, and tail variability in at least six of ten seeds over $\max(1000, 4d)$ rounds (Appendix D.2). This tests finite-horizon rate stability, not regret, control cost or infinite-horizon Riccati stability.

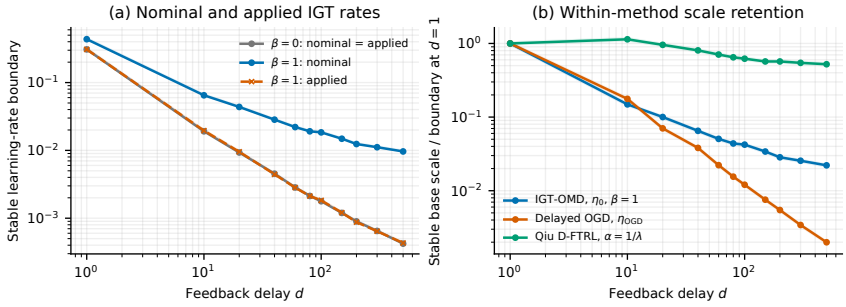


Figure 2: LQR learning-rate stability. (a) Damping changes nominal rates; applied-rate curves nearly coincide. (b) Stable base scale divided by each method's boundary at $d = 1$: normalization removes constant rate rescalings, not schedule differences. Shading shows search brackets, not confidence intervals. D-FTRL uses the published cumulative-gradient update. Higher retention means less delay-induced loss of stable base scale, not lower control cost.

For IGT-OMD, a log–log fit over $d \geq 20$ gives slopes -0.962 for the undamped nominal boundary, -0.470 for the damped nominal boundary, and -0.963 for the damped *applied* boundary. At $d = 60$, the nominal boundaries are 0.00284 and 0.02207: a $7.77\times$ ratio between $\beta = 0$ and $\beta = 1$, a within-IGT calibration effect. With constant delay and an empty initial queue, scaling η_0 by $\sqrt{1 + \beta d}$ leaves the applied IGT trajectory unchanged; the observed differences fit the search resolution.

We compare delayed online gradient descent (OGD) (Quanrud & Khashabi, 2015) with fixed rate η_{OGD} and the unconstrained delayed follow-the-regularized-leader (D-FTRL) update of Qiu et al. (2025), which uses $\alpha/t = 1/(\lambda t)$ on its cumulative observed gradient, around the mean of all past decisions. Figure 2(b) compares their within-method scale retention. At $d = 500$, it is approximately 2.21% for damped IGT, 0.20% for delayed OGD, and 52.21% for D-FTRL. Thus IGT's nominal scale degrades less than OGD's in this experiment, while exact D-FTRL retains the largest fraction. These distinct schedules do not define a common applied step. The methods use their stated rate rules, not a cross-method matched-damping comparison (Appendix D.2).

5.2 Warcraft: Delayed Feedback with Discrete Decisions

Can memory help discrete decisions? The broader benchmark's lowest mean gaps belong to the prediction-only 2-Stage MSE baseline (Appendix E). Fixing optimizer and damping isolates trans-

Table 2: Warcraft tail decision-cost gap with drift 0.05, Adam, and $\beta = 1$. The comparison isolates transport. Gain is the percentage reduction in the mean; Δ is on minus off with an unadjusted paired 95% interval across ten seeds. Lower is better.

d	Transport off	Transport on	Gain (%)	Paired Δ [95% CI]
1	4.240	4.122	2.77	$-0.118\ [-0.284, 0.049]$
5	3.689	3.294	10.71	$-0.395\ [-0.560, -0.230]$
10	3.366	3.030	9.98	$-0.336\ [-0.533, -0.139]$
50	2.964	2.801	5.50	$-0.163\ [-0.238, -0.088]$

port's contribution. On official 12×12 Warcraft maps, a shared tile predictor supplies costs to a shortest-path solver. We freeze the blackbox cost direction computed at the arrival prediction (Vlastelica et al., 2020) (perturbation 20); reuse updates only the predictor derivative. Over 5000 rounds and ten paired seeds, we cross SGD/Adam, transport off/on, and $\beta = 0/1$ at rate 0.001, delays $\{0, 1, 5, 10, 50\}$, and stationary or drifting log costs (volatility 0.05). We report the mean decision-cost gap to the current shortest path over the final 200 rounds.

With SGD, enabling transport and damping increases the gap by 13–146% under drift. Holding Adam and damping fixed, transport adds a 5.50–10.71% reduction at delays 5, 10, and 50 (Table 2); the $d = 1$ contrast is inconclusive. Turning on both transport and damping instead gives 14.6–41.4% reductions relative to undamped transport-off Adam. Stationary-cost results are mixed (Appendix E, Table E.4). Warcraft thus tests discrete transport–optimizer interactions with blackbox gradients, beyond the smooth theory.

5.3 Sinkhorn: Measuring Gradient-Memory Increments

Here *optimal transport* is the decision task: assigning mass under predicted costs. *Gradient transport* is our learning update; the two uses of "transport" are distinct. This differentiable decision task lets us measure buffered gradient changes and decision quality in the same setting. A predictor maps 20-dimensional features to 10×10 costs; ten Sinkhorn iterations with entropy 0.05 produce the plan (Cuturi, 2013). True costs drift through an Ornstein–Uhlenbeck process with volatility 0.05 and mean reversion 0.02. We run 1000 rounds for $d \in \{0, 1, 5, 10, 50\}$. The SGD/Adam, transport off/on, and $\beta = 0/1$ factorial shares rate 0.001, initialization, and environmental trace within each seed. The frozen adapter caches the decision-loss derivative at the source prediction; arrival and reuse combine it with the current predictor derivative, without Sinkhorn solves during reuse.

A4 controls memory through how strongly increments reinforce. We measure their alignment ratio $\rho_t = \|\sum_{s \in \mathcal{B}_t} u_{s,t}\|^2 / \sum_{s \in \mathcal{B}_t} \|u_{s,t}\|^2$ when the denominator is nonzero. The numerator squares the length of the combined correction; the denominator adds individual squared lengths. Near zero indicates cancellation; identical increments give $\rho_t = |\mathcal{B}_t|$. For damped Adam, the mean is 1.00–1.08 and the observed maximum is 2.35 across positive-delay cells. Ratios near one mean little net reinforcement on these trajectories, not a uniform guarantee. Section 5.4 then tests whether memory improves decision loss.

5.4 Isolating Transport Contribution with Adam

Frozen transport increases loss with undamped Adam. With damping, it reduces cumulative loss by 0.37–2.00% across all four positive delays; all four paired tests pass the correction for multiple comparisons (Holm). These are neural Adam results; Theorem 1 covers the frozen Euclidean update. Does full re-evaluation change the empirical conclusion?

Oracle and schedule interact. A study of 400 runs crosses frozen/full oracles, transport off/on, and $\beta = 0/1$ with Adam on ten paired seeds. Undamped full transport reduces loss by 5.72–9.82% at $d = 5, 10, 50$ against full arrival-only re-evaluation; damping gives 2.13%, 1.78%, and 0.61%. The base rate is fixed across schedules. Our planned test was whether full re-evaluation improves on frozen transport with damping. It failed its success criterion: Holm $p = 0.063$ at $d = 10$ and mean gain -0.25% at $d = 50$. The benefit without damping emerged on outcomes already examined during development, so we tested it next on independent random realizations.

Table 3: Fresh Sinkhorn confirmation: undamped full re-evaluation Adam+IGT versus arrival-only re-evaluation. Gain: mean paired reduction in cumulative pre-update loss (%); pointwise 95% t interval, $n = 10$. Time: median paired training-update time ratio; equal feedback horizons.

Delay	Gain (%)	95% interval	Holm p	Time ratio
5	6.31	[5.54, 7.08]	1.79×10^{-8}	$5.09\times$
10	8.57	[7.76, 9.38]	3.59×10^{-9}	$9.23\times$
50	10.81	[9.97, 11.66]	1.03×10^{-9}	$40.28\times$

Fresh confirmation of full transport. Before generating new data, we fixed full re-evaluation Adam+IGT versus Adam that recomputes gradients only for newly arrived feedback (arrival-only). Both use rate 0.001 without damping: ten fresh paired seeds, $d = 5, 10, 50$, and 1000 rounds. All three pre-update comparisons meet the planned Holm-adjusted two-sided $p < 0.05$ and mean paired gain $\geq 1\%$ criteria (Table 3).

At $d = 50$, the 10.81% loss reduction costs $40.28\times$ the paired training-update time (median): 303–416 s versus 7.4–10.6 s over 1000 rounds on A5500/V100 GPUs. These minutes of learning may be affordable between infrequent decisions (Appendix F.3). Both arms do the same work per newly arrived example; full transport also re-solves buffered examples (47,175 versus 950 training Sinkhorn evaluations). Frozen transport avoids those extra solves.

5.5 DOOMD: A Modern Delayed-Online Comparison

Question. How does IGT compare with a modern delayed-online baseline across tested rates? Delayed optimistic online mirror descent (DOOMD) (Flaspohler et al., 2021) predicts missing gradients from the most recent one (*optimism*); a zero prediction gives delayed OGD. Using the same Sinkhorn inputs and hardware, full re-evaluation Adam+IGT reduces loss by 39.0–45.5% across the four-rate grid fixed before these runs. A tenfold upper-rate extension leaves 18.2–26.6% gains over the best tested cells, all hint-free (Table D.1). The best rate remains at the grid edge; these comparisons reuse the confirmation's random realizations and establish tested-rate performance (Appendix D.4).

Where does the gap arise? At rate 0.001 without damping, delayed SGD, Adam with decision-time gradients, arrival-re-evaluating Adam, and full-transport Adam form three comparisons: optimizer, freshness, transport (Table D.5). Holding Adam fixed, decision-time gradients incur 2.6%, 5.3%, and 19.9% extra loss at $d = 5, 10, 50$ relative to arrival re-evaluation: empirical staleness amplification in this decision problem. Buffered transport yields the separate 6.31–10.81% gain. These conditional percentages use different reference losses and cannot be added. Successive gradients are nearly orthogonal (mean cosine 0.014–0.016); the recent-gradient prediction changes loss by less than 1% across the initial grid despite much larger steps. This suggests weak predictive value for this hint; the observation alone does not identify a cause.

Additional experiments. Replay studies cover solver budgets, variable delays, and anchored Adam (Appendices G–H). Scoring timing does not explain the complete delay-50 gain; other delays remain partial (Appendix F.4). Appendix I expands the literature discussion.

6 Conclusion

Frozen IGT-OMD avoids inner re-solves; full re-evaluation yields 6.31–10.81% fresh matched Sinkhorn gains with undamped Adam. Factorials reveal oracle–damping interactions; regret and stability analyses support the frozen Euclidean update. **Limitations and future work.** Regret assumes stationary losses, constant delay, bounded trajectories and quadratic growth; Adam and full re-evaluation are empirical extensions. Stability requires common positive-definite curvature; LQR is a finite-horizon, one-step proxy. Factorials fix nominal rates; confirmation fixes feedback horizons. Tuned, compute-matched comparisons, broader guarantees, and cost-effective re-evaluation schedules remain open. Delayed feedback can remain useful after arrival: frozen reuse limits computation; full re-evaluation can improve decisions further.

REPRODUCIBILITY STATEMENT

Appendix C supplies complete assumptions and proofs. Appendices D and F specify clocks, oracle variants, baseline equations, model architectures, data, random seeds, metrics, and search rules. The artifact pipeline validates saved runs and generates tables and figures without retraining. It records code identity and input hashes separately from the manuscript build. Numerical regression tests check the stated restricted recurrence and regret-bound accounting; they are not substitutes for proofs.

ETHICS STATEMENT

The experiments use synthetic control and transport problems and a public shortest-path image dataset. They do not involve human participants or personal data. A controller or planner built from this research would require application-specific safety checks: our finite-horizon stability criterion and restricted theory are not deployment safety guarantees. Comparisons report negative results as well as gains, and distinguish decision cost from regret. In deployment, cached feedback, inner decisions and adjoints may contain sensitive information even when raw records are not retained; access control and retention limits would therefore be needed. Full re-evaluation also increases training computation. We report that cost alongside decision-quality gains rather than treating a fixed-horizon gain as resource efficiency.

DISCLOSURE OF AI USE

Generative AI tools were used to help draft, edit, and improve the clarity of the manuscript; retrieve and summarize related literature. The authors are responsible for the final text, mathematical claims, code, and reported results.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145, 2017. URL https://proceedings.mlr.press/v70/amos17a.html.

Sébastien M. R. Arnold, Pierre-Antoine Manzagol, Reza Babanezhad, Ioannis Mitliagkas, and Nicolas Le Roux. Reducing the variance in online optimization by transporting past gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/1dba5eed8838571e1c80af145184e515-Abstract.html.

Davoud Ataee Tarzanagh, Parvin Nazari, Bojian Hou, Li Shen, and Laura Balzano. Online bilevel optimization: Regret analysis of online alternating gradient methods. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2854–2862, 2024. URL https://proceedings.mlr.press/v238/ataee-tarzanagh24a.html.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9508–9519, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6bb56208f672af0dd65451f869fedfd9-Paper.pdf.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. doi: 10.1016/0041-5553(67)90040-7.

Aymeric Capitaine, Maxime Haddouche, Eric Moulines, Michael I. Jordan, Etienne Boursier, and Alain Durmus. Online decision-focused learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4e14a2a225d5d186b7eb1ecc7605bc2d-Abstract-Conference.html. Also available as arXiv:2505.13564.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://papers.neurips.cc/paper_files/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

Asen L. Dontchev and R. Tyrrell Rockafellar. *Implicit Functions and Solution Mappings: A View from Variational Analysis*. Springer, 2 edition, 2014. doi: 10.1007/978-1-4939-1037-3.

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

Genevieve Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3363–3373, 2021. URL https://proceedings.mlr.press/v139/flaspohler21a.html.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1568–1577, 2018. URL https://proceedings.mlr.press/v80/franceschi18a.html.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018. URL https://arxiv.org/abs/1802.02246.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3748–3758, 2020. URL https://proceedings.mlr.press/v119/grazzi20a.html.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. doi: 10.6028/jres.049.044.

Jiatai Huang, Yan Dai, and Longbo Huang. Banker online mirror descent: A universal approach for delayed online bandit learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13814–13844, 2023. URL https://proceedings.mlr.press/v202/huang23e.html.

Arieh Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511995569.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4882–4892, 2021. URL https://proceedings.mlr.press/v139/ji21c.html.

Pooria Joulani, András György, and Csaba Szepesvári. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1453–1461, 2013. URL https://proceedings.mlr.press/v28/joulani13.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Sen Lin, Daouda Sow, Kaiyi Ji, Yingbin Liang, and Ness Shroff. Non-convex bilevel optimization with time-varying objective functions. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5ee60ca5686bbcf756e56a6c75e66f32-Abstract-Conference.html.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024. doi: 10.1613/jair.1.15320. URL https://www.jair.org/index.php/jair/article/view/15320.

H. Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, 18(90):1–50, 2017. URL https://jmlr.org/papers/v18/14-428.html.

Parvin Nazari, Bojian Hou, Davoud Ataee Tarzanagh, Li Shen, and George Michailidis. Stochastic regret guarantees for online zeroth- and first-order bilevel optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2511.01126.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 737–746, 2016. URL https://proceedings.mlr.press/v48/pedregosa16.html.

Hao Qiu, Emmanuel Esposito, and Mengxiao Zhang. Exploiting curvature in online convex optimization with delayed feedback. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50448–50479, 2025. URL https://proceedings.mlr.press/v267/qiu25a.html.

Hao Qiu, Mengxiao Zhang, and Juliette Achddou. Decentralized online convex optimization with unknown feedback delays. *arXiv preprint arXiv:2601.07901*, 2026. URL https://arxiv.org/abs/2601.07901.

Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/72da7fd6d1302c0a159f6436d01e9eb0-Abstract.html.

Alexander Ryabchenko, Idan Attias, and Daniel M. Roy. A reduction from delayed to immediate feedback for online convex optimization with improved guarantees. *arXiv preprint arXiv:2602.02634*, 2026. URL https://arxiv.org/abs/2602.02634.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkevoJSYPB.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019.

Siyuan Yu, Wei Chen, and H. Vincent Poor. Distributed stochastic gradient descent with staleness: A stochastic delay differential equation based framework. *arXiv preprint arXiv:2406.11159*, 2025. URL https://arxiv.org/abs/2406.11159. Version 2, February 26, 2025.

A NOTATION

Table A.1 collects the symbols used in the algorithm, proofs, and experiments. Symbols for an exact optimum, a frozen surrogate, and a finite-iteration oracle denote different objects. Unless stated otherwise, norms are Euclidean norms or their induced matrix norms.

Table A.1: Notation, grouped by role.

Symbol	Meaning
Problem, domains, and dimensions	
$T;\ t, s$	Number of source rounds; current round and source round.
p, q	Dimensions of outer parameters and inner decisions, respectively.
$\theta_t \in \mathbb{R}^p$	Predictor parameters before the update at round t.
$\Omega \subseteq \mathbb{R}^p$	Fixed convex comparison domain containing the outer trajectory.
$\mathcal{D}$	Mirror-update domain; here $\mathbb{R}^p$, distinct from the comparison set Ω.
$\mathcal{W} \subseteq \mathbb{R}^q$	Domain containing inner states, optima, and comparison segments.
w_t	Approximate inner decision executed at source round t.
$w_t^\star(\theta)$	Exact optimizer of the round-t inner problem at θ.
$f_t(w;\theta)$	Model-based inner objective.
$\ell_t(w,\theta)$	Loss used to evaluate an executed decision.
$F_t(\theta)$	Exact-inner envelope $\ell_t(w_t^\star(\theta),\theta)$. Stationary analysis writes f, ℓ, F.
$\theta^\star$	Fixed hindsight minimizer over Ω; in the stationary analysis, a minimizer of F.
Feedback clock and arrived memory	
$d_s;\ a_s = s + d_s$	Feedback delay and arrival round of source item s.
$A_t;\ m_t = \lvert A_t\rvert$	Items arriving at round t and their number.
Q_t	Outstanding sources after processing arrivals: $\{s \le t : a_s > t\}$.
$\bar{q}_t$	Running maximum $\max_{1\le j\le t}\lvert Q_j\rvert$; zero before the first round.
$\mathcal{B}_t;\ b_t = \lvert\mathcal{B}_t\rvert$	Arrived items retained at the start of round t, before new admissions; their number.
$B_{\max}$	Capacity of the arrived buffer; not a bound on item age.
$d;\ D_{\mathrm{delay}}$	Constant delay in the regret theorem; an upper delay bound used in the queue-age discussion.
Δ_t	Outer displacement $\theta_{t+1} - \theta_t$, with $\Delta_0 = 0$.
Derivatives, accuracy, and sensitivity	
$H(w,\theta)$	Inner Hessian $\nabla^2_{ww} f(w;\theta) \in \mathbb{R}^{q\times q}$.
$J(w,\theta)$	Mixed derivative $\nabla^2_{w\theta} f(w;\theta) \in \mathbb{R}^{q\times p}$.
$a(w,\theta), b(w,\theta)$	Outer partial derivatives $\nabla_\theta \ell(w,\theta)$ and $\nabla_w \ell(w,\theta)$.
$\mathcal{H}(\theta, w)$	Recomputed-adjoint map $a - J^\top H^{-1} b$; see (13).
$\mu_w;\ L_w$	Lower inner-curvature bound; upper Hessian bound when used to establish inner contraction.
C_J	Uniform bound on $\lVert J(w,\theta)\rVert$.
M_a, M_b	Bounds on $\lVert a\rVert, \lVert b\rVert$ in the sufficient derivative conditions.
$L_{a,w}, L_{b,w}$	Lipschitz constants of a, b with respect to the inner state.
$L_{J,w}, L_{H,w}$	Lipschitz constants of J, H with respect to the inner state.
$L_{a,\theta}, L_{J,\theta}$	Lipschitz constants of a, J with respect to outer parameters.
L_g	Inner-state Lipschitz constant of $\mathcal{H}$; a sufficient value is given in (14).
L_S	Outer-parameter Lipschitz constant of a realized frozen surrogate.
$\kappa_w = C_J/\mu_w$	Sensitivity of the exact inner solution to outer parameters; not the inner condition number L_w/μ_w.
$\kappa = L_g\kappa_w$	Sensitivity coefficient in the source-to-arrival drift bound.
$\epsilon;\ \delta$	Uniform inner-state distance tolerance; arrival-adjoint residual tolerance.
$r_s^v;\ \widehat{v}_s$	Residual vector $H\widehat{v}_s - b$ and approximate arrival adjoint.
L_S^δ	Frozen-surrogate Lipschitz bound when arrival adjoints have residual at most δ.

Continued on the next page.

Table A.1 continued.

Symbol	Meaning
L_F	Envelope-gradient Lipschitz constant, used only in the conditional source-frozen comparison.
Oracles, update, and work	
H_s, J_s, v_s	Source-indexed derivative maps and arrival adjoint. Only v_s, evaluated at (w_s, θ_{a_s}), is frozen; derivative maps retain their arguments.
$g_s(\theta)$	Frozen-inner-state, frozen-adjoint surrogate in (5).
$u_{s,t}$	One old item's increment $g_s(\theta_t) - g_s(\theta_{t-1})$.
$c_t;\ h_t$	Sum of old-buffer increments; complete estimator (arrivals plus c_t).
ρ	Uniform alignment bound in (A4); $\rho = B_{\max}$ is always sufficient for a nonempty buffer.
η_0, η_t, β	Nominal rate, applied rate $\eta_0/\sqrt{1+\beta\bar{q}_t}$, and nonnegative damping coefficient.
$K;\ w_s^K(\theta)$	Number of inner iterations; decision obtained by those iterations at current parameters.
$g_s^{\mathrm{full,K}}(\theta)$	Gradient through the full K-step solver, defined in Section 3; not an exact-optimum gradient.
$\widehat{C}_\theta(x_s), r_s$	Predicted cost and cached cost-space direction in the task-specific frozen adapter.
C_g, C_a, C_K	Costs of one frozen gradient, constructing one arrival cache, and a full K-step forward/backward evaluation.
C_{src}, C_v	Cost of executing the current inner decision and of one adjoint solve.
$C_{\mathrm{inner}}(K_{\mathrm{ref}})$	Work of re-solving an available item's inner problem to a reference budget.
ψ, D_ψ	Mirror map and its Bregman divergence, defined in Appendix B.
Regret and inner-solver budget	
$R_T^{\mathrm{dec}}, R_T^{\mathrm{env}}$	Executed-decision and exact-inner envelope regret against the same fixed comparator.
E_T	Execution gap, with $R_T^{\mathrm{dec}} = R_T^{\mathrm{env}} + E_T$.
D	Bound on $\|\theta_t - \theta^\star\|$, not automatically the diameter of Ω.
G, G_F, G_a	Bounds on the full estimator, envelope gradient, and one arrival gradient, respectively.
μ_F	Quadratic-growth constant of the stationary envelope on the stated trajectory/domain.
M, L_ℓ	Bound on $\|\nabla_w \ell(w^\star(\theta_t), \theta_t)\|$ and inner-state gradient Lipschitz constant of ℓ.
C_{exec}	Additional quadratic execution-gap constant, when that stronger premise is imposed.
$N = T - d;\ c$	Number of active rounds at constant delay; positive scale in $\eta_0 = c/\sqrt{T}$.
$R_w;\ \gamma$	Uniform initialization radius and per-iteration contraction in Theorem 3.
$e_t^{\mathrm{in}};\ r = \gamma^K$	Warm-start decision error and contraction over K inner iterations.
$R_{\mathrm{init}};\ V$	First inner initialization radius and bound on outer step length in the warm-start analysis.
Restricted stability and discretization	
$C;\ \lambda$	Common positive-definite surrogate curvature and one of its eigenvalues. C is not H_s.
B	Fixed old-buffer size in the common-curvature recurrence.
$x_n;\ x(u)$	Scalar eigenmode of the discrete update and its continuous counterpart.
$h;\ u = n\eta$	Continuous memory lag and slow time; the Euler identity uses $h = \eta$, not the feedback delay d.
a, b	Proof-local scalar coefficients $\lambda(1+B), \lambda B$; not the derivative maps $a(w,\theta), b(w,\theta)$.
$U;\ M_E$	Fixed comparison-time horizon and uniform derivative Lipschitz bound for the Euler estimate.
r_-, r_+	Characteristic roots used to test discrete stability.

Continued on the next page.

Table A.1 continued.

Symbol	Meaning
γ, M (DDE proof)	Proof-local exponential decay rate and weighted history bound; distinct from the solver's contraction and execution constant.
Experimental metrics and abbreviations	
$C_t;\ P_K(\widehat{C}_t)$	Round-t true cost matrix and plan from K Sinkhorn iterations.
$\varepsilon_{\mathrm{OT}}$	Entropy coefficient of the Sinkhorn problem; distinct from inner error ϵ and Adam's numerical epsilon.
ρ_t	Observed squared norm of summed increments divided by their sum of squared norms; not a certified uniform ρ.
$S_T^{\mathrm{pre}}, S_T^{\mathrm{post}}$	Cumulative decision costs scored before and after the outer update; neither is fixed-comparator regret.
r_i (fresh study)	Seedwise percentage reduction in pre-update cost relative to the matched transport-off arm.
K (LQR)	Matrix-valued control gain, not the scalar inner iteration budget.
α, λ (D-FTRL)	D-FTRL base scale and regularizer weight, with $\alpha = 1/\lambda$; distinct from a curvature eigenvalue. Its gradient coefficient is α/t.
DFL; OCO	Decision-focused learning; online convex optimization.
IGT; OMD; FTRL	Implicit gradient transport; online mirror descent; follow-the-regularized-leader.
DDE; LQR; OT	Delay differential equation; linear quadratic regulator; optimal transport.

B BACKGROUND AND ALGORITHMIC COMPARISONS

This section connects the objects in the paper without requiring prior familiarity with bilevel optimization, mirror descent, or delay equations. Table A.1 is the symbol reference; the formal result dependencies are summarized in Table C.1.

B.1 WHY A BILEVEL GRADIENT CONTAINS AN ADJOINT

A bilevel learner does not choose its final decision directly. It first chooses parameters θ, and an inner optimization problem returns $w^\star(\theta)$. Even when the outer loss has no direct dependence on θ, changing θ changes this decision. Differentiating the inner stationarity equation $\nabla_w f(w^\star(\theta);\theta) = 0$ gives

$$H\, D_\theta w^\star + J = 0, \qquad D_\theta w^\star = -H^{-1}J.$$

The chain rule then gives $\nabla F = a - J^\top H^{-1} b$. Rather than form an inverse, solve $Hv = b$ and evaluate $a - J^\top v$. The implicit-function viewpoint and the regularity of solution maps are developed by Dontchev & Rockafellar (2014); Appendix C.2 derives the particular bounds used here.

Delayed feedback separates three parameter values: the source value that produced the decision, the arrival value at which its feedback is processed, and a later value at which the item is reused. Frozen IGT holds the source inner decision and arrival adjoint fixed during that later reuse. Full re-evaluation instead differentiates the finite inner computation again. Both use gradient differences in memory; they retain different approximations and pay different costs.

B.2 MIRROR DESCENT AND THE EUCLIDEAN UPDATE

For a differentiable strongly convex mirror map ψ, the Bregman divergence is

$$D_\psi(\theta, z) = \psi(\theta) - \psi(z) - \langle \nabla\psi(z), \theta - z\rangle \quad \text{(Bregman, 1967)}.$$

It measures the gap above a first-order approximation of ψ; it need not be symmetric. Mirror descent balances a linearized loss against that distance:

$$\theta_{t+1} \in \arg\min_{\theta\in\mathcal{D}} \{\eta_t \langle h_t, \theta\rangle + D_\psi(\theta, \theta_t)\} \quad \text{(Beck \& Teboulle, 2003)}.$$

For $\psi(\theta) = \|\theta\|^2/2$ and the unconstrained update domain $\mathcal{D} = \mathbb{R}^p$, the first-order condition gives $\theta_{t+1} = \theta_t - \eta_t h_t$. This is Algorithm 1's update. The comparison domain Ω in the regret analysis is not a projection domain; boundedness of the unprojected trajectory there is a premise. Other mirror maps require their own potential inequality and norm conditions. Adam adds a stateful preconditioner and moment estimates, so using the same estimator does not transfer the Euclidean regret proof.

B.3 WHAT THE TRANSPORT CORRECTION CHANGES

In a route-planning example, the source model predicts the costs used to choose a path. Its feedback later reveals the consequence of that choice. Reusing the feedback at a new model can mean three things: (1) apply the stored gradient unchanged; (2) hold a decision-gradient direction fixed while differentiating the current predictor; or (3) recompute the finite decision layer at the current prediction. The last two choices can each be used with or without the same old-buffer correction. Thus *transport on versus off* and *frozen versus full oracle* are distinct comparisons. Table B.1 makes their work explicit.

Let C_{src} be the cost of the current executed inner decision and $m_t = |A_t|$, $b_t = |\mathcal{B}_t|$. The entries below give additional gradient work beyond C_{src} and an $O(p)$ Euclidean update. They exclude queue bookkeeping, diagnostic logs and task-specific data storage.

A source-time gradient involving an unknown outcome cannot be computed before that outcome arrives. The stored-gradient comparison therefore presupposes delayed gradient delivery or later evaluation at saved source parameters. It is not an extra-information baseline for an experiment that reveals only outcomes. The LQR delayed-OGD and D-FTRL baselines specify their source-gradient oracle explicitly in Appendix D.3.

For the formal frozen-adjoint construction, an idealized full re-solving comparison would compute a new inner optimum and a new adjoint for each available item. If their costs are $C_{\mathrm{inner}}(K_{\mathrm{ref}})$ and C_v,

Oracle/control	Information recomputed after delivery	Gradient work per clock
Stored source gradient	No gradient re-evaluation; sum delivered gradients	$O(m_t p)$, plus source-gradient construction when its required information is available
Frozen, transport off	Arrival cache only; retain no old items	$m_t C_a$
Frozen, transport on	Arrival cache plus current frozen surrogates on old items	$m_t C_a + b_t C_g$
Full K-step, off	Current predictor and K-step solver for arrivals	$m_t C_K$
Full K-step, on	Same computation for arrivals and old items	$(m_t + b_t) C_K$

Table B.1: Oracle and work comparison. C_a includes arrival-gradient/cache construction; C_g is a frozen-gradient evaluation; C_K includes the full finite-solver forward/backward computation. Transport-on rows sum old gradient differences; transport-off rows use arrivals only.

its gradient work is $(m_t + b_t)\{C_{\mathrm{inner}}(K_{\mathrm{ref}}) + C_v + C_g\}$. This symbolic comparison is not the same oracle as differentiating K solver steps, and it is not a measured speedup. The actual full-Sinkhorn costs appear in Table F.4.

B.4 Continuous and discrete stability

A delay differential equation specifies a derivative using both the present state and its history. It can clarify how memory interacts with curvature, but a digital update still advances in finite steps. Here the common-curvature recurrence has one-clock memory. Setting the continuous lag to that step length gives its exact forward-Euler form. Proposition 1 proves stability of that continuous family; Proposition 2 separately finds the step sizes for which both discrete characteristic roots lie inside the unit disk. Lemma 6 then bounds the approximation error over a fixed time interval. These are three complementary statements, not three ways to assume that continuous stability automatically survives discretization.

C COMPLETE ASSUMPTIONS AND PROOFS

This appendix analyzes Algorithm 1, not a replay algorithm that re-solves every old inner problem. It uses an arrival-time adjoint and an unprojected Euclidean update. We give deterministic statements; a stochastic version would require the corresponding conditional-moment assumptions.

C.1 ASSUMPTIONS AND NOTATION

Use the same fixed comparison set Ω as in (3). Let $\Omega \subseteq \mathbb{R}^p$ be convex and contain $\theta^\star$ and all iterates under consideration. Let $\mathcal{W} \subseteq \mathbb{R}^q$ be convex and contain all executed inner states, exact inner solutions, and segments used below. The stationary inner problem is unconstrained and has a unique optimizer $w^\star(\theta)$. For $w \in \mathcal{W}$ and $\theta \in \Omega$, write

$$H(w,\theta) = \nabla^2_{ww} f(w;\theta), \qquad J(w,\theta) = \nabla^2_{w\theta} f(w;\theta),$$
$$a(w,\theta) = \nabla_\theta \ell(w,\theta), \qquad b(w,\theta) = \nabla_w \ell(w,\theta).$$

and define the recomputed-adjoint map

$$\mathcal{H}(\theta, w) = a(w,\theta) - J(w,\theta)^\top H(w,\theta)^{-1} b(w,\theta). \tag{13}$$

This map differs from a frozen g_s away from that item's arrival point.

(A1) Inner curvature and accuracy. f is twice continuously differentiable on an open neighborhood of $\mathcal{W} \times \Omega$, $H(w,\theta) \succeq \mu_w I$ for $\mu_w > 0$, $\|J(w,\theta)\| \le C_J$, and $\|w_t - w^\star(\theta_t)\| \le \epsilon$ for each source decision. The regularity and curvature hold wherever the comparisons below are made.

(A2) Gradient regularity. ℓ is continuously differentiable on an open neighborhood of $\mathcal{W} \times \Omega$, $\mathcal{H}(\theta,\cdot)$ is uniformly L_g-Lipschitz, and each realized frozen surrogate g_s is uniformly L_S-Lipschitz on Ω. The adjoints in (5) are exact. Appendix C.2 gives explicit sufficient derivative bounds.

(A3) Outer geometry and bounded trajectory. $F(\theta) = \ell(w^\star(\theta),\theta)$ is differentiable and convex on Ω and has a fixed minimizer $\theta^\star$ there. For every iterate,

$$F(\theta_t) - F(\theta^\star) \ge \frac{\mu_F}{2}\|\theta_t - \theta^\star\|^2, \qquad \|\theta_t - \theta^\star\| \le D, \qquad \|h_t\| \le G.$$

Here $\mu_F > 0$, and $\|\nabla F(\theta)\| \le G_F$ on the segments between $\theta^\star$ and the iterates. In particular these segments lie in Ω. The algorithm is not projected; remaining in this set is a trajectory premise.

(A4) Increment geometry. For some $\rho \ge 0$ and every actual start-of-round buffer,

$$\left\| \sum_{s \in \mathcal{B}_t} u_{s,t} \right\|^2 \le \rho \sum_{s \in \mathcal{B}_t} \|u_{s,t}\|^2.$$

When $B_{\max} \ge 1$, $\rho = B_{\max}$ always satisfies this inequality. For an empty buffer the two sides are zero, and we may use $\rho = 0$ if $B_{\max} = 0$. A smaller uniform constant is an optional additional premise.

For the regret theorem we further require $0 \le d < T$, $d_t = d$, $B_{\max} \le d$, an empty initial queue and buffer, exactly one source decision per round, and the clock in Algorithm 1. There are no external parameter updates before feedback arrives. The asymptotic notation in Theorem 1 fixes $c, \beta > 0$ and requires $D, G, G_F, \mu_F^{-1}, L_g, L_S, \kappa_w$ to stay uniformly bounded as T, d change. Buffer capacity and the worst-case $\rho = B_{\max}$ may grow with d; we do not require that choice of ρ to be delay-independent. These conditions cannot be inferred just from finite experimental traces.

One can impose (A3) on a fixed affine subspace $\theta^\star + S$ instead of on the ambient space, provided all iterates and comparison segments stay in it and convexity is understood on that subspace. The error bound below uses full norms and thus also bounds the error paired with a displacement in S. By contrast, a sequence of changing subspaces S_t does not guarantee a common quadratic-growth inequality relative to one comparator. No such implication is used here.

C.2 Deriving the sensitivity constants

Lemma 1 (Inner-solution and hypergradient sensitivity)**.** *Under (A1), $w^\star$ is $\kappa_w = C_J/\mu_w$-Lipschitz. If, in addition, $\|b\| \le M_b$ and a, J, H, b are respectively $L_{a,w}, L_{J,w}, L_{H,w}, L_{b,w}$-Lipschitz in w, then (A2)'s first bound holds with*

$$L_g = L_{a,w} + \frac{L_{J,w} M_b}{\mu_w} + \frac{C_J L_{b,w}}{\mu_w} + \frac{C_J M_b L_{H,w}}{\mu_w^2}. \tag{14}$$

If a and J are also $L_{a,\theta}$- and $L_{J,\theta}$-Lipschitz in θ, frozen surrogates satisfy $L_S = L_{a,\theta} + L_{J,\theta} M_b/\mu_w$.

Proof. Let $w_i = w^\star(\theta_i)$. Strong monotonicity of $\nabla_w f$ gives

$$\begin{aligned}\mu_w \|w_1 - w_2\|^2 &\le \langle \nabla_w f(w_1;\theta_1) - \nabla_w f(w_2;\theta_1), w_1 - w_2\rangle \\ &= \langle \nabla_w f(w_2;\theta_2) - \nabla_w f(w_2;\theta_1), w_1 - w_2\rangle \\ &\le C_J \|\theta_1 - \theta_2\| \, \|w_1 - w_2\|.\end{aligned}$$

If $w_1 = w_2$ the claim is immediate; otherwise divide by $\mu_w \|w_1 - w_2\|$. The mixed-derivative bound along the parameter segment justifies the final line.

Differentiating $\nabla_w f(w^\star(\theta);\theta) = 0$ gives $D_\theta w^\star = -H^{-1} J$. The chain rule therefore gives $\nabla F(\theta) = \mathcal{H}(\theta, w^\star(\theta))$. For two inner states at a fixed θ, use the inverse identity $H_1^{-1} - H_2^{-1} = H_1^{-1}(H_2 - H_1)H_2^{-1}$ to obtain

$$\|H_1^{-1} b_1 - H_2^{-1} b_2\| \le \left(\frac{L_{b,w}}{\mu_w} + \frac{L_{H,w} M_b}{\mu_w^2} \right) \|w_1 - w_2\|.$$

Now subtract the two expressions in (13), separating the change in J from the change in $H^{-1}b$. The direct term contributes $L_{a,w}$, the change in J contributes $L_{J,w} M_b/\mu_w$, and the remaining product contributes the last two terms of (14). Finally, every arrival adjoint has norm at most M_b/μ_w. Subtracting $a(w_s,\theta) - J(w_s,\theta)^\top v_s$ at two parameter vectors gives the claimed L_S. This proof requires the stated Lipschitz bounds; twice continuous differentiability alone does not supply them with uniform constants. □

C.3 Proof of Theorem 2

For $t > d$ the unique new arrival is $s = t - d$. By its arrival-time construction, $g_s(\theta_t) = \mathcal{H}(\theta_t, w_s)$. Thus

$$\begin{aligned}\|g_s(\theta_t) - \nabla F(\theta_t)\| &\le L_g \|w_s - w^\star(\theta_t)\| \\ &\le L_g \epsilon + L_g \kappa_w \|\theta_s - \theta_t\| \\ &\le L_g \epsilon + \kappa \sum_{k=t-d}^{t-1} \|\Delta_k\|.\end{aligned} \tag{15}$$

Every retained item was evaluated at the previous clock, including items admitted then, so its cached value is exactly $g_s(\theta_{t-1})$. By (A2) and (A4),

$$\|c_t\|^2 \le \rho \sum_{s \in \mathcal{B}_t} \|u_{s,t}\|^2 \le \rho |\mathcal{B}_t| L_S^2 \|\Delta_{t-1}\|^2.$$

Combining this with (15) proves (8). For $d = 0$, $B_{\max} = 0$, and the parameter-drift sum is empty. The proof does not replace the frozen surrogate by an exact hypergradient away from arrival, and it does not assert that the memory is a telescope.

For comparison, applying an exact but source-frozen envelope gradient would incur $\|\nabla F(\theta_s) - \nabla F(\theta_t)\| \le L_F \sum_{k=s}^{t-1} \|\Delta_k\|$ if F is L_F-smooth. The constants differ from (15), and the latter also has a memory term. Neither bound orders the two algorithms' errors pointwise.

C.4 EUCLIDEAN OMD AND PROOF OF THEOREM 1

Lemma 2 (Euclidean OMD with non-increasing rates)**.** *If $\theta_{t+1} = \theta_t - \eta_t h_t$, $\eta_t > 0$ is non-increasing, and $\|\theta_t - \theta^\star\| \le D$, then for $m \le n$,*

$$\sum_{t=m}^{n} \langle h_t, \theta_t - \theta^\star \rangle \le \frac{D^2}{2\eta_n} + \frac{1}{2} \sum_{t=m}^{n} \eta_t \|h_t\|^2.$$

Proof. Let $V_t = \|\theta_t - \theta^\star\|^2$. Expanding the update gives the exact identity $\langle h_t, \theta_t - \theta^\star \rangle = (V_t - V_{t+1})/(2\eta_t) + \eta_t \|h_t\|^2/2$. The first sum telescopes to

$$\frac{V_m}{2\eta_m} - \frac{V_{n+1}}{2\eta_n} + \frac{1}{2} \sum_{t=m+1}^{n} V_t(\eta_t^{-1} - \eta_{t-1}^{-1}) \le \frac{D^2}{2\eta_n}.$$

Non-increasing rates make each difference nonnegative; $V_{n+1} \ge 0$ suffices for the terminal term. This includes the queue-envelope schedule, whose denominator cannot decrease. □

The exact finite-horizon bound. Set $N = T - d$ and $\eta = \eta_0/\sqrt{1 + \beta d}$. Under (A1)–(A4),

$$\begin{aligned} R_T^{\mathrm{env}} \le\ & dG_F D + \frac{D^2}{\eta} + N\eta G^2 \\ & + \frac{6N}{\mu_F} \left[L_g^2 \epsilon^2 + \eta^2 G^2 \left(\kappa^2 d^2 + \rho L_S^2 B_{\max} \right) \right]. \end{aligned} \tag{16}$$

This is the bound whose order is stated in the main text.

Proof. In rounds $1, \dots, d$ there are no arrivals and no buffer items. Therefore $h_t = 0$ and $\theta_{t+1} = \theta_t$. For $t \ge d + 1$, exactly d items remain outstanding, so $\bar{q}_t = d$. Every nonzero update uses η, and

$$\|\Delta_t\| \le \eta G \quad \text{for every round, including burn-in.} \tag{17}$$

It follows from Theorem 2 and $(a + b + c)^2 \le 3(a^2 + b^2 + c^2)$ that

$$e_t^2 := \|h_t - \nabla F(\theta_t)\|^2 \le 3L_g^2 \epsilon^2 + 3\eta^2 G^2 (\kappa^2 d^2 + \rho L_S^2 B_{\max}), \qquad t > d. \tag{18}$$

Let $r_t = F(\theta_t) - F(\theta^\star)$. Convexity and Young's inequality yield

$$\begin{aligned} r_t &\le \langle h_t, \theta_t - \theta^\star \rangle + e_t \|\theta_t - \theta^\star\| \\ &\le \langle h_t, \theta_t - \theta^\star \rangle + \frac{e_t^2}{\mu_F} + \frac{\mu_F}{4} \|\theta_t - \theta^\star\|^2 \\ &\le \langle h_t, \theta_t - \theta^\star \rangle + \frac{e_t^2}{\mu_F} + \frac{r_t}{2}. \end{aligned}$$

The last step uses quadratic growth, not merely convexity. Rearrange and sum over $t = d + 1, \dots, T$. Lemma 2 with constant rate gives

$$\sum_{t=d+1}^{T} r_t \le \frac{D^2}{\eta} + N\eta G^2 + \frac{2}{\mu_F} \sum_{t=d+1}^{T} e_t^2.$$

For each burn-in round, the gradient bound along the comparison segment gives $0 \le r_t \le G_F D$. Insert (18) to obtain (16). Finally add the exact E_T from (3) to obtain decision regret; there is no substitution of executed losses in this calculation. □

With $\eta_0 = c/\sqrt{T}$, replacing N by T in positive terms gives

$$\begin{aligned} R_T^{\mathrm{env}} \le\ & dG_F D + \frac{D^2}{c} \sqrt{T(1 + \beta d)} + \frac{cG^2\sqrt{T}}{\sqrt{1 + \beta d}} + \frac{6L_g^2}{\mu_F} T\epsilon^2 \\ & + \frac{6c^2 G^2}{\mu_F} \frac{\kappa^2 d^2 + \rho L_S^2 B_{\max}}{1 + \beta d}. \end{aligned} \tag{19}$$

For fixed $\beta > 0$, $d^2/(1+\beta d) \le d/\beta$. The always-valid choice $\rho = B_{\max} \le d$ bounds the memory numerator by $L_S^2 d^2$. Since $0 \le d < T$, $d \le \sqrt{T(1+d)}$, so the additive $O(d)$ terms are absorbed in (7). A bounded coherence constant improves that part of the finite bound, but is not needed for the displayed asymptotic order under $B_{\max} \le d$. If $\beta = 0$, the proof instead gives $O(\sqrt{T} + T\epsilon^2 + d^2)$ with worst-case coherence. The condition $d^2 = o(T)$ is sufficient for that undamped bound, not for the damped one.

The distinction relies on constant delay and empty initialization. With variable delays, early nonzero updates can use larger rates than the eventual queue-envelope rate. Replacing all such steps by the final damped rate would not be valid. Likewise, a capacity growing independently of delay requires retaining its explicit term in (16).

The restriction $d < T$ excludes a different, simple boundary case. If $d \ge T$ and initialization is empty, no feedback arrives during the scored horizon, the buffer stays empty, and $\theta_t = \theta_1$ throughout. Under the same comparison-gradient and radius bounds, $R_T^{\mathrm{env}} = T[F(\theta_1) - F(\theta^\star)] \le T G_F D$. Executed regret still adds E_T. This is not a sublinear guarantee and does not use the active-round calculation with $N = T - d$.

When the full-estimator bound follows from derivative bounds. Assuming a uniformly bounded estimator need not hide an extra delay factor. Suppose, in addition to the sufficient conditions of Lemma 1, that $\|a(w,\theta)\| \le M_a$. Every arrival then has norm at most $G_a = M_a + C_J M_b/\mu_w$. On the constant-delay clock,

$$\|h_t\| \le G_a + B_{\max} L_S \eta \|h_{t-1}\|.$$

Before arrival the estimator is zero. If $B_{\max} L_S \eta \le r < 1$, induction therefore gives $\|h_t\| \le G_a/(1-r)$ for every active round. For fixed c, β, L_S, $B_{\max} \le d$, $\eta = c/\sqrt{T(1+\beta d)}$, and $d = o(T)$, this coefficient tends to zero. If G_a is also uniformly bounded in T, d, the bounded-estimator premise therefore follows with a uniform constant for sufficiently large T. In particular, uniform bounds on $M_a, C_J, M_b, \mu_w^{-1}$ suffice for this extra requirement. Instance-dependent finite derivative bounds alone would not establish it. Bounded outer trajectories remain a separate condition.

C.5 Proof of Theorem 3 and an execution-error example

If $\ell(\cdot,\theta)$ has L_ℓ-Lipschitz gradient and $\|\nabla_w \ell(w^\star(\theta),\theta)\| \le M$, Taylor's inequality gives

$$|\ell(w_t,\theta_t) - \ell(w^\star(\theta_t),\theta_t)| \le M\epsilon + \tfrac{1}{2} L_\ell \epsilon^2.$$

This proves the general execution-gap statement in the main text. In particular, $f(w;\theta) = (w-\theta)^2/2$ and $\ell(w,\theta) = w$ give a linear gap ϵ when $w = \theta + \epsilon$. Strong inner convexity alone therefore cannot justify a quadratic execution gap. A quadratic upper bound does follow if the first-order outer-loss term vanishes at the inner optimum, or under a separately stated orthogonality or expectation condition. None is assumed silently.

A solver-budget corollary without a quadratic execution-gap premise. The general Taylor bound yields $R_T^{\mathrm{dec}} = O(\sqrt{T(1+d)} + T\epsilon + T\epsilon^2 + d)$ under uniform M, L_ℓ and the other theorem conditions. Thus $d = o(T)$ and $\epsilon = o(1)$ already suffice for sublinearity. More specifically, $\epsilon = O(T^{-1/2})$ gives $R_T^{\mathrm{dec}} = O(\sqrt{T(1+d)} + d)$ without requiring the first-order execution term to vanish. If each inner solve starts within a uniform distance R_w of its exact optimum and its iteration map contracts that distance by a uniform $\gamma \in (0,1)$, then K iterations give $\epsilon \le R_w \gamma^K$. Taking

$$K \ge \max\left\{0, \left\lceil \frac{\log R_w + \frac{1}{2}\log T}{\log(1/\gamma)} \right\rceil \right\}$$

ensures $\epsilon \le T^{-1/2}$ when $R_w > 0$ (the case $R_w = 0$ needs no iterations). This is an $O(\log T)$ sufficient budget under explicit uniform contraction and initialization bounds. A warm-start implementation must justify its own R_w; inner strong convexity alone does not bound the drift of the starting point.

Example 1 (Execution error under a mandated inexact oracle)**.** *For any T and $\epsilon > 0$, there is a smooth strongly convex inner problem and an oracle with inner error ϵ such that every outer strategy forced to execute this oracle's output incurs $R_T^{\mathrm{dec}} \ge T\epsilon^2/2$, for any feedback delays.*

Proof. Take $\Omega = \mathbb{R}$, $f(w;\theta) = (w-\theta)^2/2$ and $\ell(w,\theta) = \theta^2/2 + (w-\theta)^2/2$. The exact inner optimum is $w^\star(\theta) = \theta$, so $F(\theta) = \theta^2/2$ and the fixed optimal comparator is $\theta^\star = 0$. Require the oracle to return $w_t = \theta_t + \epsilon\xi_t$, where each $\xi_t \in \{-1, 1\}$; no randomness is needed. Then for every strategy,

$$R_T^{\mathrm{dec}} = \sum_{t=1}^{T} \frac{\theta_t^2}{2} + \frac{T\epsilon^2}{2} \geq \frac{T\epsilon^2}{2}.$$

The identity is independent of arrival timing. Moreover the recomputed-adjoint map is exactly $\mathcal{H}(\theta, w) = \theta$, so this example isolates execution error from hypergradient bias. □

This is an existence statement for a mandated inexact execution oracle. It is not a lower bound against a learner permitted to replace the oracle with an exact solve, nor does it show that finite inner iteration counts always yield persistent error. To translate a tolerance requirement into a bound on inner iterations, one must separately bound the solver's convergence and warm-start drift. We make no general claim that those are independent of delay.

C.6 Proof of the DDE and discrete propositions

In a common-curvature eigenmode, the arrival term is λx_n and each of the B old-buffer increments is $\lambda(x_n - x_{n-1})$. Their sum gives (10). Diagonalizing the symmetric positive-definite C separates the vector update into these scalar modes.

DDE stability (Proposition 1). Write $a = \lambda(1+B)$ and $b = \lambda B$, so $a - b = \lambda > 0$. For $h = 0$, the equation is $x' = -\lambda x$ and the claim is immediate. For $h > 0$, the method of steps gives a unique solution for each continuous history: solve an inhomogeneous linear ODE on each successive interval of length h. For a fixed $h > 0$, choose $\gamma > 0$ small enough that $\gamma + b\exp(\gamma h) < a$; such a choice exists by continuity. Given a continuous history ϕ on $[-h, 0]$, define $M = \max_{s\in[-h,0]} e^{\gamma s}|\phi(s)|$. For $y(u) = e^{\gamma u}|x(u)|$, its upper right Dini derivative satisfies

$$D^+ y(u) \leq (\gamma - a)y(u) + be^{\gamma h} y(u-h).$$

The bound $y \leq M$ is invariant. Indeed, at a first time reaching $M + \delta$ from below, where $\delta > 0$, the differentiable positive quantity y must have nonnegative derivative. Its delayed value is at most $M + \delta$, so the displayed upper bound on the derivative is strictly negative, a contradiction. Letting $\delta \downarrow 0$ gives $y(u) \leq M$ for all $u \geq 0$. Consequently $|x(u)| \leq Me^{-\gamma u}$, proving exponential stability since $M \leq \|\phi\|_\infty$. The whole history segment satisfies $\sup_{s\in[u-h,u]} |x(s)| \leq e^{\gamma h} Me^{-\gamma u}$ as well. For a vector system, an orthogonal eigenbasis and the minimum of the finitely many modal decay rates give an exponential bound for the full state. Forward Euler at $u = n\eta$ with lag $h = \eta$ gives

$$x_{n+1} = x_n + \eta[-\lambda(1+B)x_n + \lambda B x_{n-1}],$$

which is (10). This is an exact discretization identity for a family of DDEs indexed by η, not a fixed-feedback-delay limit theorem.

Discrete stability (Proposition 2). Put $z = \eta\lambda$. The characteristic polynomial is

$$p(r) = r^2 - [1 - z(1+B)]r - zB.$$

For $z > 0$ and $B \geq 0$, its discriminant $[1 - z(1+B)]^2 + 4zB$ is nonnegative and the product of the roots is $-zB$. Thus the roots are real and can be ordered as $r_- \leq 0 \leq r_+$, including the zero root when $B = 0$. Now

$$p(1) = (1 - r_-)(1 - r_+) = z > 0$$

implies $r_+ < 1$, because $1 - r_- > 0$. Also

$$p(-1) = (-1 - r_-)(-1 - r_+) = 2 - z(1 + 2B) > 0$$

is equivalent to $r_- > -1$, because $-1 - r_+ < 0$. Both roots therefore lie strictly in the unit disk exactly when $0 < z < 2/(1+2B)$. At the upper endpoint a root is -1; at $z = 0$ a root is 1. If negative steps were allowed, $p(1) = z < 0$ and $p(r) \to +\infty$ as $r \to +\infty$ would imply a root above one. Repeated roots strictly inside the disk still decay: their possible nr^n term tends to zero.

Intersect the scalar intervals over eigenvalues of C to obtain (12). Conversely, outside that interval a nondecaying eigenmode provides initial data for which convergence fails. Substituting the queue-damped rate gives the nominal interval. The common positive-definite curvature, fixed buffer size, one-arrival clock, and Euclidean optimizer are essential restrictions. A finite initial buffer-filling phase changes the initial modal states, not the subsequent constant-coefficient criterion.

An exact constant-delay calibration identity. For any deterministic frozen oracle and constant delay d, two executions of Algorithm 1 with the same initial states, inputs, and random seeds produce identical iterates if their nominal rates obey $\eta_0^{(1)}/\sqrt{1+\beta_1 d} = \eta_0^{(2)}/\sqrt{1+\beta_2 d}$. Before arrival their outer states do not move; afterwards both have the same constant applied rate. Induction on the clock then matches decisions, caches, estimators, and updates. This identity does not require a quadratic objective. It explains why the LQR β ablation must show both nominal and applied rates. It does not apply unchanged to variable delays or to an optimizer with independent warm-up updates.

Executable checks. Direct tests compare the implemented clock with the recurrence, test roots below and above the Schur boundary, verify constant-delay calibration on a nontrivial bilevel surrogate, and check every inequality in the finite regret bound on small trajectories. These are regression checks of the derivation and implementation; the proofs above, not numerical tests, establish the general statements.

C.7 Queue timing and variable arrivals

Outstanding age is not arrived-buffer age. The queue definition gives $t - s < d_s$ for $s \in Q_t$, whereas an arrived item has $t - s \geq d_s$. Under constant delay, a nonempty full start-of-round buffer of capacity $B_{\max}$ contains sources of ages $d+1, \ldots, d+B_{\max}$. A capacity bound therefore controls count, not age. With irregular arrivals, an item can remain for arbitrarily long if no later item arrives to displace it. If one source is registered per round and $d_s \leq D_{\text{delay}}$, then $|Q_t| \leq D_{\text{delay}}$: only sources with $s > t - D_{\text{delay}}$ can be outstanding. A delay percentile is not a deterministic upper bound.

Lemma 3 (Estimator error with variable arrivals)**.** *Retain stationarity, (A1), (A2), (A4), and the cache clock of Algorithm 1, but allow arbitrary nonnegative integer delays. Suppose $\|\nabla F(\theta_t)\| \leq G_F$ at the iterates. Writing $m_t = |A_t|$, the estimator satisfies*

$$\|h_t - \nabla F(\theta_t)\| \leq m_t L_g \epsilon + \kappa \sum_{s \in A_t} \|\theta_s - \theta_t\| + |m_t - 1| G_F + L_S \sqrt{\rho |\mathcal{B}_t|} \|\Delta_{t-1}\|. \tag{20}$$

Outer convexity and quadratic growth are not needed for this error bound.

Proof. Add and subtract $m_t \nabla F(\theta_t)$ in the arrival sum. Each arrived item uses its exact arrival adjoint, so the same sensitivity argument as (15) gives

$$\|g_s(\theta_t) - \nabla F(\theta_t)\| \leq L_g \epsilon + \kappa \|\theta_s - \theta_t\|.$$

The remaining arrival term is $(m_t - 1)\nabla F(\theta_t)$. The old-buffer increment bound in Appendix C.3 is unchanged: every retained item's previous evaluation occurred at the preceding clock. The triangle inequality proves (20), including $m_t = 0$. □

The multiplicity term matters because the estimator sums arrivals, whereas the comparison target is one current gradient. Since $A_t \subseteq Q_{t-1} \cup \{t\}$, one has $m_t \leq |Q_{t-1}| + 1$; m_t need not be bounded by the post-arrival $|Q_t|$. If (A3) also holds and the positive step sizes are non-increasing, the OMD–Young argument gives

$$R_T^{\text{env}} \leq \frac{D^2}{\eta_T} + \sum_{t=1}^{T} \eta_t G^2 + \frac{2}{\mu_F} \sum_{t=1}^{T} \|h_t - \nabla F(\theta_t)\|^2. \tag{21}$$

Indeed convexity, Young's inequality and quadratic growth imply $r_t/2 \leq \langle h_t, \theta_t - \theta^\star \rangle + \|h_t - \nabla F(\theta_t)\|^2/\mu_F$, where $r_t = F(\theta_t) - F(\theta^\star)$; sum and apply Lemma 2. Equation (21) is a conditional finite bound, not an arbitrary-delay sublinear-regret theorem. In particular, earlier nonzero steps cannot be bounded by replacing their rates with the smaller final rate.

Memory geometry and the meaning of an error bound. For any m vectors, Cauchy–Schwarz gives $\|\sum_i v_i\|^2 \le m\sum_i \|v_i\|^2$. When the denominator is nonzero, the ratio lies in $[0, m]$. Aligned equal vectors attain m, while cancellation can give zero. The upper bound is not a measured cancellation factor or an algorithmic gain. For example, $f(w;\theta) = (w-\theta)^2/2$ and $\ell(w,\theta) = \lambda\theta^2/2$, $\lambda > 0$, give the exact arrival gradient $g_s(\theta) = \lambda\theta$ for every source state. A full buffer adds $B\lambda\Delta_{t-1}$ to that exact gradient. Thus transport need not reduce arrival-gradient error on every problem. Its positive empirical effect on other objectives is a separate statement, which the matched experiments test.

C.8 Residual-controlled approximate adjoints

Exact linear solves are a convenient starting point for analysis. Iterative solvers, including conjugate gradients (Hestenes & Stiefel, 1952), instead provide an approximate vector whose residual can be checked. The next result makes that residual's effect explicit.

Lemma 4 (Adjoint residual contribution)**.** *Assume (A1), the L_g bound in (A2), and the sufficient bounds $\|b\| \le M_b$, $L_{a,\theta}$ and $L_{J,\theta}$ from Lemma 1. Replace the exact-adjoint part of (A2) by an arrival vector $\widehat{v}_s$ with residual*

$$r_s^v = H(w_s, \theta_{a_s})\widehat{v}_s - b(w_s, \theta_{a_s}), \qquad \|r_s^v\| \le \delta,$$

and freeze $\widehat{v}_s$ after arrival. Its arrival-gradient error is at most

$$L_g\epsilon + \kappa\|\theta_s - \theta_{a_s}\| + C_J\delta/\mu_w.$$

The approximate frozen surrogate is Lipschitz with constant

$$L_S^\delta = L_{a,\theta} + L_{J,\theta}(M_b + \delta)/\mu_w.$$

Proof. The exact adjoint v_s obeys $\widehat{v}_s - v_s = H^{-1}r_s^v$. Since $H \succeq \mu_w I$, $\|\widehat{v}_s - v_s\| \le \delta/\mu_w$. Multiplication by $J^\top$ contributes at most $C_J\delta/\mu_w$ to the arrival error. Add the exact-adjoint arrival bound. Also $\|\widehat{v}_s\| \le (M_b+\delta)/\mu_w$. Subtracting $a(w_s,\theta) - J(w_s,\theta)^\top\widehat{v}_s$ at two parameters and using the stated Lipschitz bounds proves the expression for L_S^δ. □

If (A3), (A4) and the constant-delay conditions hold for this modified execution, the finite bound (16) remains valid with

$$L_g^2\epsilon^2 \;\longmapsto\; (L_g\epsilon + C_J\delta/\mu_w)^2, \qquad L_S \;\longmapsto\; L_S^\delta.$$

The estimator still has three error terms: combined inner/adjoint error, source drift, and memory. The same squared-sum and OMD arguments apply. For fixed $\beta > 0$, $\eta_0 = c/\sqrt{T}$, $B_{\max} \le d < T$, uniformly bounded primitive constants, and bounded δ, use $(x+y)^2 \le 2x^2+2y^2$ in both replacements. Besides constant multiples of the original terms, this gives $O(T\delta^2)$ from the arrival error and $O(d\delta^2) \le O(T\delta^2)$ from the memory term, even with $\rho = B_{\max}$. These are the damping and uniformity conditions of the asymptotic envelope result, not a new arbitrary-schedule guarantee. Thus $\delta = O(T^{-1/2})$ suffices for Theorem 3's rate when its remaining premises hold. A fixed linear-solver iteration count, or a residual in a different norm, requires its own guarantee.

C.9 Warm starts and the inner iteration budget

Theorem 3 assumes that each inner solve starts sufficiently close to its optimum. Warm starts make the initialization depend on the previous decision and on outer-parameter movement. The following recurrence provides an explicit way to check their accuracy.

Lemma 5 (Warm-start tracking)**.** *Under (A1)'s curvature and mixed-derivative bounds, suppose that each inner iteration at fixed θ_t contracts distance to $w^\star(\theta_t)$ by a uniform factor $\gamma \in (0,1)$. Take $K \ge 1$ steps in every round, including the first, and start each round $t \ge 2$ from w_{t-1}. Set $r = \gamma^K$ and $e_t^{\mathrm{in}} = \|w_t - w^\star(\theta_t)\|$. Then*

$$e_t^{\mathrm{in}} \le r\{e_{t-1}^{\mathrm{in}} + \kappa_w\|\Delta_{t-1}\|\}. \tag{22}$$

If the first initialization is within R_{init} of its optimum and $\|\Delta_t\| \le V$, then

$$e_t^{\text{in}} \le r^{t-1} e_1^{\text{in}} + \frac{r\kappa_w V(1-r^{t-1})}{1-r} \le \max\left\{ rR_{\text{init}}, \frac{r\kappa_w V}{1-r} \right\}, \qquad e_1^{\text{in}} \le rR_{\text{init}}. \tag{23}$$

The source-accuracy condition in (A1) need not be assumed to derive this bound; it follows if the right-hand side is at most the chosen ϵ.

Proof. Contraction bounds e_t^{in} by r times the distance from w_{t-1} to $w^\star(\theta_t)$. Insert $w^\star(\theta_{t-1})$ and apply the inner-solution sensitivity bound to obtain (22). Iterating it gives $r^{t-1}e_1^{\text{in}} + r\kappa_w V \sum_{j=0}^{t-2} r^j$. Summing the geometric series proves the first inequality in (23); the second is a convex-combination bound. The first round follows by contraction from its own initialization. □

A sufficient contraction condition is $\mu_w I \preceq H(w,\theta) \preceq L_w I$ on each segment from an inner iterate to its optimum, with all required iterates in that domain. For a gradient step of size $1/L_w$, integrate the Hessian along the segment. The error is multiplied by $I - \overline{H}/L_w$, whose norm is at most $1 - \mu_w/L_w$. If $\mu_w = L_w$, one such step is exact. This condition number L_w/μ_w is distinct from $\kappa_w = C_J/\mu_w$.

The recurrence does not force a positive limiting error at fixed K. More generally, iterating (22) gives $r^{t-1}e_1^{\text{in}} + r\kappa_w \sum_{j=1}^{t-1} r^{t-1-j}\|\Delta_j\|$. If $\|\Delta_j\| \to 0$, split this sum into a fixed finite prefix and a tail of arbitrarily small steps. The prefix vanishes geometrically and the tail is bounded by the small step bound times $r\kappa_w/(1-r)$. Hence $e_t^{\text{in}} \to 0$. A uniform finite-horizon accuracy guarantee still requires the initialization and movement bounds above.

C.10 Finite-horizon consistency of the DDE discretization

The continuous and discrete propositions address stability. A different question is whether a discrete trajectory approximates the corresponding continuous one over a fixed time interval. The local-error and geometric-summation argument below makes that connection explicit, following the standard distinction between consistency and stability in numerical differential equations (Iserles, 2009).

Lemma 6 (Euler error on a fixed time interval)**.** *Fix $\lambda > 0$, $B \ge 0$, and $U > 0$, and set $a = \lambda(1+B)$, $b = \lambda B$. For each step $\eta > 0$, let x_η solve* (11) *with lag $h = \eta$. Suppose x'_η is M_E-Lipschitz on $[0, U]$, uniformly over the step sizes compared. Initialize* (10) *with matching values at $-\eta$ and 0. At every grid point $n\eta \le U$,*

$$|x_n - x_\eta(n\eta)| \le \frac{M_E \eta}{2(a+b)} \left(e^{(a+b)U} - 1 \right). \tag{24}$$

Proof. The integral of x'_η over one step differs from its left-endpoint approximation by at most $M_E\eta^2/2$. Let E_n be the largest grid error through index n, including the matching history values at $-1, 0$. Subtracting the exact and numerical recurrences gives

$$E_{n+1} \le [1 + (a+b)\eta]E_n + M_E\eta^2/2, \qquad E_0 = 0,$$

using $|1 - a\eta| \le 1 + a\eta$ and $b \ge 0$. Summing the geometric series yields $E_n \le M_E\eta\{[1 + (a + b)\eta]^n - 1\}/[2(a+b)]$. Finally, $(1+v)^n \le e^{nv}$ for $v \ge 0$ proves (24). □

This is first-order accuracy at fixed U under the uniform derivative premise. Arbitrary continuous histories need not provide that regularity. The comparison uses the member $h = \eta$ of a DDE family, not a fixed physical feedback lag as $\eta \to 0$. It does not control arbitrarily long trajectories. Even $x' = -\lambda x$ has unstable Euler steps when $\eta\lambda > 2$, which explains the need for the separate discrete root test.

C.11 Which assumptions each result uses

The results answer different questions. Table C.1 separates their premises so that the stronger regret assumptions are not mistaken for requirements of every error calculation.

Result	Premises and conclusion
Theorem 2	(A1), (A2), (A4), constant-delay arrival and cache clock: estimator error. No outer convexity or quadratic growth is needed.
Theorem 1	All four assumptions, the constant-delay initialization/capacity conditions, Euclidean updates, and uniform constants: envelope regret.
Theorem 3	Theorem 1 plus Taylor regularity: executed regret. The iteration-budget conclusion additionally needs contraction and initialization bounds.
Lemma 3	(A1), (A2), (A4) and bounded current envelope gradients: variable-arrival error. Adding (A3) yields (21), not a general-delay rate.
Lemma 4	Inner curvature, sensitivity and stated primitive bounds; exact arrival adjoints replaced by a residual condition.
Lemma 5	Inner-solution sensitivity, uniform contraction, initialization and outer movement control: warm-start accuracy.
Propositions 1–2	Common positive-definite curvature, fixed old-buffer count and constant applied Euclidean rate: continuous stability and exact discrete interval.
Lemma 6	The specified scalar DDE family, matching history and uniformly Lipschitz derivatives: finite-time Euler approximation.

Table C.1: Dependencies of the formal results. Stationarity and the stated comparison domains apply to the hypergradient/regret results.

C.12 Scope for the experimental decision layers

The mathematical results are conditional, while each benchmark tests a specified algorithm on its own decision layer. These roles are complementary. The LQR proxy has a quadratic inner problem with a positive-definite Hessian. That fact does not establish outer convexity, bounded parameter trajectories, or common frozen curvature throughout its nonlinear parameterization.

Entropy regularization does not remove optimal transport's marginal constraints. Differentiation of its exact optimum requires its constrained optimality system, or justified reduced coordinates; the unconstrained inverse-Hessian formula cannot simply be inserted. Our Sinkhorn experiments instead differentiate a finite iteration map, either once for a frozen direction or repeatedly for the full oracle. Neither finite iteration count nor inner entropy alone verifies a uniform error tolerance, contraction constant, or convexity of the neural outer objective. The exact Dijkstra problem in Warcraft is likewise not a smooth, strongly convex unconstrained inner problem. Its blackbox gradient is an explicitly identified surrogate.

Setting	Differentiated object	What is established here
Formal smooth model	Arrival implicit adjoint at cached inner state	Conditional stationary regret, error and solver-accuracy results.
Quadratic stability model	Common-curvature frozen surrogate	DDE stability, discrete rate interval and finite-time consistency.
LQR proxy	Analytic frozen hypergradient with exact arrival adjoint	Measured stable learning-rate boundaries and beta calibration.
Frozen Sinkhorn	Source-cost derivative of the truncated solver times current predictor Jacobian	Matched decision-loss and increment measurements; no transport inner solves.
Full Sinkhorn	Current predictor and complete K-step solver	Fresh paired loss improvements and measured additional work.
Warcraft	Arrival-time blackbox path direction times current predictor Jacobian	Matched decision-quality effects on a nonsmooth, possibly drifting task.

Table C.2: Theory and benchmark roles. The empirical rows do not assert that all premises of the formal stationary theorem hold.

D EXPERIMENTAL PROTOCOLS AND IMPLEMENTATION IDENTITIES

D.1 COMMON CLOCK AND FACTORIAL DESIGN

Each main-text matched run registers its source decision before collecting same-round arrivals, scores the decision before updating parameters, and invokes the update on every clock. Constant-delay transport-on cells use $B_{\max} = d$; transport-off cells retain no transport buffer. The matched estimators are neither averaged nor clipped. SGD is the Euclidean update. Adam uses moment coefficients $(0.9, 0.999)$, numerical epsilon 10^{-8}, no weight decay, and the same queue-dependent rate schedule. An exactly zero estimator skips the Adam step, so its moment counter does not advance during empty burn-in. Nonzero transport-only updates are allowed. All eight arms in an original optimizer-factorial bundle use the same accelerator type; model-initialization and environment-trace hashes verify pairing.

The common nominal rate 0.001 isolates factors rather than comparing separately optimized hyperparameters. The original optimizer-factorial settings were not chosen independently of earlier development results and are retrospective. The separate oracle factorial and fresh confirmation have their own protocols and seedwise-percentage statistics (Appendix F). For the original optimizer factorial's ten paired seeds, let x_i, y_i be the control and treatment metrics and $z_i = y_i - x_i$. Percentage gains use $100(\bar{x} - \bar{y})/\bar{x}$, not a mean of seedwise percentages. Unadjusted 95% intervals use $\bar{z} \pm t_{9,0.975} s_z/\sqrt{10}$. Holm correction is applied within the four positive-delay transport contrasts of each task, with Adam and $\beta = 1$ fixed. It does not correct every exploratory factorial comparison. Seeds, not correlated rounds, are the independent units. Identical zero-delay cells have undefined zero-variance tests, not artificial p values.

D.2 THE ONE-STEP BILEVEL LQR PROXY

For seeds $0, \ldots, 9$, draw $A_{\text{raw}} \in \mathbb{R}^{10\times 10}$ with independent $N(0, 0.3^2)$ entries and set $A = A_{\text{raw}}/(\|A_{\text{raw}}\|_2 + 0.5)$. Draw $B \in \mathbb{R}^{10\times 3}$ with independent $N(0, 0.5^2)$ entries. Initialize the 130 parameters $\theta = (\widehat{A}, \widehat{B})$ with independent $N(0, 0.1^2)$ entries. The inner gain $K \in \mathbb{R}^{3\times 10}$ minimizes

$$f(K;\theta) = \operatorname{tr}[(\widehat{A} - \widehat{B}K)^\top Q(\widehat{A} - \widehat{B}K) + K^\top RK], \quad Q = I_{10},\ R = 0.1 I_3.$$

Each decision uses ten gradient steps of size 0.01, warm-started from the previous gain (zero initially), and is scored by

$$\ell(K) = \operatorname{tr}[(A - BK)^\top Q(A - BK) + K^\top RK] + 0.1^2 \operatorname{tr}(Q).$$

The last term is expected one-step isotropic process-noise cost. There is no infinite-horizon rollout or Riccati equation. IGT stores the source gain, solves the exact arrival adjoint with matrix $2(\widehat{B}^\top Q\widehat{B} + R)$, and evaluates analytic frozen gradients without later inner or adjoint solves.

For each $d \in \{1, 10, 20, 40, 60, 80, 100, 150, 200, 300, 500\}$, every candidate base scale is tested on ten seeds. The horizon is $N_d = \max(1000, 4d)$ and warm-up $W_d = \max(200, d)$. The tail contains the final $n_d = \max(100, \lceil 0.25(N_d - W_d)\rceil)$ losses. The criterion requires finite losses throughout; maximum absolute *tail* loss at most $\max(1000, 100|\ell_1|)$; least-squares tail slope divided by $\max(|\bar{\ell}_{\text{tail}}|, 10^{-12})$ at most 10^{-4}; and tail population standard deviation divided by the same denominator at most 0.5. A falling tail is allowed. Large finite transients before the tail are not excluded by its ceiling. A candidate passes when at least six seeds satisfy every condition, with enough post-warm-up samples. This is an operational finite-horizon criterion, not a certificate of asymptotic stability.

Search starts from $[10^{-5}, 0.1]$, expands by factors of two as needed, and then uses geometric bisection until the unstable upper endpoint divided by the stable lower endpoint is at most 1.05, with at most 20 bisections. Safety bounds are $[10^{-9}, 10^6]$; reaching one without a bracket records censoring, not a boundary. Every reported primary boundary is bracketed. Figure points are stable lower endpoints, not means of ten separately estimated boundaries. For brackets $[l_d, u_d]$ and $[l_1, u_1]$, the normalized point is l_d/l_1 and its shaded search-resolution interval is $[l_d/u_1, u_d/l_1]$. This removes constant rescalings of a method's base parameter, not differences in update schedule. Matrix parameterization and loss scale are common to all methods. The search supplies an accepted/rejected operational

bracket; it does not establish global monotonicity in the searched scale or rule out disconnected stable regions. The comparison uses each method's stated rate rule, not a new cross-method experiment with identical damping schedules.

There is a restricted analytical calibration for constant delay. With source registration before arrival collection, the queue envelope has reached d before the first nonzero update. If a method keeps its parameters fixed during empty-feedback clocks and damping changes only a scalar update coefficient, replacing its base coefficient a by $a\sqrt{1+\beta d}$ exactly cancels the factor $1/\sqrt{1+\beta d}$ on every effective update. This explains the within-IGT calibration; it also applies to delayed OGD under that specified schedule extension. For D-FTRL it applies only if damping multiplies the α/t coefficient in (25), leaving its decision mean and all other state rules unchanged, not if it scales the entire displacement from the current iterate. These are algebraic parameterizations, not newly measured baseline curves or evidence of a transport-only advantage.

D.3 Published baselines and benchmark implementations

Let $\widetilde{g}_s$ be the implicit gradient evaluated at source parameters and the approximate source gain, held fixed until delivery. Delayed OGD uses

$$\theta_{t+1} = \theta_t - \eta_{\mathrm{OGD}} \sum_{s \in A_t} \widetilde{g}_s.$$

This is the unconstrained specialization of the projected delayed OGD update in Quanrud & Khashabi (2015). Their paper does not prescribe a queue-drift Adam penalty or name an algorithm "Robust OMD." Its bounded-domain regret theorem is not validated by scanning rates on this unconstrained bilevel proxy.

Algorithm 1 of Qiu et al. (2025), checked against local arXiv:2506.07595v1, specializes to

$$\begin{aligned}\theta_{t+1} &= \arg\min_{\theta} \left\{ \sum_{s: s+d_s \le t} \langle \widetilde{g}_s, \theta \rangle + \frac{\lambda}{2} \sum_{j=1}^{t} \|\theta - \theta_j\|^2 \right\} \\ &= \frac{1}{t} \sum_{j=1}^{t} \theta_j - \frac{\alpha}{t} \sum_{s: s+d_s \le t} \widetilde{g}_s, \qquad \alpha = 1/\lambda. \end{aligned} \tag{25}$$

We scan α, whose gradient coefficient at clock t is α/t, around the mean of all past decisions. This is a valid learning-rate parameterization, but not the same update as fixed-rate OGD or queue-damped SGD. The implementation updates the decision mean on every clock and accumulates all observed source gradients. An online mean preserves the empty-feedback fixed point without float32 accumulation drift. These bilevel losses are not established to have Qiu et al.'s required strong convexity.

The broader Warcraft benchmark also contains methods with different semantics. "Adam + L2-SP" uses a fixed-start anchor penalty; "queue-drift Adam" uses a queue-weighted drift penalty; "Adam arrival replay" recomputes arrived losses at current parameters. These use the repository's arrival-triggered, averaged/clipped Adam workflow. They are not published D-FTRL or delayed OGD, nor matched transport-only controls. "SPO-style heuristic" denotes the repository's shifted surrogate, not an asserted faithful implementation of the full published SPO+ algorithm. Only the matched factorial isolates optimizer, transport, and damping.

D.4 Modern optimistic delayed baseline (DOOMD)

Algorithm. From Flaspohler et al. (2021) we implement delayed optimistic OMD with Euclidean $\psi = \frac{1}{2}\|\cdot\|^2$ on unconstrained parameters,

$$\theta_{t+1} = \theta_t - \eta \left(\widetilde{g}_{t-d} + h_{t+1} - h_t \right), \qquad h_1 = 0,$$

with their recent-gradient hint $h_{t+1} = (d{+}1)\, \widetilde{g}_{t-d}$ and the hint-free form $h \equiv 0$, which is delayed OGD. Here $\widetilde{g}_s$ is the fully stale hypergradient: the K-step Sinkhorn decision loss differentiated entirely at the cached source parameters θ_s. The Euclidean step size is $\eta = 1/\lambda$. Our neural surrogate is nonconvex, so we evaluate this update empirically without invoking the published regret theorem.

Table D.1: Delayed baselines versus full re-evaluation Adam+IGT on ten paired, previously inspected seeds. Each baseline cell has the lowest mean loss among the tested configurations at that delay. Reduction is $100\,\mathrm{mean}_s[(L_{D,s} - L_{I,s})/L_{D,s}]$. Here $L_{D,s}$ and $L_{I,s}$ are the baseline and full-transport losses for seed s. These selected-cell summaries are descriptive; no post-selection confidence interval is claimed.

d	selected baseline	baseline loss	full/on loss	IGT reduction
Original four-rate grid; selected point summaries only				
5	recent-g, $\eta = 0.003$	904.3	546.6	39.5%
10	recent-g, $\eta = 0.003$	905.1	539.0	40.4%
50	hint-free, $\eta = 0.003$	916.3	558.5	39.0%
Extended six-rate grid; selected point summaries only				
5	hint-free, $\eta = 0.03$	668.5	546.6	18.2%
10	hint-free, $\eta = 0.03$	680.7	539.0	20.8%
50	hint-free, $\eta = 0.03$	761.6	558.5	26.6%

Design. The prespecified grid crosses the two hints, four rates $\eta \in \{10^{-4}, 3{\times}10^{-4}, 10^{-3}, 3{\times}10^{-3}\}$, and three delays $d = 5, 10, 50$. Each cell uses ten paired seeds, 10–19, reusing the inputs, initializations and full-oracle controls from the earlier fresh confirmation. Thus this baseline study uses previously inspected realizations. All 240 arms match their controls' host, GPU model and software; physical GPU UUID equality is recorded for $57/240$ arms. Device matching and computational accounting are detailed in Appendix D.10.

Results. All 24 fixed contrasts against full re-evaluation Adam+IGT (undamped, base rate 0.001) pass Holm at $p \leq 1.1 \times 10^{-12}$; every one of the 240 paired comparisons favors IGT, with a smallest single-seed reduction of 37.5%. The mean within-seed reduction is 39.0–45.5% across this grid. Table D.2 reports all 24 fixed contrasts, while Table D.1 summarizes the lowest-mean-loss cells descriptively. Loss improves at every grid transition in all 60 seed×hint×delay series: the largest rate is the best tested rate, which motivates the extension below.

Interpretation. The baseline's mean loss changes by only $+1.2$–1.9% from $d{=}5$ to 50 at $\eta{=}3{\times}10^{-3}$. This observation does not identify the cause of the gap: comparing DOOMD with arrival-only Adam changes both the optimizer and the gradient-evaluation point.

Two trajectory diagnostics suggest hypotheses for the optimization gap. Successive hypergradients are nearly orthogonal (mean adjacent cosine ≈ 0.014–0.016; measured recent-g/hint-free step-length ratios of 9.35, 16.49, and 72.46 at $d = 5, 10, 50$ match the orthogonal prediction $\sqrt{(d{+}2)^2 + (d{+}1)^2}$ within 1.4%). At $d = 50$ and $\eta = 0.003$, the hint increases loss on all ten seeds, by 0.54% on average relative to the hint-free loss within each seed. Thus it substantially changes step lengths even where its loss effect is small. Enabling transport shortens average Adam parameter steps by 23–31% while improving loss. These observations motivate study of hint quality and the interaction between transport and adaptive updates; neither identifies the cause of the performance gap.

Other comparator families. DOOMD supplies a modern optimistic delayed baseline. Online bilevel alternating gradients (OAGD; Ataee Tarzanagh et al., 2024, Algorithm 1) instead assume that both round-t objectives are revealed at round t, carry an inner iterate between rounds, and combine windowed outer hypergradients at the current outer and inner iterates. They do not solve a separate inner problem for each buffered example. Decision-focused OGD (Capitaine et al., 2026) requires a compact domain, a per-round approximate-minimization oracle, and a randomized evaluation point, with no delay model. Their adaptation to this delayed benchmark requires additional design and validation; both remain unperformed comparisons.

Prospective extension: upper grid and fully stale Adam. The extension protocol was committed and release-hashed before its smoke test and production outcomes. It added 150 arms on the same paired realizations: 120 upper-rate arms and 30 fully stale Adam arms, all validated with no numerical failures. The protocol timing does not make these reused realizations a new independent confirmation.

Rate response. Extending the grid tenfold to $\eta \in \{10^{-2}, 3{\times}10^{-2}\}$ improves the group-mean loss of all six hint×delay series at both new rates. In five series every seed improves; at $d = 50$ with recent-g, five seeds worsen from 0.01 to 0.03, while the mean improves from 890.69 to 889.85.

The best extended cells are all hint-free at $\eta = 0.03$; full re-evaluation Adam+IGT reduces loss by 18.2–26.6% relative to these cells (Table D.1). The twelve new fixed cells are reported descriptively, without inferential tests, in Table D.3. Both grids place the best tested rate at their upper edge; neither brackets an optimum or establishes performance against a globally tuned baseline.

Gradient freshness at fixed Adam. The remaining thirty arms use fully stale gradients with undamped Adam at rate 0.001, matching the arrival-only control's optimizer, clock, inputs and initialization. The stale-gradient arm's mean within-seed excess over arrival re-evaluation is 2.6%, 5.3%, and 19.9% at $d = 5, 10, 50$, using the arrival control as denominator. This isolates the choice of source versus current evaluation at fixed Adam; the resulting trajectories may differ. The descriptive staleness cost increases across these three tested delays. Stale Adam also lowers loss by 18–34% relative to the best *original four-rate* DOOMD cell at each delay, using mean within-seed reductions with that selected baseline as denominator. This latter comparison changes the optimizer and rate, and at $d = 5, 10$ also the hint; it is a descriptive performance comparison.

The three prespecified stale-Adam contrasts instead compare with *full transport*, and hence combine freshness and transport. Writing $L_{S,s}$ for stale Adam's loss and $L_{I,s}$ for full transport's loss, we report $r_s = 100(L_{I,s} - L_{S,s})/L_{I,s}$. The means are -9.5%, -15.2%, -34.5%, and their Holm-adjusted p-values are 1.43×10^{-8}, 1.45×10^{-9}, and 3.88×10^{-11} (Table D.4).

Conditional optimizer comparison. The fixed-rate path from hint-free delayed SGD to stale Adam, then arrival re-evaluation and full transport, gives three conditional contrasts (Table D.5). The first comparison fixes $\eta = 0.001$ and the stale oracle in both arms; it is not a comparison to the selected best SGD rate. These contrasts describe optimizer choice, gradient freshness and transport under specific schedules. Their percentages use different reference losses and are not an additive causal decomposition.

D.5 Sinkhorn data and frozen direction

For each seed $0, \ldots, 9$, sample $x_t \sim N(0, I_{20})$. True costs are the elementwise softplus of $(A_0 + Z_t)x_t + b_0$, reshaped to 10×10. Normal initial entries have standard deviations 0.3 for A_0 and 0.1 for b_0. The drift is $Z_{t+1} = \mathrm{clip}(0.98Z_t + 0.05\xi_t, -1, 1)$ with independent standard-normal noise. Cost logits are clipped to $[-20, 20]$ and costs have floor 0.01. The predictor is a 20–128–128–100 multilayer perceptron with positive outputs and 32,100 trainable parameters. Marginals are uniform. The horizon is 1000 rounds.

The executed plan $P_{10}(\widehat{C}_t)$ uses ten Sinkhorn iterations with entropy coefficient 0.05. Its score is $\langle C_t, P_{10}(\widehat{C}_t)\rangle$, without entropy in the reported loss. The reference is $P_{100}(C_t)$, a 100-iteration true-cost plan with the same entropy coefficient. This is a contemporaneous regularized reference, not a fixed predictor comparator or the unregularized optimum. Its gap is therefore not static regret or a certified nonnegative unregularized optimality gap.

The source round saves x_s and $\widehat{C}_s$. At arrival, the adapter differentiates $\langle C_s, P_{10}(\widehat{C}_s)\rangle$ at that saved prediction, producing a frozen direction r_s. It subsequently evaluates $g_s(\theta) = D_\theta \widehat{C}_\theta(x_s)^\top r_s$. Cache creation uses one truncated Sinkhorn solve; transport uses predictor differentiation only. This is a source-prediction truncated-solver derivative, not the exact arrival adjoint of (5). Entropic smoothness also does not make the neural outer objective convex.

D.6 Warcraft data and frozen direction

We use the official Warcraft shortest-path oneskin data associated with Vlastelica et al. (2020), DOI 10.17617/3.YJCQ5S. Instances contain 96×96 RGB images and 12×12 eight-connected, vertex-weighted graphs from upper-left to lower-right. The data have 10,000 training, 1000 validation, and 1000 test examples. The online score uses a seed-specific permutation of training examples, with each of 5000 decisions evaluated before learning from its feedback. This is an online prequential score, not a held-out test-set statistic. Channels are standardized from training-split statistics only.

The shared tile predictor is a 192–128–1 multilayer perceptron on each 8×8 RGB tile, with softplus outputs and 24,833 trainable parameters, not a CNN or a frozen-feature linear head. Each graph position's log-cost drift obeys $z_{t+1} = \mathrm{clip}(0.98z_t + 0.05\xi_t, -1.2, 1.2)$; true cost is base cost times

Table D.2: All fixed cells of the original DOOMD grid ($n = 10$ paired seeds). Each row reports the mean $\bar{r}$ of $r_s = 100(L_{I,s} - L_{D,s})/L_{I,s}$, its pointwise 95% t interval, and the Holm-adjusted test of zero mean. Negative values favor full re-evaluation Adam+IGT. Holm adjustment covers these 24 contrasts only; the intervals are not simultaneous or selection-adjusted.

d	hint	η	n	baseline loss	full/on loss	$\bar{r}$ [95% CI]	Holm p
5	recent-g	0.0001	10	988.60	546.61	-80.89 [-83.53, -78.25]	1.10×10^{-12}
5	recent-g	0.0003	10	981.72	546.61	-79.63 [-82.23, -77.03]	1.10×10^{-12}
5	recent-g	0.001	10	958.89	546.61	-75.45 [-77.92, -72.99]	1.10×10^{-12}
5	recent-g	0.003	10	904.30	546.61	-65.46 [-67.66, -63.27]	1.10×10^{-12}
5	hint-free	0.0001	10	988.66	546.61	-80.90 [-83.54, -78.26]	1.10×10^{-12}
5	hint-free	0.0003	10	981.91	546.61	-79.67 [-82.27, -77.06]	1.10×10^{-12}
5	hint-free	0.001	10	959.48	546.61	-75.56 [-78.04, -73.08]	1.10×10^{-12}
5	hint-free	0.003	10	905.67	546.61	-65.71 [-67.94, -63.49]	1.10×10^{-12}
10	recent-g	0.0001	10	988.60	539.04	-83.41 [-85.40, -81.42]	1.55×10^{-13}
10	recent-g	0.0003	10	981.72	539.04	-82.13 [-84.08, -80.18]	1.55×10^{-13}
10	recent-g	0.001	10	958.91	539.04	-77.90 [-79.70, -76.10]	1.42×10^{-13}
10	recent-g	0.003	10	905.12	539.04	-67.92 [-69.49, -66.35]	1.42×10^{-13}
10	hint-free	0.0001	10	988.71	539.04	-83.43 [-85.42, -81.44]	1.55×10^{-13}
10	hint-free	0.0003	10	982.06	539.04	-82.20 [-84.15, -80.25]	1.55×10^{-13}
10	hint-free	0.001	10	959.97	539.04	-78.10 [-79.90, -76.29]	1.42×10^{-13}
10	hint-free	0.003	10	907.07	539.04	-68.28 [-69.85, -66.71]	1.42×10^{-13}
50	recent-g	0.0001	10	988.63	558.52	-77.02 [-79.06, -74.99]	2.80×10^{-13}
50	recent-g	0.0003	10	981.89	558.52	-75.81 [-77.80, -73.82]	2.80×10^{-13}
50	recent-g	0.001	10	960.58	558.52	-72.00 [-73.87, -70.13]	2.80×10^{-13}
50	recent-g	0.003	10	921.27	558.52	-64.96 [-66.65, -63.26]	2.80×10^{-13}
50	hint-free	0.0001	10	989.06	558.52	-77.10 [-79.14, -75.05]	2.80×10^{-13}
50	hint-free	0.0003	10	983.09	558.52	-76.03 [-78.04, -74.01]	2.80×10^{-13}
50	hint-free	0.001	10	963.22	558.52	-72.47 [-74.38, -70.56]	2.80×10^{-13}
50	hint-free	0.003	10	916.32	558.52	-64.07 [-65.76, -62.38]	2.80×10^{-13}

Table D.3: All twelve upper-grid cells ($n = 10$). Mean losses and paired IGT reductions with the baseline as denominator are descriptive: the extension did not prescribe inferential tests for this rate-response family.

d	hint	η	baseline loss (mean $\pm$ SD)	full/on loss	IGT reduction
5	recent-g	0.01	790.52 $\pm$ 11.49	546.61	30.85%
5	recent-g	0.03	701.71 $\pm$ 10.63	546.61	22.09%
5	hint-free	0.01	785.95 $\pm$ 12.77	546.61	30.44%
5	hint-free	0.03	668.52 $\pm$ 14.10	546.61	18.21%
10	recent-g	0.01	804.32 $\pm$ 11.83	539.04	32.97%
10	recent-g	0.03	743.72 $\pm$ 12.17	539.04	27.51%
10	hint-free	0.01	790.22 $\pm$ 12.94	539.04	31.77%
10	hint-free	0.03	680.69 $\pm$ 16.20	539.04	20.78%
50	recent-g	0.01	890.69 $\pm$ 11.13	558.52	37.29%
50	recent-g	0.03	889.85 $\pm$ 11.93	558.52	37.22%
50	hint-free	0.01	819.53 $\pm$ 13.15	558.52	31.84%
50	hint-free	0.03	761.65 $\pm$ 14.09	558.52	26.65%

Table D.4: The extension's three prespecified stale-Adam versus full-transport contrasts ($n = 10$). Each row reports the mean $\bar{r}$ of $r_s = 100(L_{I,s} - L_{S,s})/L_{I,s}$; negative favors IGT. The pointwise 95% t interval and three-test Holm correction use this same estimand. These are reused-seed comparisons, not independent confirmation.

d	stale Adam loss	full/on loss	$\bar{r}$ [95% CI]	Holm p
5	598.83	546.61	-9.55 [-10.69, -8.41]	1.43×10^{-8}
10	621.12	539.04	-15.22 [-16.51, -13.93]	1.45×10^{-9}
50	751.22	558.52	-34.50 [-36.37, -32.63]	3.88×10^{-11}

Table D.5: Conditional optimizer, freshness and transport contrasts at base rate 0.001, without damping ($n = 10$). Each gain is the mean within-seed reduction: stale SGD to stale Adam, stale Adam to arrival Adam, then arrival Adam to full transport, respectively, relative to the first arm in each pair. These descriptive percentages have different denominators and must not be added.

d	stale SGD	stale Adam	arrival Adam	optimizer gain	freshness gain	transport gain
5	959.48	598.83	583.53	37.58%	2.54%	6.31%
10	959.97	621.12	589.69	35.29%	5.05%	8.57%
50	963.22	751.22	626.36	22.00%	16.61%	10.81%

e^{z_t}. The target path is recomputed under current true costs. Zero-drift runs retain base costs. Seeds are $42, \ldots, 51$.

On arrival of source example s, let $\widehat{c}$ be the current prediction for its saved image, y_s the delayed target path, and $P(c)$ a Dijkstra path indicator. The blackbox Hamming surrogate uses

$$u_s = 1 - 2y_s, \qquad \widehat{c}' = \max(\widehat{c} + 20u_s, 0), \qquad r_s = [P(\widehat{c}') - P(\widehat{c})]/20.$$

Both paths and r_s are frozen. Later transport evaluates $D_\theta \widehat{c}_\theta(x_s)^\top r_s$ without Dijkstra. Cache creation uses two solves, separate from executed and reference-path scoring. Unlike Sinkhorn, the direction is constructed at the arrival prediction. This distinction is not covered by the stationary implicit-adjoint theorem.

D.7 Artifacts and computational accounting

Raw runs record Git commit, implementation class, loss, drift, grid, paired seeds, command, algorithm contract, and solver counters. A compact manifest hashes every input JSON. Tables and figures are regenerated after strict validation against the saved study summaries. The main-text evidence sets contain 22 LQR ablation brackets, 11 exact D-FTRL brackets, 400 Sinkhorn factorial runs, 800 Warcraft factorial runs, and 480 broader Warcraft runs. An additional 66-boundary LQR dataset is validated separately; its frozen IGT cells duplicate the damped ablation rather than independent replications. The D-FTRL comparison uses the 11 boundaries for (25).

A T-round run with constant delay $d < T$ has $T - d$ arrivals. The frozen Sinkhorn adapter uses one direction construction per arrival; Warcraft uses two Dijkstra solves per arrival, in addition to decision scoring. Both frozen adapters record zero inner solves during transport, although up to $B_{\max}$ frozen-gradient evaluations per clock still cost computation. Bundles use one GPU and two CPU cores, running eight factorial arms sequentially on that GPU. Accelerator identity is saved; heterogeneous-GPU wall times are not pooled into a speedup claim. The oracle factorial adds 400 same-hardware arms and the fresh confirmation adds 60 arms; their derivative identities and measured training costs appear in Appendix F. Appendix G separately reports 500 validated event-triggered replay runs; their post-update diagnostics do not establish the matched algorithm's executed-decision claims or its theorem.

D.8 Configuration reference

Table D.6 collects the settings used in the main comparisons. The detailed protocols above specify the objectives, arrival oracles and measurement rules.

Table D.6: Main benchmark configurations. The fresh full-Sinkhorn cohort is specified separately below.

Setting	LQR stability	Sinkhorn optimizer factorial	Warcraft optimizer factorial
Decision problem	One-step quadratic control; 10 states, 3 controls	10×10 OT plan with uniform marginals	12×12 eight-connected shortest path
Predictor	Matrices $(\widehat{A}, \widehat{B})$; 130 parameters	20–128–128–100 MLP; 32,100 parameters	Shared 192–128–1 tile MLP; 24,833 parameters
Inner computation	10 gradient steps, inner rate 0.01, warm-started gain	10 Sinkhorn steps, entropy 0.05	Exact Dijkstra path; blackbox backward perturbation 20
Arrival / memory	Exact arrival adjoint at the saved gain; analytic frozen gradients	Frozen source-prediction cost derivative; current predictor Jacobian	Frozen arrival-prediction path direction; current predictor Jacobian
Environment	Stationary synthetic matrices; quadratic weights $Q = I$, $R = 0.1I$	Cost-coefficient OU drift: volatility 0.05, reversion 0.02	Stationary and log-cost OU drift: volatility 0.05, reversion 0.02
Rounds	$\max(1000, 4d)$ per rate candidate	1000	5000
Seeds	$0, \ldots, 9$ for every candidate scale	$0, \ldots, 9$	$42, \ldots, 51$
Delays	$1, 10, 20, 40, 60, 80, 100, 150, 200, 300, 500$	$0, 1, 5, 10, 50$	$0, 1, 5, 10, 50$
Outer comparison	IGT $\beta = 0/1$, delayed OGD, published D-FTRL	SGD/Adam $\times$ transport off/on $\times$ $\beta = 0/1$	SGD/Adam $\times$ transport off/on $\times$ $\beta = 0/1$
Base rate	Automatic bracket expansion and geometric bisection	0.001, held fixed across cells	0.001, held fixed across cells
Memory	IGT capacity $B_{\max} = d$	On: $B_{\max} = d$; off: zero	On: $B_{\max} = d$; off: zero
Scoring	Pre-update one-step cost used in the stability criterion	Cumulative pre-update plan cost	Last-200-round mean pre-update gap to current true-cost shortest path
Primary interpretation	Largest accepted base update scale; bracket resolution at most 5%	Conditional effects of optimizer, transport and damping	Conditional effects on decision quality, under two drift settings

The oracle factorial keeps the Sinkhorn data, model, base rate and horizon above, fixes Adam, and crosses frozen/full oracle, transport off/on and $\beta = 0/1$. Its five delays and ten seeds give 400 arms. The prospective confirmation fixes the full oracle and $\beta = 0$, compares transport on/off at $d = 5, 10, 50$, and uses new seeds $10, \ldots, 19$: 60 arms forming 30 pairs. It uses the same $K = 10$, entropy 0.05 and base rate 0.001.

The delayed-baseline studies reuse these same thirty seed–delay pairs; they do not constitute another fresh cohort. The original DOOMD grid has 240 arms: two hint choices, four rates, three delays and ten seeds. Its extension adds 120 arms at two larger rates and 30 fully stale Adam arms at rate 0.001. All use $K = 10$, entropy 0.05, 1000 rounds, the same input bytes and initialization, and no queue damping. The Adam controls are the existing full-oracle arrival-only and transport arms. Tables D.2–D.5 give every fixed contrast and distinguish descriptive extensions from the original 24-test and new three-test families.

The five additional replay studies use the shared protocol in Appendix G.1; their different scoring, clipping and cache clock must not be inferred from this table. Likewise, the broader Warcraft benchmark includes additional methods and constant/Poisson delays whose cells are given in the complete-results tables.

D.9 SUMMARY OF EXPERIMENTAL EVIDENCE

Table D.7 separates the questions answered by the experiments and records both positive findings and the limits of each comparison.

Study and location	Question and metric	Main observation and scope
LQR, Section 5.1	How does delay change stable base rates?	Damping changes IGT's nominal calibration. IGT retains more of its base scale than delayed OGD; exact D-FTRL retains the largest fraction of its own reference scale.
Warcraft, Section 5.2	Does frozen transport help at fixed optimizer and schedule?	Damped Adam gains at delays 5, 10 and 50 under drift; effects depend on optimizer and drift. The delay-one contrast is inconclusive.
Frozen Sinkhorn, Section 5.3	What increments are actually summed, and what do they buy?	Direct alignment measurement and positive damped-Adam transport effects; neither proves uniform cancellation or outer convexity.
Oracle factorial, Appendix F	Does re-solving change the effect of transport and damping?	Substantial undamped full-oracle benefits; the different damped primary comparison fails its prespecified eligibility rule.
Fresh confirmation, Table 3	Does undamped full transport improve pre-update loss on fresh realizations?	All three prespecified delays pass, with 6.31–10.81% mean paired reductions and substantial additional training work.
Delayed baselines, Appendix D.4	How competitive is a modern delayed optimizer, and what changes at fixed Adam?	IGT beats the best tested extended-grid cells by 18.2–26.6%; the best rate remains at the edge. Fixed-Adam stale losses exceed arrival re-evaluation by 2.6–19.9%. Reused-seed, schedule-specific comparisons.
Replay studies, Appendix G	Do replay effects persist across anchored Adam, solver budgets and delays?	Positive post-update effects across the reported settings, with queue and solver diagnostics; exploratory and not the matched Algorithm 1 contract.
Replay timing, Appendix F.4	How much does pre/post scoring change replay gains?	Small timing shifts in completed pairs; incomplete cohorts and historical archive reproduction remain separately reported.

Table D.7: What each experiment contributes. Learning-rate stability, decision cost, dynamic-reference gaps and fixed-comparator regret are distinct quantities; the experiments do not substitute one for another.

D.10 SOFTWARE, RESOURCES, AND DATA ATTRIBUTION

The implementation uses PyTorch (Paszke et al., 2019). All 60 fresh-confirmation artifacts record Python 3.12.8, PyTorch 2.4.1+cu121, CUDA 12.1 and float32 computation. They comprise 36 arms on RTX A5500, 16 on V100-SXM2-32GB, and 8 on H200 NVL; each treatment/control pair uses the same device. Each job reserves two CPU cores and one GPU. These resource records apply to this cohort; other cohorts retain their own environment metadata.

The 390 delayed-baseline arms match the comparator's host, GPU model, software and thread configuration. Exact physical-GPU UUID equality holds for 57/240 original and 35/150 extension arms. Same-device timing ratios are not reported for the other arms; host matching is not device identity.

Summing the recorded fresh-run wall timers gives 1.47 hours, including environment/reference computation, decision scoring and updates; the synchronized training-update phases sum to 1.25 hours. These are sums across arms, not elapsed campaign time, scheduler allocation time, or the total resource use of the project. Compilation, imports, preparation, diagnostics, failed attempts and earlier development are not included in those totals. A complete all-campaign compute total has not been reconstructed. The paired timing ratios in Table F.4 answer the narrower comparison of training work on matched devices.

The synthetic LQR and Sinkhorn instances are generated by the experiment code. Warcraft inputs come from the original archive cited in Appendix D.6; the manuscript package does not redistribute its image assets or claim ownership of them. PyTorch's main source license is BSD-style, with its copyright notices and redistribution conditions in the versioned license. Third-party data and software retain their respective terms.

E COMPLETE REPORTED RESULTS

The full-cell tables show means ± sample standard deviations across ten seeds, not confidence intervals or standard errors. Warcraft paired intervals use within-seed absolute differences; the fresh Sinkhorn table uses within-seed relative reductions. Machine-readable summaries additionally include secondary metrics and the remaining factorial contrasts.

Table E.1: All LQR stable lower endpoints, with algorithm-specific meanings. Applied IGT divides its damped nominal rate by $\sqrt{1+d}$.

d	IGT $\beta = 0$	IGT $\beta = 1$	Applied, $\beta = 1$	Delayed OGD	D-FTRL α
1	0.308442	0.436203	0.308442	0.8	2.69087
10	0.0191095	0.0649382	0.0195796	0.141421	3.06433
20	0.00930572	0.0437144	0.00953927	0.0562341	2.57678
40	0.00453158	0.0283874	0.00443336	0.0305053	2.16681
60	0.00283874	0.0220673	0.00282543	0.0177828	1.90273
80	0.00212875	0.0191095	0.00212328	0.0124094	1.74481
100	0.00177828	0.0184342	0.00183427	0.00964662	1.67084
150	0.00119709	0.0148551	0.00120889	0.00604296	1.53217
200	0.000897687	0.0124094	0.00087529	0.00437144	1.53217
300	0.000649382	0.0111397	0.000642084	0.00273842	1.46721
500	0.000421697	0.00964662	0.000430979	0.00159634	1.405

Table E.2: Frozen-only Sinkhorn optimizer factorial: cumulative decision loss, all 40 cells (SGD/Adam, transport, damping, and delay).

Opt.	Memory	β	$d = 0$	$d = 1$	$d = 5$	$d = 10$	$d = 50$
SGD	off	0	955.04 ± 14.58	955.14 ± 14.59	955.52 ± 14.60	955.99 ± 14.63	959.15 ± 14.87
SGD	off	1	955.04 ± 14.58	964.15 ± 15.14	973.86 ± 15.72	977.47 ± 15.94	983.19 ± 16.30
SGD	on	0	955.04 ± 14.58	955.05 ± 14.59	955.42 ± 14.60	955.88 ± 14.64	958.95 ± 14.87
SGD	on	1	955.04 ± 14.58	964.12 ± 15.14	973.86 ± 15.72	977.47 ± 15.95	983.19 ± 16.31
ADAM	off	0	578.88 ± 12.41	583.09 ± 11.42	602.27 ± 10.92	624.67 ± 11.70	835.25 ± 24.14
ADAM	off	1	578.88 ± 12.41	595.87 ± 10.73	647.74 ± 10.98	689.26 ± 11.85	814.75 ± 14.11
ADAM	on	0	578.88 ± 12.41	589.77 ± 13.45	759.47 ± 19.83	915.22 ± 19.15	935.45 ± 18.80
ADAM	on	1	578.88 ± 12.41	593.67 ± 11.89	638.58 ± 10.95	675.46 ± 11.35	804.76 ± 14.95

Table E.3: Frozen-only Sinkhorn factorial with Adam and damping: transport on versus off. Gain is the percentage reduction of group means; the interval is for the paired absolute loss difference.

d	Transport off	Transport on	Gain (%)	Paired Δ [95% CI]
1	595.866	593.667	0.37	-2.199 $[-3.900, -0.498]$
5	647.738	638.578	1.41	-9.159 $[-11.324, -6.995]$
10	689.264	675.463	2.00	-13.801 $[-15.279, -12.323]$
50	814.748	804.759	1.23	-9.989 $[-11.263, -8.716]$

Table E.4: Warcraft tail-200 decision-cost gap, stationary costs.

Opt.	Memory	β	$d = 0$	$d = 1$	$d = 5$	$d = 10$	$d = 50$
SGD	off	0	1.19 ± 0.20	1.19 ± 0.20	1.19 ± 0.20	1.19 ± 0.21	1.20 ± 0.20
SGD	off	1	1.19 ± 0.20	1.65 ± 0.35	3.06 ± 0.51	3.99 ± 0.61	6.89 ± 0.93
SGD	on	0	1.19 ± 0.20	1.18 ± 0.20	1.18 ± 0.20	1.18 ± 0.20	1.16 ± 0.20
SGD	on	1	1.19 ± 0.20	1.65 ± 0.35	3.06 ± 0.51	3.98 ± 0.61	6.87 ± 0.94
ADAM	off	0	0.24 ± 0.06	0.23 ± 0.06	0.23 ± 0.06	0.24 ± 0.07	0.24 ± 0.07
ADAM	off	1	0.24 ± 0.06	0.16 ± 0.05	0.09 ± 0.02	0.07 ± 0.01	0.08 ± 0.03
ADAM	on	0	0.24 ± 0.06	0.20 ± 0.03	0.19 ± 0.05	0.25 ± 0.06	0.18 ± 0.03
ADAM	on	1	0.24 ± 0.06	0.14 ± 0.04	0.12 ± 0.03	0.12 ± 0.04	0.05 ± 0.02

Table E.5: Warcraft tail-200 decision-cost gap, drift volatility 0.05.

Opt.	Memory	β	$d = 0$	$d = 1$	$d = 5$	$d = 10$	$d = 50$
SGD	off	0	3.82 ± 0.53	3.83 ± 0.53	3.83 ± 0.53	3.83 ± 0.54	3.85 ± 0.54
SGD	off	1	3.82 ± 0.53	4.32 ± 0.71	5.60 ± 0.86	6.56 ± 1.03	9.50 ± 1.51
SGD	on	0	3.82 ± 0.53	3.82 ± 0.53	3.82 ± 0.54	3.82 ± 0.53	3.81 ± 0.52
SGD	on	1	3.82 ± 0.53	4.32 ± 0.71	5.59 ± 0.87	6.56 ± 1.03	9.50 ± 1.51
ADAM	off	0	4.83 ± 0.57	4.82 ± 0.59	4.81 ± 0.56	4.79 ± 0.51	4.78 ± 0.53
ADAM	off	1	4.83 ± 0.57	4.24 ± 0.41	3.69 ± 0.34	3.37 ± 0.39	2.96 ± 0.29
ADAM	on	0	4.83 ± 0.57	4.48 ± 0.46	3.70 ± 0.29	15.41 ± 12.01	4.72 ± 2.79
ADAM	on	1	4.83 ± 0.57	4.12 ± 0.40	3.29 ± 0.32	3.03 ± 0.31	2.80 ± 0.27

Table E.6: Transport-on versus off inference, Adam and $\beta = 1$. Holm adjustment is within four positive delays per task; Warcraft drift is 0.05.

Task	d	Gain (%)	Raw p	Holm p
OT	1	0.37	0.0169	0.0169
OT	5	1.41	5.14e-06	1.03e-05
OT	10	2.00	5.6e-09	2.24e-08
OT	50	1.23	2.6e-08	7.8e-08
Warcraft	1	2.77	0.146	0.146
Warcraft	5	10.71	0.00043	0.00172
Warcraft	10	9.98	0.00387	0.00773
Warcraft	50	5.50	0.000816	0.00245

Table E.7: Direct Sinkhorn increment geometry for Adam with transport and $\beta = 1$. Statistics pool defined rounds over ten seeds; zero denominators are excluded. These describe vector increments, not hypergradient error.

d	Mean ρ_t	95th percentile	Maximum
1	1.000	1.000	1.000
5	1.077	1.330	1.882
10	1.073	1.367	2.349
50	1.023	1.108	1.456

Table E.8: Broader Warcraft benchmark, constant delays and drift 0.05: tail-200 decision-cost gap. The Adam-based comparison rows use the benchmark implementations specified in Appendix D.3.

Method	0	10	50	100
IGT-OMD (SGD)	3.82 ± 0.53	6.56 ± 1.03	9.50 ± 1.51	11.19 ± 2.23
Adam + L2-SP	4.81 ± 0.45	4.82 ± 0.58	4.91 ± 0.79	4.87 ± 0.82
Queue-drift Adam	4.95 ± 0.49	4.33 ± 0.40	3.74 ± 0.28	3.54 ± 0.27
Adam arrival replay	4.88 ± 0.50	4.87 ± 0.50	4.84 ± 0.47	4.84 ± 0.48
2-Stage MSE	2.46 ± 0.40	2.47 ± 0.40	2.46 ± 0.35	2.43 ± 0.33
SPO-style heuristic	11.70 ± 6.04	11.71 ± 6.02	11.60 ± 6.01	11.32 ± 6.02

Table E.9: The same broader Warcraft benchmark with Poisson delays; columns give their means. Theorem 1 does not cover variable delays.

Method	10	25	50	100
IGT-OMD (SGD)	7.01 ± 1.06	8.39 ± 1.38	9.75 ± 1.55	11.35 ± 2.25
Adam + L2-SP	4.74 ± 0.75	4.69 ± 0.71	4.44 ± 0.63	4.63 ± 0.68
Queue-drift Adam	4.02 ± 0.46	3.67 ± 0.27	3.48 ± 0.28	3.32 ± 0.27
Adam arrival replay	4.30 ± 0.58	4.29 ± 0.44	4.22 ± 0.38	4.13 ± 0.32
2-Stage MSE	2.43 ± 0.36	2.41 ± 0.32	2.44 ± 0.38	2.45 ± 0.36
SPO-style heuristic	6.49 ± 5.48	7.75 ± 6.09	6.39 ± 5.19	5.08 ± 4.86

F FULL RE-EVALUATION: MATCHED EVIDENCE AND COMPUTATIONAL COST

This appendix distinguishes three Sinkhorn studies. The optimizer factorial crosses SGD/Adam, transport, and damping using only the frozen oracle. The subsequent *oracle factorial* fixes Adam and crosses frozen/full re-evaluation, transport, and damping. Both use seeds $0, \ldots, 9$. The *fresh confirmation* uses independent realizations from seeds $10, \ldots, 19$ and one prespecified transport-on/off comparison. Their results are not pooled. All three use the data, network, loss, and ten-step Sinkhorn layer in Appendix D.5.

F.1 TWO ORACLES UNDER THE SAME FEEDBACK CLOCK

The frozen oracle holds the source cost-space derivative fixed and updates only the predictor Jacobian. The full oracle recomputes the prediction for the saved source features, runs ten Sinkhorn iterations from the same fixed initialization, and differentiates the whole decision loss $\langle C_s, P_{10}(\widehat{C}_\theta(x_s))\rangle$. *Full* means differentiating that complete truncated computation, not solving the optimization problem exactly or eliminating truncation error.

Both oracles use the same pre-update scoring, summed arrivals and buffered increments, and admission-after-transport rule. No gradient clipping or averaging is applied. Full/off re-solves only arrivals; full/on also re-solves buffered items. Adam's parameters, empty-estimator skip, and queue-dependent rate are shared. The source's delayed true costs become available only at arrival. Re-evaluation never reveals future feedback.

With $T = 1000$ and capacity $B_{\max} = d$, both full arms have $T - d$ arrival solves. The full/on arm additionally has

$$N_{\text{buffer}} = \sum_{t=0}^{T-1} \min\{d, \max(0, t - d)\}$$

buffered solves. At $d = 50$, these counts are 950 and 46,225, respectively. Each is a ten-step forward-and-backward Sinkhorn evaluation. Frozen transport has the same number of buffered gradient evaluations but zero buffered Sinkhorn solves. Decision scoring and the 100-step reference are additional work common to the paired arms.

F.2 ORACLE–TRANSPORT–DAMPING FACTORIAL

The 400 arms cover two oracles, transport off/on, $\beta = 0/1$, delays $0, 1, 5, 10, 50$, and ten seeds. Inputs are replayed from immutable, checksummed float32 caches. Initialization and input hashes are checked; all four oracle/transport arms within each (β, d, seed) cell share hardware and software. Zero-delay transport-on/off traces coincide. For full/on and full/off, the initial loss traces also coincide through the rounds preceding the first possible transport effect. These controls check the implementation; they are not mathematical proofs of all future trajectory behavior.

The design was fixed before this factorial, but its seeds had already been inspected in earlier work. Its two planned contrasts are full/on versus frozen/on and full/on versus full/off. Two-sided paired tests use seedwise relative reductions, with Holm adjustment over eight comparisons (two contrasts and four positive delays) separately at each β. The damped family ($\beta = 1$) was primary. Its declared headline rule required the planned effects to pass statistical and 1% mean-benefit criteria at both delays 10 and 50; that rule was not met. Tables F.2–F.3 retain every planned contrast, including the negative damped full-versus-frozen result. The undamped findings remain secondary for this inspected-seed study.

What each comparison measures. Arrival refresh compares full/off to frozen/off. Frozen transport compares frozen/on to frozen/off; full transport compares full/on to full/off. Full versus frozen/on changes the oracle in both arrivals and memory. Its large undamped gains are not estimates of transport alone. At $d = 50, \beta = 0$, the absolute loss difference between frozen/on and full/on is 370.666: arrival refresh contributes 207.642, removal of the frozen-transport penalty contributes 101.396, and full transport's incremental benefit contributes 61.628. These are an accounting identity along a specified path, not three independent causal mechanisms. Percentages with different denominators must not be added.

Table F.1: Oracle factorial: cumulative pre-update decision loss, mean ± sample standard deviation over ten seeds. This is separate from the frozen-only optimizer factorial.

β	Oracle	Transport	$d = 0$	$d = 1$	$d = 5$	$d = 10$	$d = 50$
0	frozen	off	578.83 ± 12.54	583.10 ± 11.41	602.46 ± 10.76	624.71 ± 11.62	834.36 ± 24.54
0	frozen	on	578.83 ± 12.54	589.93 ± 13.65	757.32 ± 33.81	911.68 ± 20.81	935.76 ± 22.22
0	full	off	578.83 ± 12.54	580.26 ± 12.30	584.74 ± 12.36	590.77 ± 11.87	626.72 ± 12.68
0	full	on	578.83 ± 12.54	567.08 ± 12.20	551.28 ± 11.43	546.25 ± 10.93	565.09 ± 8.76
1	frozen	off	578.83 ± 12.54	595.87 ± 10.73	647.74 ± 10.98	689.26 ± 11.85	814.75 ± 14.11
1	frozen	on	578.83 ± 12.54	593.67 ± 11.89	638.57 ± 10.95	675.46 ± 11.35	804.76 ± 14.95
1	full	off	578.83 ± 12.54	594.35 ± 10.43	643.74 ± 11.18	685.05 ± 11.69	811.70 ± 13.36
1	full	on	578.83 ± 12.54	583.99 ± 10.42	630.05 ± 10.24	672.86 ± 10.79	806.70 ± 12.47

Table F.2: Undamped oracle-factorial contrasts ($\beta = 0$), inspected seeds. Positive gains favor the named treatment. Intervals are pointwise paired 95% intervals. Holm p values use the planned eight-test family; dashes denote descriptive contrasts outside it. Time ratios are paired medians for training updates, not end-to-end speedups.

Delay	Contrast	Gain (%)	95% interval	Holm p	Time ratio
1	Arrival refresh	0.49	[0.08, 0.90]	—	0.95×
1	Frozen transport	-1.17	[-1.88, -0.45]	—	1.27×
1	Full transport	2.27	[1.76, 2.78]	3.16×10^{-6}	1.84×
1	Full vs. frozen/on	3.86	[3.10, 4.62]	2.21×10^{-6}	1.34×
5	Arrival refresh	2.94	[2.10, 3.78]	—	0.95×
5	Frozen transport	-25.70	[-29.25, -22.15]	—	1.38×
5	Full transport	5.72	[5.11, 6.33]	1.70×10^{-8}	5.14×
5	Full vs. frozen/on	27.08	[24.64, 29.53]	4.99×10^{-9}	3.55×
10	Arrival refresh	5.43	[4.53, 6.32]	—	0.95×
10	Frozen transport	-46.00	[-49.39, -42.60]	—	1.47×
10	Full transport	7.53	[7.00, 8.07]	7.43×10^{-10}	9.20×
10	Full vs. frozen/on	40.04	[38.47, 41.62]	5.14×10^{-12}	5.92×
50	Arrival refresh	24.85	[23.42, 26.28]	—	0.96×
50	Frozen transport	-12.24	[-15.26, -9.22]	—	2.11×
50	Full transport	9.82	[9.19, 10.46]	3.65×10^{-10}	38.20×
50	Full vs. frozen/on	39.59	[38.51, 40.66]	2.06×10^{-13}	17.42×

At fixed base rate 0.001, enabling damping deliberately changes the applied rate to $0.001/\sqrt{1 + \bar{q}_t}$. The experiment measures the effect of that schedule, not an optimizer-independent property of damping. Frozen transport has negative undamped effects and positive damped effects in these Adam cells. Full transport has larger undamped than damped incremental gains. A high finite-horizon loss is not, by itself, a proof of asymptotic instability. Nor do these comparisons establish the best outcome after independently tuning each schedule.

F.3 Prospective fresh-realization confirmation

The fresh study fixes $\beta = 0$, base rate 0.001, the full oracle, $T = 1000$, and delays $5, 10, 50$. Seeds $10, \ldots, 19$ generate new environmental trajectories and initializations. A metadata audit of 4,960 relevant saved JSON files found no earlier outcomes for these candidate seeds; this is a repository-scope check, not a claim about inaccessible private runs. The source, protocol, and input manifest were frozen before production. Six diagnostic arms used previously inspected seed 0, never the confirmation realizations. No tuning, seed replacement, optional stopping, or pooling with earlier runs was allowed. All 60 production arms and 30 pair receipts passed validation.

Each pair runs on one verified GPU with two CPU cores and identical input and initialization hashes. Pair order alternates by seed and delay. The two arms differ only in buffered transport. For seed i, the prespecified effect is

$$r_i = 100 \, \frac{S^{\mathrm{pre}}_{i,\mathrm{off}} - S^{\mathrm{pre}}_{i,\mathrm{on}}}{S^{\mathrm{pre}}_{i,\mathrm{off}}}.$$

Table F.3: Damped oracle-factorial contrasts ($\beta = 1$), with the same definitions as Table F.2. The planned damped comparison fails its success criteria; this does not establish equivalence.

Delay	Contrast	Gain (%)	95% interval	Holm p	Time ratio
1	Arrival refresh	0.25	[0.11, 0.39]	—	$0.95\times$
1	Frozen transport	0.37	[0.08, 0.66]	—	$1.23\times$
1	Full transport	1.74	[1.43, 2.05]	2.87×10^{-6}	$1.84\times$
1	Full vs. frozen/on	1.62	[1.30, 1.95]	5.94×10^{-6}	$1.36\times$
5	Arrival refresh	0.62	[0.54, 0.69]	—	$0.95\times$
5	Frozen transport	1.41	[1.08, 1.75]	—	$1.38\times$
5	Full transport	2.13	[1.81, 2.44]	7.81×10^{-7}	$5.14\times$
5	Full vs. frozen/on	1.33	[0.89, 1.78]	2.48×10^{-4}	$3.58\times$
10	Arrival refresh	0.61	[0.53, 0.69]	—	$0.95\times$
10	Frozen transport	2.00	[1.79, 2.21]	—	$1.47\times$
10	Full transport	1.78	[1.51, 2.04]	7.87×10^{-7}	$9.11\times$
10	Full vs. frozen/on	0.38	[0.04, 0.72]	6.32×10^{-2}	$5.91\times$
50	Arrival refresh	0.37	[0.27, 0.47]	—	$0.96\times$
50	Frozen transport	1.23	[1.07, 1.39]	—	$2.11\times$
50	Full transport	0.61	[0.45, 0.78]	4.87×10^{-5}	$38.74\times$
50	Full vs. frozen/on	-0.25	[-0.54, 0.04]	8.57×10^{-2}	$18.23\times$

The analysis averages r_i; unlike the optimizer-factorial tables, this is not a ratio of the two group means. A two-sided one-sample t test of r_i is adjusted by Holm across the three delays. A delay meets the benefit criterion when its adjusted $p < 0.05$ and its mean reduction is at least 1%. The latter is a mean-effect reporting threshold, not a requirement that a confidence interval exclude 1%. The 95% t intervals are pointwise, not simultaneous. The protocol specified conclusions for complete all-pass, some-pass, none-pass, and incomplete outcomes before production.

All three delays pass (Table 3). This confirms a benefit of undamped full re-evaluation transport over its matched arrival-only control on fresh realizations of this task. This conclusion applies to the undamped comparison at the fixed rate, separately from the damped comparison's failed criterion. Performance after separate rate tuning and a general monotonic relationship with delay remain open questions.

Table F.4: Fresh cohort: mean cumulative pre-update loss and training work. Inner evaluations count ten-step Sinkhorn solves used in training. Time ratio is the median of within-pair training-update time ratios. The lower panel gives total training-update seconds over 1000 rounds: median [minimum, maximum] across ten runs per arm, not confidence intervals.

Delay	Arrival only	Full transport	Inner evals: off/on	Time: on/off
5	583.527	546.613	995/5,955	$5.09\times$
10	589.692	539.044	990/10,835	$9.23\times$
50	626.362	558.524	950/47,175	$40.28\times$

Delay	Arrival-only time (s)	Full-transport time (s)
5	7.9 [4.9, 11.1]	40.5 [25.2, 54.8]
10	7.7 [4.9, 8.0]	71.1 [44.8, 73.8]
50	7.8 [7.4, 10.6]	310.5 [303.2, 416.3]

Absolute cost and practical use. Training-time measurements synchronize GPU work and exclude environment generation, decision scoring, and dynamic-reference computation. Ratios are computed within same-device pairs before taking the median. They are observational shared-cluster measurements, not isolated hardware benchmarks; absolute summaries pool the GPUs used at each delay. At $d = 50$, six pairs use RTX A5500 GPUs and four use V100 GPUs. Full transport takes 303–416 s (5.1–6.9 minutes) of learning over 1000 rounds, versus 7.4–10.6 s for arrival-only re-evaluation, with a 10.81% mean paired loss reduction. Its average learning time per round is 0.30–0.42 s, including the initial rounds without feedback; this is neither a worst-case update latency nor a decision-serving latency.

Full re-evaluation trades computation for better use of the same feedback. At these measured runtimes, infrequent decisions can leave ample time for learning between them, and costly decision errors may justify the extra compute. Thus the $40.28\times$ multiplier alone does not establish practical infeasibility. For rapid feedback, many concurrent models, or more expensive inner solves, however, this work could exhaust the update budget and create a backlog. Our implementation is sequential; background learning would require managing model-update lag. Application-specific throughput and equal-time or equal-compute superiority remain untested.

F.4 Event-triggered replay: scoring timing and numerical replication

A separate diagnostic adds pre-update scoring to the event-triggered replay implementation of Appendix G.1. On a V100, post-only and dual pre/post scoring give identical round-end parameter and optimizer state hashes, applied Adam gradients, and post-update losses for two delay/seed configurations and both algorithms. This establishes observer-invariance on those four tested cells, not a hash of every intermediate backward value or a per-device proof.

The production checkpoint has 92 of 100 arms and 42 of 50 validated pairs. Eight treatment arms have not yet been computed because their matched hardware is unavailable; no values are imputed and no seeds are substituted. Table F.5 reports only observed pairs. Their hardware-linked incompleteness prevents treating the partial groups as complete ten-seed estimates.

Table F.5: Paired replay timing diagnostic at the fixed checkpoint. Gains are means of seedwise percentages. Only delay 50 has all ten pairs; the other rows are descriptive partial groups.

Delay	Pairs	Pre gain (%)	Post gain (%)	Post minus pre (pp)
1	8/10	0.0000	0.0000	0.0000
5	8/10	4.5688	4.5727	0.0039
10	7/10	6.3252	6.3415	0.0163
20	9/10	7.9171	7.9581	0.0410
50	10/10	9.4874	9.5172	0.0298

At the complete delay 50, pre-update and post-update gains are 9.4874% and 9.5172%. The mean timing contribution is 0.0298 percentage points (pointwise 95% interval $[0.0149, 0.0447]$). In absolute loss units, the pre-update advantage is 59.510 and the timing contribution is 0.108 (95% interval $[0.002, 0.215]$). Scoring timing therefore does not explain the magnitude of the replay gain at this delay. These descriptive results are not pooled with the fresh confirmation, and do not transfer the replay cache, clipping, or event-clock semantics to the matched algorithm.

Numerical replication is a separate outcome: 12 of 92 available arms match the saved post-score archive at unchanged absolute and relative tolerances of 10^{-5}; 80 do not. All 12 V100 runs match, while runs on other tested GPU models do not. This association is consistent with hardware-dependent numerical trajectories, but the archive lacks per-run hardware provenance and no causal kernel attribution was established. Archive matching remains unresolved beyond those reproduced arms; it does not invalidate the independently validated fresh comparison. In this separate checkpoint the maximum cumulative-loss difference from the archive is below 0.94%. This numerical context neither changes the strict replication outcome nor identifies the cause of differences between older supplementary executions that reused the same seed integers.

G Additional Experiments

We complement the frozen-state experiments with five studies: replay with Adam, replay with an anchored optimizer, uniform delays, inner-solver sensitivity, and delay-pattern variation. Their 500 provenance-complete runs use ten paired seeds per cell. These event-triggered replay variants re-solve old inner problems instead of freezing their states. Their clipping, post-update cache admission, and arrival-event clock differ from the matched full variant in Appendix F. Their post-update score, defined below, answers a different question from the main experiments' executed-decision loss.

G.1 Shared replay protocol and what its score means

The task has a 10×10 transport plan, 20-dimensional features, a 20–128–128–100 predictor, entropy coefficient 0.05, 1000 rounds, and Ornstein–Uhlenbeck cost drift with volatility 0.05 and reversion 0.02. Seeds are $0, \dots, 9$. Except in the inner-K sweep, each plan uses ten Sinkhorn iterations. Adam uses rate 0.001, moment coefficients $(0.9, 0.999)$, and global gradient-norm clipping at one. Queue damping is off. The anchored variant additionally uses the penalty $0.01\|\theta - \theta_{\text{init}}\|^2/2$ in its arrival loss.

Replay off re-evaluates arrived losses at current parameters and averages simultaneous arrivals. *Replay on* also re-solves and differentiates each retained old loss, adding changes in its gradient. The old items store raw feedback and a previous gradient, not a frozen inner state and adjoint. An arriving item is cached *after* the optimizer step and has a budget of d_s subsequent arrival events; each such event decrements that budget. Clocks with no arrival do not update or decrement replay budgets. The first replay difference after admission is zero because its cached gradient is post-update. In particular, at constant delay one the sole difference is zero and the item then expires. The observed zero effect there is a property of this implementation, not a universal identity for transport.

Crucially, the runner updates before scoring the current context. If θ_t^+ denotes the parameters after processing round t's feedback, the reported statistic is

$$S_T^{\text{post}} = \sum_{t=1}^{T} \langle C_t, P_K(\widehat{C}_{\theta_t^+}(x_t)) \rangle.$$

It is a cumulative *post-update plan cost*, not the cost of decisions executed before those updates and not comparator regret. At zero-delay arrivals the current example's feedback can already affect its score. This matters especially when comparing delay distributions that include zero. The artifact field `cumulative_decision_loss` stores this statistic; its name does not change the post-update evaluation timing.

All tables show mean $\pm$ sample standard deviation across ten seeds. Gain is $100(\bar{S}_{\text{off}} - \bar{S}_{\text{on}})/\bar{S}_{\text{off}}$. Within each configuration, trace and initialization hashes, arrival counts, and queue traces establish pairing. The experiments are exploratory; paired intervals and unadjusted p values are reported in Table G.6, not pooled with the main factorial's Holm families. Absolute scores from distinct experimental studies are not pooled. Constant-delay rows repeated across supplementary tables come from separate executions of the same seed labels and recorded configurations, not independent seed populations. In particular, the constant-$d = 20$ rows in the uniform-delay and delay-pattern studies share their initialization hashes and queue traces but not their loss traces. We retain each study's own observations, neither pool them nor count them as independent replications. Their retained device provenance does not isolate the cause of these differences.

G.2 Adam+IGT replay

The Adam+IGT replay implementation improves the post-update score by 4.73–9.57% at delays 5–50 (Table G.1). The delay-one traces and final parameters are exactly identical across the two arms. This is a positive empirical signal for full-loss replay. It does not establish that the frozen Algorithm 1 gains this much: that different question is answered by the frozen matched experiment. The independent full re-evaluation confirmation instead establishes pre-update benefits under the explicitly matched full oracle. A separate paired timing diagnostic is reported in Appendix F.4.

Table G.1: Adam replay: cumulative post-update plan cost.

d	Replay off	Replay on	Gain (%)
1	578.02 ± 11.95	578.02 ± 11.95	0.00
5	582.95 ± 11.58	555.38 ± 10.92	4.73
10	589.25 ± 11.44	550.15 ± 9.64	6.63
20	599.91 ± 11.17	552.12 ± 9.97	7.97
50	626.01 ± 10.52	566.12 ± 8.81	9.57

G.3 Replay with an Adam plus L2-SP optimizer

The anchored pair uses Adam plus an L2-SP penalty around the initial parameters, with or without the same replay correction. Its fixed-reference penalty differs from the D-FTRL update of Qiu et al. (2025) in (25). Table G.2 reports gains of 4.89–9.94% at delays 5–50. This shows a positive replay effect with an anchored Adam variant, not optimizer independence or a result about published D-FTRL. Figure 3 shows the separate small-cohort comparison. The regularization term is applied once in the arrival update; the replay increments differentiate the old decision loss only.

Table G.2: Adam plus L2-SP, with and without replay: cumulative post-update plan cost. The optimizer uses the fixed-reference penalty defined above.

d	Replay off	Replay on	Gain (%)
1	594.01 ± 11.54	594.01 ± 11.54	0.00
5	599.17 ± 11.03	569.86 ± 10.22	4.89
10	605.64 ± 10.80	564.55 ± 9.31	6.78
20	617.16 ± 11.14	562.89 ± 7.98	8.79
50	643.31 ± 11.07	579.40 ± 9.21	9.94

G.4 Uniform Delays and Queue Load

Compare constant delay $d_{\max}$ with independently sampled discrete uniform delays on $\{0, \dots, d_{\max}\}$, for $d_{\max} \in \{10, 20, 50\}$. Table G.3 reports both scores and measured queue load. Replay reduces the post-update score by 3.54–4.96% under uniform delay, versus 6.45–9.50% under constant delay. Uniform-to-constant mean queue ratios are approximately 49.8–50.5% in these finite runs.

There is a simple reason for the queue ratio, but not for a proportional performance gain. Once $t \geq d_{\max}$, the outstanding queue obeys

$$|Q_t| = \sum_{j=0}^{d_{\max}-1} \mathbf{1}\{d_{t-j} > j\}, \qquad \mathbb{E}|Q_t| = \sum_{j=0}^{d_{\max}-1} \Pr(d > j) = \mathbb{E}d.$$

The first equality follows by enumerating source ages; the second is the tail-sum identity for bounded nonnegative integer delays with a common distribution. Hence discrete uniform delays have steady expected queue $d_{\max}/2$. Initial transients affect the finite-horizon average. Neither this identity nor Theorem 1 predicts proportional loss gains; the theorem assumes constant delay. Random uniform delays are also not an adversarial-delay validation. Indeed, the replay-on uniform-delay costs in Table G.3 are higher than the corresponding constant-delay costs despite the smaller queues. This is not a contradiction: an expected queue identity, or a smaller upper bound, does not order realized decision costs. This experiment does not isolate a causal explanation for the difference.

G.5 Sensitivity to Inner-Solver Quality

At constant delay 20, vary Sinkhorn iterations $K \in \{1, 3, 5, 10, 20, 50\}$ for both replay arms. Table G.4 includes an independently recorded solver diagnostic: the round-averaged Frobenius distance $\|P_K(\widehat{C}_{\theta_t^+}(x_t)) - P_{100}(\widehat{C}_{\theta_t^+}(x_t))\|_F$. Both plans use the same predicted costs. This measures distance to a 100-iteration reference, not a certified distance to the exact optimum and not the theorem's uniform worst-round ϵ.

Table G.3: Constant and uniform delays: post-update plan cost and measured mean outstanding queue. The queue is shared by the paired arms.

Pattern	$d_{\max}$	Replay off	Replay on	Gain (%)	Mean $\lvert Q_t \rvert$
Constant	10	588.87 ± 11.52	550.91 ± 10.07	6.45	9.955
Constant	20	599.54 ± 11.25	551.02 ± 8.84	8.09	19.810
Constant	50	625.63 ± 10.64	566.21 ± 8.95	9.50	48.775
Uniform	10	595.33 ± 12.27	574.28 ± 12.48	3.54	4.960
Uniform	20	603.69 ± 10.70	573.74 ± 9.56	4.96	9.986
Uniform	50	614.56 ± 13.00	584.74 ± 10.33	4.85	24.611

Replay gains persist across all six budgets (7.71–8.58%). Its mean plan discrepancy falls by about $737\times$ from $K = 1$ to $K = 50$, while its post-update score changes by about 0.83% from $K = 3$ to $K = 50$. This supports empirical robustness to solver budget on this problem, not a general claim that inner accuracy is irrelevant. Varying K also changes backpropagation and the learned trajectory. Figures 4–5 show the separate five-seed study. The contraction assumption in Appendix C.5 is not established by this sweep.

Table G.4: Inner-K sweep at delay 20: cumulative post-update plan cost and mean Frobenius plan discrepancy to the 100-iteration reference.

K	Replay off	Replay on	Gain (%)	Plan err. off	Plan err. on
1	615.43 ± 13.45	562.60 ± 9.72	8.58	0.1	0.121
3	599.64 ± 13.57	548.51 ± 8.52	8.53	0.0448	0.0467
5	599.31 ± 11.78	549.52 ± 9.88	8.31	0.0217	0.0224
10	599.45 ± 11.30	551.15 ± 8.73	8.06	0.00622	0.00569
20	598.41 ± 12.41	552.25 ± 9.28	7.71	0.00161	0.00134
50	600.14 ± 12.44	553.09 ± 9.31	7.84	0.000213	0.000164

G.6 Delay Pattern Variations

Use constant delay 20, discrete uniform delays on $\{0, \ldots, 40\}$, and periodic ten-round blocks alternating delays zero and 40. These have the same expected or scheduled mean delay 20; a finite uniform trace need not have exactly that realized mean. Replay improves post-update scores by 8.03%, 5.09%, and 7.40%, respectively (Table G.5). Similar mean queue sizes do not imply equal losses. In particular, the periodic and uniform schedules change both arrival batching and which rounds are scored after seeing their own feedback. The experiment is descriptive evidence across schedules, not a proof that queue length alone determines stability or performance.

Table G.5: Delay patterns with expected or scheduled mean delay 20: cumulative post-update plan cost and measured mean queue.

Pattern	$d_{\max}$	Replay off	Replay on	Gain (%)	Mean $\lvert Q_t \rvert$
Constant	20	599.43 ± 11.31	551.32 ± 8.53	8.03	19.810
Periodic	40	578.49 ± 10.19	535.66 ± 10.07	7.40	19.510
Uniform	40	612.56 ± 12.33	581.39 ± 10.24	5.09	19.788

Table G.6: Exploratory paired inference for all replay studies. Δ is replay on minus off; intervals and two-sided tests use ten paired seeds and are unadjusted for multiplicity. C, U, P denote constant, uniform, and periodic delay, followed by $d_{\max}$. Dashes mark identical zero-variance contrasts, for which the paired t test is undefined.

Study	Setting	Paired Δ	95% interval	Raw p
Adam	1	0.000	$[0.000, 0.000]$	—
Adam	5	-27.575	$[-30.686, -24.463]$	8.89e-09
Adam	10	-39.096	$[-43.507, -34.686]$	8.87e-09
Adam	20	-47.794	$[-53.087, -42.502]$	7.53e-09
Adam	50	-59.890	$[-64.555, -55.225]$	3.32e-10
Adam+L2-SP	C 1	0.000	$[0.000, 0.000]$	—
Adam+L2-SP	C 5	-29.316	$[-33.404, -25.228]$	5.7e-08
Adam+L2-SP	C 10	-41.093	$[-47.274, -34.911]$	1.1e-07
Adam+L2-SP	C 20	-54.276	$[-59.273, -49.279]$	1.47e-09
Adam+L2-SP	C 50	-63.914	$[-68.124, -59.705]$	7.43e-11
Uniform study	C 10	-37.966	$[-42.520, -33.412]$	1.52e-08
Uniform study	C 20	-48.518	$[-53.901, -43.135]$	7.66e-09
Uniform study	C 50	-59.425	$[-64.293, -54.558]$	5.19e-10
Uniform study	U 10	-21.051	$[-24.733, -17.369]$	4.06e-07
Uniform study	U 20	-29.949	$[-33.100, -26.799]$	4.78e-09
Uniform study	U 50	-29.825	$[-36.198, -23.453]$	2.22e-06
Inner K	$K = 1$	-52.835	$[-58.981, -46.688]$	1.16e-08
Inner K	$K = 3$	-51.127	$[-57.948, -44.305]$	3.88e-08
Inner K	$K = 5$	-49.787	$[-55.261, -44.314]$	7.06e-09
Inner K	$K = 10$	-48.303	$[-53.696, -42.909]$	8.1e-09
Inner K	$K = 20$	-46.154	$[-52.825, -39.484]$	7.79e-08
Inner K	$K = 50$	-47.053	$[-53.799, -40.307]$	7.27e-08
Pattern study	C 20	-48.110	$[-54.131, -42.089]$	2.21e-08
Pattern study	P 40	-42.828	$[-45.789, -39.868]$	1.14e-10
Pattern study	U 40	-31.166	$[-37.995, -24.338]$	2.74e-06

H SUPPLEMENTARY REPLAY AND TRAJECTORY DIAGNOSTICS

These small-cohort studies illustrate replay benefits and parameter movement. Their sample sizes and descriptive uncertainty are specified in each caption. They are separate from the ten-seed replay studies in Appendix G and the independent pre-update confirmation in Appendix F; each cohort is reported separately.

H.1 REPLAY GAINS ACROSS TWO ADAM-BASED UPDATES

Figure 3 compares replay with Adam and Adam with an L2-SP penalty around the initial parameters (anchored Adam). This penalty differs from the cumulative-gradient D-FTRL update in (25). The comparisons examine replay across two related optimizers; their ten-seed counterparts are Tables G.1 and G.2.

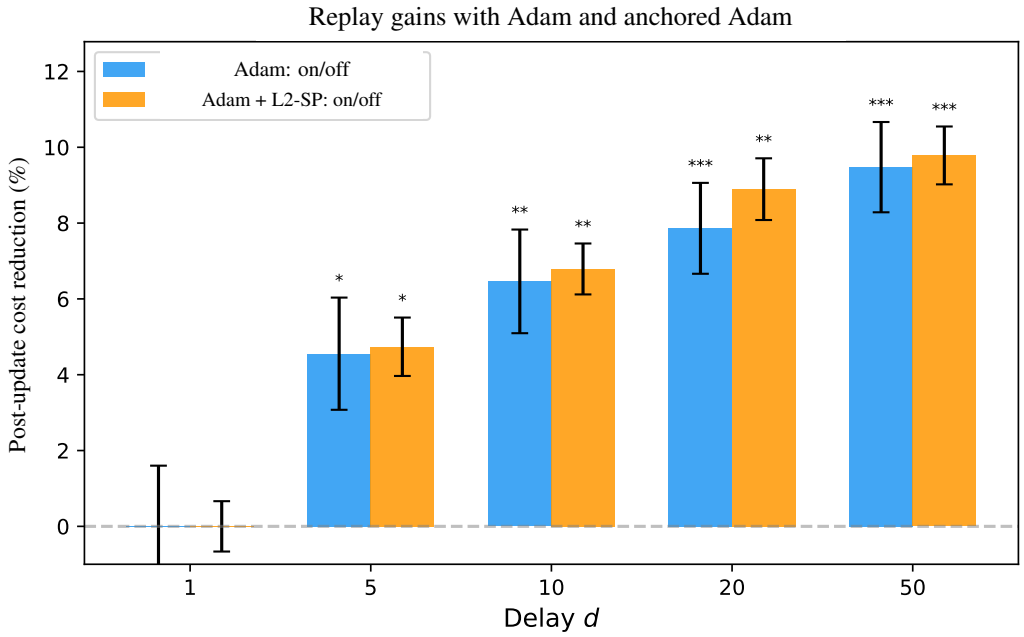


Figure 3: **Replay gains with Adam and anchored Adam.** The Adam comparison uses five seeds per cell; the anchored Adam comparison uses three. The plotted improvement is the percentage reduction of mean cumulative post-update decision cost, not comparator regret. Error bars show independently resampled bootstrap standard deviations; stars denote unadjusted Welch tests ($*$: $p < 0.05$, $**$: $p < 0.01$, $***$: $p < 0.001$). These descriptive analyses are distinct from the paired confirmation tests.

H.2 INNER-SOLVER BUDGET: PERFORMANCE AND MOVEMENT

Figure 4 shows a five-seed decision-cost comparison across inner iteration budgets. Its roughly 7.5–8.6% replay gains complement the ten-seed experiment in Table G.4, which additionally measures discrepancy from a 100-iteration solver. These are distinct cohorts; their estimates need not coincide. This plot measures decision cost, whereas the theorem's inner-solution error concerns the distance to an exact optimum.

Figure 5 shows the corresponding movement statistics. To interpret these plots, let $\Delta_k = \theta_{k+1} - \theta_k$ and let W_t be the trailing step window, whose requested length is set by the recorded outstanding count (with a minimum requested length of one). The two stored scalars are

$$S_{\text{step}} = \sum_t \sum_{k \in W_t} \|\Delta_k\|^2, \qquad S_{\text{path}} = \sum_t \left(\sum_{k \in W_t} \|\Delta_k\| \right)^2 .$$

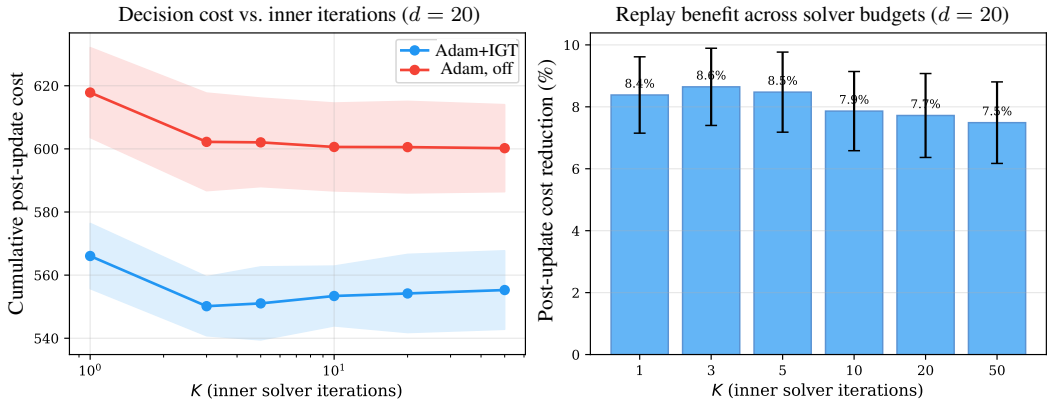


Figure 4: **Replay benefit across Sinkhorn iteration budgets.** At delay 20, the left panel reports cumulative post-update plan cost with one-standard-deviation shading over five seeds. The right panel reports percentage reductions of group means with independent-bootstrap standard-deviation bars, not paired confidence intervals. The metric is decision cost, with no comparator subtraction. The ten-seed experiment and solver-discrepancy measurement are in Table G.4.

The image labels R_3 and `sqDrift` mean S_{step} and S_{path}, respectively. They summarize parameter movement, not measured hypergradient error. In particular, S_{path} is not $\sum_t \| \sum_{k \in W_t} \Delta_k \|^2$. Their ratio is at most the largest window length by Cauchy–Schwarz, and can approach that length whenever the step norms are similar, regardless of step directions.

H.3 Delay scaling of trajectory statistics

Figures 6 and 7 show delay scaling with five seeds for each of four implementations and each delay, for 100 saved runs. Both statistics are computed on every method's own trajectory. Adam+IGT's lower movement budget is a descriptive finding; by itself, less parameter movement does not establish a smaller gradient error or better decisions. The SGD, Robust-OMD and Stale-OMD labels identify the experimental implementations used for this trajectory diagnostic; published-baseline comparisons are specified separately in Appendix D.3.

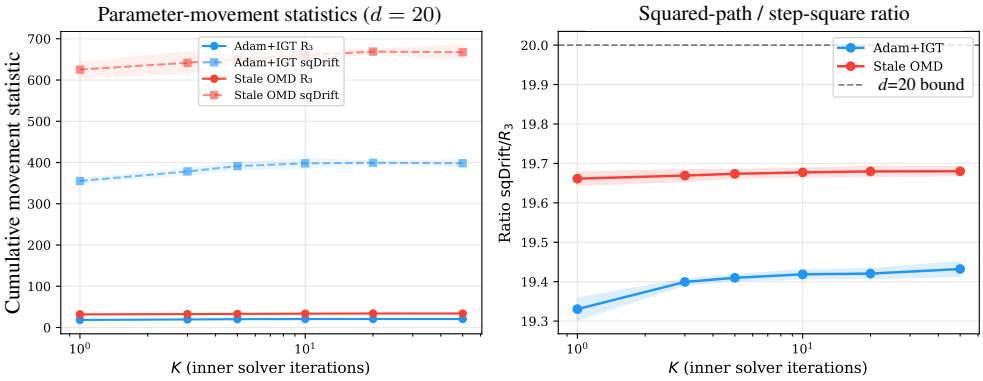


Figure 5: **Movement diagnostics across inner iteration budgets.** Curves and shading summarize five seeds. The left panel compares S_{step} and S_{path}; the right shows their ratio near the window-size bound of 20. This documents the behavior of those trajectory statistics across K, not invariance of estimation error to inner accuracy. The source artifacts retain aggregate metrics here, rather than full per-step displacement vectors.

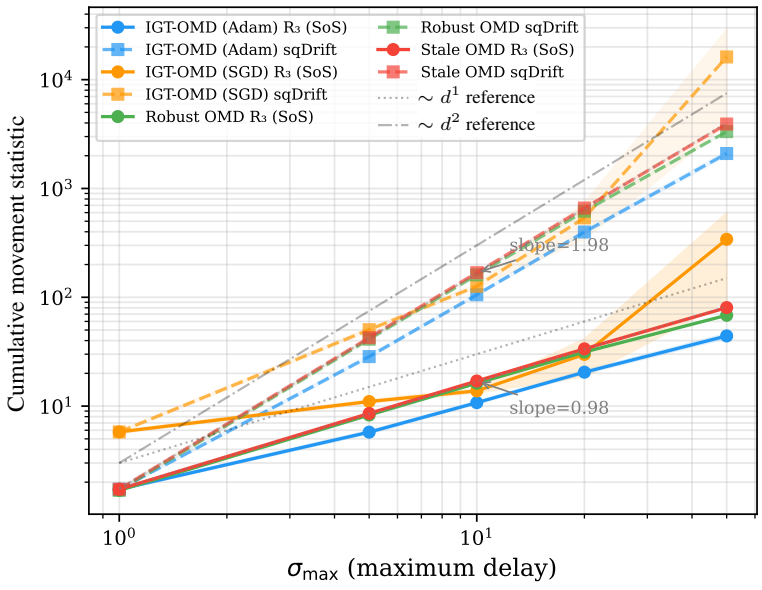


Figure 6: **Delay scaling of parameter-movement statistics.** Solid curves show S_{step}; dashed curves show S_{path}. Bands show one standard deviation across seeds. The annotations 0.98 and 1.98 are fits to the Stale-OMD-labeled implementation, not to IGT. The gray power laws are reference guides, not a test of the hypergradient-error bound.

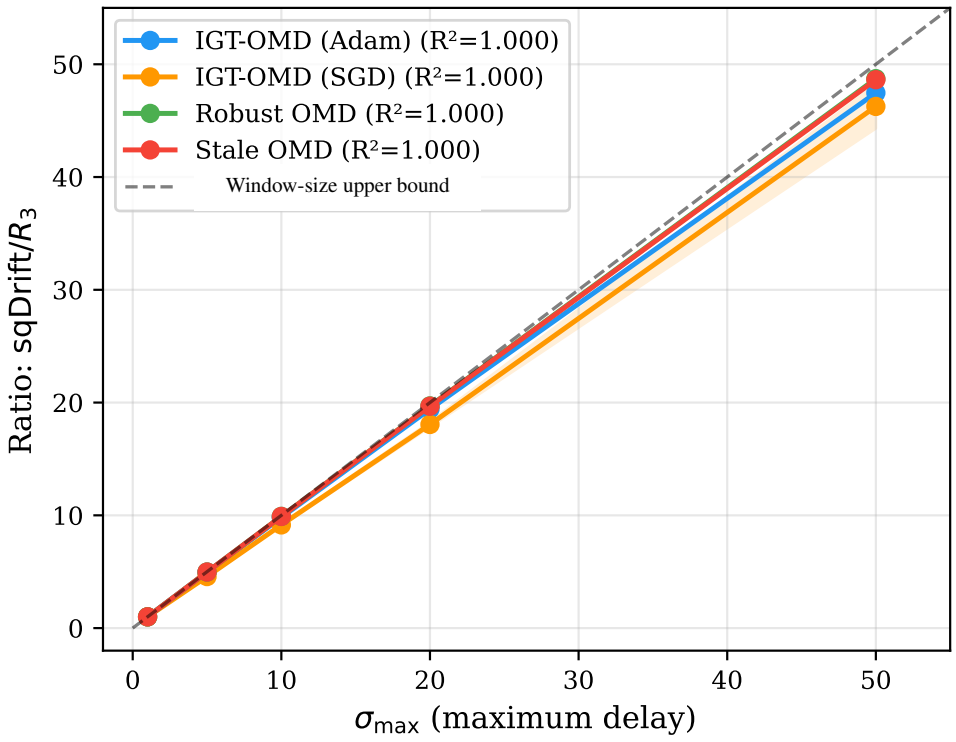


Figure 7: **Squared-path to step-square ratios.** Curves summarize per-seed ratios with standard-deviation shading. Under these constant delays, d bounds the step-window length. Ratios near d describe the concentration of step magnitudes and do not demonstrate cancellation of vector increments or a factor-d algorithmic improvement. The direct transport-increment ratio ρ_t in Section 5.3 measures a different object; its values are reported in Appendix D.

I Extended literature discussion

The closest connections concern three different objects: the decision map being differentiated, the information available when an online update is made, and the approximation retained in memory. Keeping these objects separate makes it possible to compare methods without treating a shared algorithm name as evidence of a shared guarantee.

I.1 Learning through an optimization problem

Decision-focused learning changes the training objective from prediction accuracy to downstream decision quality. Wilder et al. (2019) develop this approach for combinatorial optimization, while Elmachtoub & Grigas (2022) study a decision-aware predict-then-optimize loss and its tractable surrogate. The survey and benchmark of Mandi et al. (2024) place these approaches in a broader landscape of gradient-based and gradient-free methods. This literature motivates our choice of decision losses, but a decision loss is not automatically the fixed-comparator online regret defined in (3).

The decision layer also determines which derivative is available. Amos & Kolter (2017) differentiate quadratic-program solution maps, and Agrawal et al. (2019) extend differentiable layers to disciplined convex programs. Discrete decisions require different constructions: Vlastelica et al. (2020) provide a blackbox differentiation rule, whereas Berthet et al. (2020) use random perturbations to smooth optimizer outputs. Our Warcraft adapter follows the former construction; it does not implement the latter or a smooth strongly convex inner problem. The entropy-regularized transport layer uses Sinkhorn iterations (Cuturi, 2013). Its frozen cost-space direction is likewise identified separately from the arrival-adjoint oracle in our smooth theory. Our full re-evaluation variant differentiates the complete truncated Sinkhorn computation at the current prediction, rather than keeping that cost-space derivative fixed. These distinctions explain why we test a memory mechanism across decision layers without claiming that one hypergradient theorem covers all of them.

I.2 Approximate hypergradients and online bilevel learning

Solving an inner problem only approximately is a longstanding computational choice, not a feature introduced by delayed feedback. Pedregosa (2016) analyze hyperparameter updates with inexact gradient information. Ghadimi & Wang (2018) give finite-time analyses for approximation methods with strongly convex inner problems. Franceschi et al. (2018) study a bilevel formulation that explicitly accounts for inner optimization dynamics. Grazzi et al. (2020) compare iterative and implicit hypergradient approximations under contraction assumptions, and Ji et al. (2021) analyze and improve bilevel algorithm designs. Together these works motivate accounting for both the inner-solution error and the work needed to control it.

Delay creates an additional distinction. An accurate solution for the source parameter need not be accurate for the arrival parameter. Our Theorem 2 bounds these two errors separately before bounding the memory increment. After arrival, the adjoint is deliberately frozen, so its later use is not the same as recomputing an implicit hypergradient at the current inner optimum. Theorem 3 further separates the gradient approximation from the loss of the decision actually executed. This distinction remains necessary even when a gradient estimate happens to be exact.

Online bilevel optimization addresses another source of difficulty: objectives themselves can change over time. Ataee Tarzanagh et al. (2024) develop online alternating methods and bilevel regret analyses involving variation of minimizers. Lin et al. (2023) use a window of hypergradient estimates in a single-loop method and analyze bilevel local regret. Nazari et al. (2025) give stochastic first- and zeroth-order guarantees without window smoothing. Online decision-focused learning also admits decision-specific methods (Capitaine et al., 2026). These are important adjacent results, but their regret criteria and information models must be distinguished from our stationary fixed-comparator analysis. Our nonstationary Warcraft study is an empirical test, not a proof of dynamic regret.

These methods are not interchangeable baselines. Decision-focused OGD in Capitaine et al. (2026) uses an approximate minimization oracle, a random interpolation point, and projection; zero-hint delayed OGD does not implement those operations. The alternating method of Ataee Tarzanagh et al. (2024) maintains an inner iterate and combines windowed implicit hypergradients at shared

current states, rather than adding cached gradient differences. Neither comparison is included in our experiments. Their adaptation to this feedback model remains a distinct comparative question, not an equivalence established by sharing an outer gradient step.

I.3 Delayed feedback, mirror descent, and FTRL

Mirror descent updates a decision using a local gradient and a chosen geometry (Beck & Teboulle, 2003). FTRL accumulates losses or their surrogates and a regularizer. Their relationships, including adaptive choices of regularization and step size, are developed by McMahan (2017). Both families have learning-rate parameters. A rate comparison still needs the update equations: equal scalar parameters do not equate a fixed-step gradient update with a cumulative-gradient update whose coefficient changes with the clock.

For delayed information, Joulani et al. (2013) give general reductions from non-delayed learning, and Quanrud & Khashabi (2015) analyze standard online algorithms under adversarial delays. Qiu et al. (2025) exploit curvature in delayed OCO. Flaspohler et al. (2021) combine optimism and delay, while Huang et al. (2023) develop a general mirror-descent framework for delayed bandit learning. Bandit feedback and delayed gradient feedback are different observation models; we do not report Banker-OMD as an evaluated baseline. Nor do we claim that these methods cannot be applied to bilevel losses. Our additional question is how an inexact, cached decision layer changes the gradient supplied to such an outer optimizer.

More recent work also studies delay adaptation outside a bilevel cache. Ryabchenko et al. (2026) use a continuous-time decomposition into learning and delay-induced drift to obtain reductions from immediate to delayed feedback. Qiu et al. (2026) study unknown, agent-dependent delays on a communication network, using local delay estimation and adaptive rates. Their networked setting is different from the centralized curvature-based D-FTRL baseline of Qiu et al. (2025). These results broaden the context for queue-sensitive learning, but do not identify the cost or approximation of the particular inner-state cache studied here.

I.4 What gradient transport and the DDE connection contribute

Arnold et al. (2019) motivate transporting past gradient information to reduce estimator bias and variance. Our use of that idea changes the object being transported: we evaluate differences of bilevel gradient approximations, using either a frozen surrogate or full truncated-solver re-evaluation. The estimator, cache, and clock therefore need their own analysis. In particular, transporting old information is not by itself a proof that the added correction cancels the current arrival error.

Yu et al. (2025) use stochastic delay differential equations to study stale distributed SGD. Our continuous model instead comes from the common-curvature recurrence in (10). Its lag represents one memory-update clock. Proposition 1 establishes continuous stability for that family; Proposition 2 gives the exact discrete step-size restriction. Keeping both results is useful precisely because a stable continuous model need not remain stable at an arbitrary Euler step. The LQR experiment consequently measures stable learning-rate scales, not decision regret. The Adam factorials test the practical extension separately from this Euclidean stability calculation.

I.5 Comparison at a glance

Table I.1 summarizes the distinctions used in the main related-work discussion. It compares questions and information models, not regret rates derived under incompatible assumptions.

Literature	Central object	Connection to this paper
Decision-focused learning (Wilder et al., 2019; Elmachtoub & Grigas, 2022; Mandi et al., 2024)	Predictor judged by downstream decisions	Motivates decision loss; feedback timing and cached-gradient updates must still be specified.
Differentiable decision layers (Amos & Kolter, 2017; Agrawal et al., 2019; Vlastelica et al., 2020)	Derivative or surrogate of an optimization map	Determines what is frozen or recomputed; smooth implicit and blackbox path derivatives differ.
Inexact bilevel optimization (Pedregosa, 2016; Grazzi et al., 2020; Ji et al., 2021)	Approximate inner solutions and hypergradients	Provides accuracy/compute context; delayed source drift creates an additional error.
Online bilevel learning (Ataee Tarzanagh et al., 2024; Lin et al., 2023; Nazari et al., 2025)	Changing objectives, variation and online regret criteria	Distinct from our stationary fixed-comparator theorem and empirical drifting tasks.
Delayed online learning (Joulani et al., 2013; Quanrud & Khashabi, 2015; Qiu et al., 2025; Ryabchenko et al., 2026)	Updates with information arriving late	Provides outer-learning frameworks and baselines; our estimator analysis makes the inner approximation explicit.
Gradient transport (Arnold et al., 2019)	Adjustment of past gradient information	Conceptual connection; our frozen/full bilevel objects and cache clock have their own analysis.
Delay equations (Yu et al., 2025)	Continuous dynamics with stale information	Motivates stability analysis; our discrete interval is proved independently for its specified recurrence.
This work	Delayed bilevel-gradient memory with frozen and full oracles	Conditional Euclidean theory and matched empirical effects, with their different computational costs.

Table I.1: Related-work map. Methods in one row may be combined with ideas in another; the table does not assert that prior outer optimizers cannot be used with bilevel objectives.