CoRoute: Certified Inverse Learning from Incomplete Trajectories

Anonymous authors
Paper under double-blind review

Abstract

Inverse reinforcement learning from incomplete trajectories alternates between estimating missing decisions and fitting a policy. Repeating the policy fit can be costly, but reusing a stored policy may become inaccurate as the estimated trajectories change. We introduce CoRoute, a learning procedure that checks when such reuse is justified. For finite deterministic MDPs with an absorbing goal, our main result bounds the loss from reusing a policy inside a two-terminal subgraph when both its completed edge counts and incoming flow change. Using the cached flow, it checks whether reuse meets a chosen error budget without computing the new optimal policy. It allows cycles and inexact caches, and controls both directions of divergence between complete path distributions. When only incoming flow changes, a sharp bound requires just the cached mean path exposure, even if paths can be arbitrarily long. Two supporting certificates check approximate completion and the accuracy of each policy-fitting step. Numerical studies approach the sharp bounds and reduce the cost of accepting shared learning proposals. Refining the reuse bound with one potential step certifies reuse of 275 of 768 caches, up from 234. The implemented outward-rounded acceptance tests certify improvement for the stored record kernels.

1 Introduction

Inverse reinforcement learning (IRL) infers a reward from observed decisions. In many applications, however, demonstrations are incomplete. A GPS record, for example, may show where a traveller entered and left an area without revealing the route between them. Several routes may explain the same record, with different lengths and repeated visits to a state. Summing over these possibilities preserves the uncertainty in the missing decisions (Mai et al., 2019; 2023).

This uncertainty also makes learning expensive. Each iteration estimates how often the hidden trajectories traverse each edge, then fits a policy to those completed counts. A natural way to save work is to retain a previously fitted policy inside part of the network. Yet the next completion step may change both the relative edge counts inside that subgraph and the total flow entering it. A policy that was accurate before these changes need not remain accurate afterward. We ask: *Can we bound the cost of reusing an internal policy before solving the new fitting problem?*

We study finite deterministic Markov decision processes (MDPs) with an absorbing goal. A soft action-value function defines both the policy and its inverse Bellman reward (Garg et al., 2021). We fit a regularized score that sums over paths compatible with each incomplete record. This choice matters: regularization changes the weights assigned to hidden paths. The ordinary behavioral posterior can therefore suggest an update that decreases the intended training objective.

Our main contribution is a computable bound on the error incurred by retaining a cached policy after both completed edge counts and incoming flow change. We consider a *module*, a subgraph with one entry and one exit; its *throughput* is the total mass entering it. Given a positive feasible cached flow, the bound controls fitting loss and Kullback–Leibler (KL) divergence in both directions between the cached and newly optimal complete-path laws. It allows cycles and inexact caches without requiring the new optimal policy. This makes reuse a decision governed by an explicit error budget: if the bound is below the budget, the learner can retain the policy with a guarantee on its accuracy. For fixed counts, a sharp bound needs only cached mean path exposure (e.g., expected distance), even when maximum path length is unbounded.

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.

The key idea is to change the state values used to evaluate the cached policy while keeping its action probabilities fixed. This produces an upper bound that can be tightened without changing that policy. Two supporting certificates connect this local reuse decision to the full learning iteration: a residual bound verifies improvement on the observed records, and an optimization gap measures fitting accuracy for the current completed counts. CoRoute combines these three checks (Figure 1).

Our contributions are:

- **Main result: a policy-reuse certificate.** We bound fitting loss and both directions of complete-path KL under joint count and throughput changes, including cycles and inexact caches. We characterize lossless reuse and prove a sharp cyclic bound using mean exposure alone (Section 4 and Corollary 5.1).
- **Supporting certificates for learning.** We use backward-value and visit residuals to certify record-score improvement, and translate the frozen learning gap into complete-path and recorded-data guarantees (Section 3).
- **Numerical validation and computational scope.** We test sharpness, acceptance cost and cache reuse. Acceptance checks reduce computation for shared proposals; additional cache retention does not consistently yield faster solving (Section 6).

Section 4 presents the main reuse theorem. Sections 2–3 first establish its objective and supporting certificates; Section 5 combines them into the learning procedure. Proofs and detailed protocols are in the appendices.

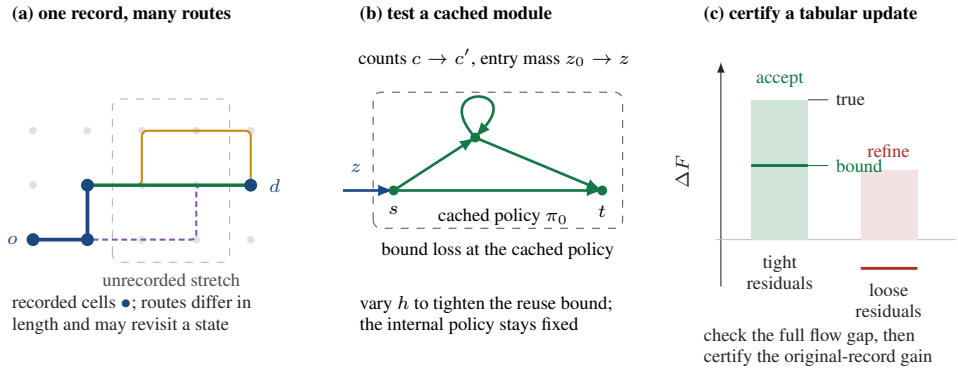

Figure 1: CoRoute's three decisions. **(a)** Estimate missing edge traversals from the routes compatible with a record. **(b)** Test whether a stored policy π_0 can be reused inside a module with entry s and exit t when counts change from c to c' and incoming mass from z_0 to z. State values h tighten the bound. **(c)** Check the proposed fit on the full graph, then certify its record-objective change ΔF. A nonnegative lower bound accepts the update; an inconclusive bound calls for refinement.

2 Problem Setup: Inverse Learning from Incomplete Trajectories

We distinguish two distributions: the policy used to predict behavior and the distribution used to complete missing decisions during training. Without regularization, completion is the usual behavioral posterior. With regularization, the two conditional distributions generally differ. The transition graph, edge exposures and recording channel are fixed inputs to fitting; the optimization variable is Q. The policy and reward are derived from Q, so an update does not require a separate forward RL solve for a proposed reward.

2.1 Policies and Inverse Rewards on an Absorbing Graph

All logarithms are natural. Let $A(s)$ be the legal actions at state s, and let $T(s, a)$ be the deterministic successor of action a. An edge $e = (s, a)$ is a state–action pair; its tail and head are denoted s_e and $t_e = T(s, a)$. Distinct actions remain distinct edges even if they share a successor. The graph is finite, the goal d has no outgoing actions, and every retained state can reach d. In a road application, states

can be road links and actions their legal successors, allowing turn restrictions and turn attributes. The pilot network instead uses cell transitions (Appendix M). Context x contains measured departure conditions and trip attributes. We fix the entropy scale to one and suppress dependence on (d, x) when unambiguous. We write $\mathbb{E}$ for expectation and $\mathrm{KL}(P\|R)$ for KL divergence between probability laws P and R.

For a finite action-value function $Q(s, a)$, the soft value V_Q and behavioral policy π_Q are

$$V_Q(s) = \log \sum_{a \in A(s)} e^{Q(s,a)}, \quad V_Q(d) = 0, \qquad \pi_Q(a \mid s) = e^{Q(s,a) - V_Q(s)}. \tag{1}$$

The learned inverse Bellman reward is

$$\boxed{r_Q(s, a) = Q(s, a) - V_Q(T(s, a)).} \tag{2}$$

This inverse Bellman relation recovers immediate utility by subtracting the successor's value from Q. The model is undiscounted: a trajectory stops at its first arrival at d. For a finite route τ starting at origin o, write $\pi_Q(e) = \pi_Q(a \mid s)$ and $p_Q(\tau) = \prod_{e \in \tau} \pi_Q(e)$. Products and sums over a route count every traversal, including repeated uses of an edge.

Finite Q gives positive probability to every legal action. Together with goal reachability on the finite graph, this ensures termination with finite expected route length, so p_Q is normalized even with loops. Appendix A proves this and the telescoping relation between path probabilities and additive rewards.

2.2 Regularized scores and incomplete observations

We use the exponential member of the inverse-Q objectives of Garg et al. (2021). This choice provides an increasing, concave transformation of reward, preserving concavity of the frozen tabular learning objective and yielding the flow formulation below. Exposure scaling makes the penalty depend on utility per unit distance in the route application. These are properties of the chosen criterion; our experiments do not establish that it predicts better than ordinary likelihood. Let $\ell_e > 0$ be a fixed exposure for edge e (such as link length), and let $\lambda > 0$ control regularization. Define the transformed reward ϕ_e and its penalty ψ_e by

$$\phi_e(r) = \frac{\ell_e}{\lambda}(1 - e^{-\lambda r/\ell_e}), \qquad \psi_e(r) = r - \phi_e(r) \geq 0. \tag{3}$$

Exposure is fixed before fitting; in the route model it is link length in a fixed unit. Scaling all exposures and λ by the same positive factor leaves ϕ_e unchanged. The complete-trajectory score is

$$S_Q(\tau) = \sum_{e \in \tau} \phi_e(r_Q(e)) - V_Q(o) = \log p_Q(\tau) - \sum_{e \in \tau} \psi_e(r_Q(e)). \tag{4}$$

At $\lambda = 0$, the continuous extension is $\phi_e(r) = r$, so S_Q becomes the route log likelihood. For small positive λ, the leading penalty is $\lambda r^2/(2\ell_e)$: it penalizes utility relative to exposure. Appendix C explains this convention and proves invariance under deterministic edge subdivision when the inserted values are freely optimized.

The learner observes a record rather than a complete trajectory. Let y_i, $i = 1, \ldots, N$, be the observed records, with their origins, goals and contexts held fixed. The recording channel $\kappa(y \mid \tau)$ is the probability of observing y from route τ; it is discrete and independent of Q. Assume each retained record has positive support. Define its total score C_Q, the training objective F, and the completion distribution q_Q by

$$\begin{aligned} C_Q(y) &= \sum_{\tau : o \to d} \kappa(y \mid \tau) e^{S_Q(\tau)}, \qquad F(Q) = \sum_i \log C_Q(y_i), \\ q_Q(\tau \mid y) &= \frac{\kappa(y \mid \tau) e^{S_Q(\tau)}}{C_Q(y)}. \end{aligned} \tag{5}$$

The completion q_Q supplies expected edge counts for training, while p_Q supplies behavioral predictions. Write $p_Q(y) = \sum_\tau \kappa(y \mid \tau) p_Q(\tau)$ for the behavioral observation likelihood and $p_Q(\tau \mid y)$ for its posterior. Then

$$C_Q(y) = p_Q(y)\, \mathbb{E}_{p_Q(\tau|y)} \exp\left\{ -\sum_{e \in \tau} \psi_e(r_Q(e)) \right\}. \tag{6}$$

At positive regularization, $C_Q(y)$ is an attenuated score and need not sum to one over records. We optimize this specified score, rather than treating it as a normalized observation likelihood. Because the penalty is inside the path sum, it changes the completion weights as well as the objective. At zero regularization, both constructions reduce to ordinary likelihood and posterior inference. At positive regularization, the completion still equals the behavioral posterior if the total penalty is constant across all routes compatible with the record.

From completion to counts. Completion averages over possible missing decisions rather than choosing a single reconstructed route. Let $n_e(\tau)$ count traversals of edge e, including revisits. At current parameters Q_0, the completed counts for one fixed destination and context are

$$c_e = \sum_i \mathbb{E}_{q_{Q_0}(\cdot|y_i)}[n_e(\tau)],$$

where the sum includes records in that fitting block. For example, if a record admits two routes with completion probabilities 0.7 and 0.3, an edge used once only by the first route receives count 0.7; an edge used once by both receives count 1. The next learning step treats these fractional counts as fixed data.

3 From approximate completion to certified learning

This section develops the supporting checks needed to use the reuse certificate within a learning iteration. We identify the completion law, bound record-score improvement under approximate completion, and give a flow-gap test for fitting accuracy. The last identity underlies the main reuse result in Section 4.

3.1 Which completion law matches the objective?

The score-induced law in (5) makes the usual variational argument apply to F. Moving the penalty outside the path sum gives a different criterion, compared in Appendix B.

At current parameters Q_0, hold the completion laws $q_{Q_0}(\cdot \mid y_i)$ fixed while updating Q. The resulting *frozen* objective is the expected complete-trajectory score. Increasing it cannot decrease F: the usual expectation–maximization (EM) variational argument applies to this score (Dempster et al., 1977; Neal & Hinton, 1998). This frozen objective is concave in tabular or affine Q, although the full incomplete-record objective need not be concave. Appendix D gives the exact identity and assumptions.

The distinction is consequential even with only two hidden routes: Appendix E constructs an update that improves the ordinary-posterior surrogate but decreases F. The completion used in training must therefore match the regularized score in (5).

3.2 Certifying an update from approximate completion

Exact completion can require expensive linear solves, especially when paths revisit the same states. CoRoute instead computes approximate backward values, which summarize the score of compatible continuations, and approximate expected visit counts. Their residuals measure how well they satisfy the two linear systems.

Theorem F.1 in Appendix F combines these residuals into a computable lower bound $b_i(Q^+, Q)$ for each record's score change under a proposed update Q^+:

$$F(Q^+) - F(Q) \;\geq\; \sum_{i=1}^{N} b_i(Q^+, Q).$$

Equation (20) gives the bound explicitly. It combines an upper bound on the old backward value, the estimated change accumulated along a completed path, and an allowance for visit-count error. A nonnegative total certifies acceptance. A negative total is inconclusive: refining the solves may make the same proposal certifiable.

Both error sources matter on cyclic graphs. A small local backward residual can accumulate over many visits and coexist with arbitrarily large path error (Proposition F.2). The certificate accounts for this accumulation using a verified bound on expected visits. It requires a recording channel with a finite-state realization and common support before and after the update. It applies to any finite Q, including neural parameterizations; it does not require an exact completion solve or a tabular optimum.

3.3 From a frozen learning gap to complete-path accuracy

Once completion is fixed, we fit a policy to its expected counts. A policy also induces expected edge-traversal counts: at an intermediate state, incoming flow equals outgoing flow. Optimizing these flows gives an equivalent tabular fitting problem and an error bound without enumerating trajectories. The completed counts describe inferred demonstrations; the fitted flow can differ because the objective also rewards policy entropy. For one fixed destination and context, freeze the completion distributions and let c_e be the resulting expected number of traversals of edge e. Let D be the edge-by-state incidence matrix, with $+1$ at an edge's tail, -1 at its head, and the goal column removed. Thus $D^\top c = b_0$ expresses flow conservation, where b_0 is the vector of origin masses. Assume $c_e > 0$ on the fitted graph, $b_0 \geq 0$, and total mass $B_0 = \sum_s b_{0,s} > 0$. Throughout this flow formulation, $\lambda > 0$. Write $k_e = \ell_e/\lambda$. For $u \geq 0$ and $v > 0$, the scalar generalized KL divergence is $D_{\mathrm{g}}(u\|v) = u\log(u/v) - u + v$, with its continuous value at $u = 0$. For a candidate fitted edge flow f, let $f_s = \sum_{e:s_e=s} f_e$. The frozen learning objective and its flow counterpart are

$$L_c(Q) = \sum_e c_e \phi_e(r_Q(e)) - b_0^\top V_Q, \qquad J_c(f) = \sum_e k_e D_{\mathrm{g}}(f_e\|c_e) + \sum_e f_e \log \frac{f_e}{f_{s_e}}, \tag{7}$$

The first term of J_c penalizes deviations from the completed counts; the second is negative conditional policy entropy. A state with $f_s = 0$ contributes zero to the entropy term. Counts c describe the completed demonstrations; flow f describes the policy being fitted. Flow duality (Garg et al., 2021; Fosgerau & Sørensen, 2026) makes the difference $\Delta = J_c(f) - L_c(Q)$ a nonnegative, computable fitting gap for any feasible f and finite Q. Let P^* denote the complete-path law of the exact unrestricted tabular fit and P_Q that of the candidate, both with origins drawn from b_0/B_0. Under the assumptions above, Theorem G.1 in Appendix G gives

$$\mathrm{KL}(P^*\|P_Q) \leq \frac{\Delta}{B_0}. \tag{8}$$

Thus $\Delta/B_0 \leq \varepsilon$ is a stopping rule with a path-level meaning, including repeated visits on cycles. The appendix proves duality and uniqueness and gives the gap decomposition and its consequences for recorded-data error and reconstructed flows. This is accuracy relative to the current frozen fit, not a guarantee of reward recovery or global convergence.

4 Main result: certified reuse under joint changes

We now give the main result: a bound that lets the learner check whether a cached internal policy meets a chosen reuse-error budget after counts and throughput change. We keep the policy fixed and vary only the state values used to certify it. The bound uses the cached flow and the new problem, without computing the new optimal policy.

4.1 The cached policy and the cost of reuse

A two-terminal module has entry s, exit t, no other boundary and no internal origins; no edge enters s or leaves t, every retained edge lies on a finite s–t walk, and internal cycles are allowed. This is a structural restriction on where the reuse theorem applies. Modules must be identified and checked before caching; the theorem does not assert that an arbitrary network admits a useful such decomposition. Use D for this module's incidence matrix with the exit column removed, and let e_s be the unit vector at its entry. The unit-throughput flow set is $\mathcal{W} = \{w \geq 0 : D^\top w = e_s\}$. Throughput $z > 0$ is the mass entering the module. A unit flow w_e counts expected traversals per entering episode, so actual flow is $f_e = z w_e$. Repeated visits can make $w_e > 1$. Write $w_v = \sum_{e:s_e=v} w_e$ for outgoing unit flow at state v.

Counts and throughput play different roles. Reference counts c come from completed demonstrations, whereas z is the incoming flow in the candidate fitted solution. The latter can change while the frozen counts stay fixed, as the learner adjusts choices outside the module. Between completion steps, the counts can change as well.

Use J_c from (7) on the isolated module: outgoing sums include only module edges, and the entry mass is z. This is the internal fitting cost. The full network also includes an entry-choice term for competition with external actions; that term cancels when comparing internal policies at the same throughput.

Let $n(\tau) = (n_e(\tau))_e$ count edge traversals and define $L(w) = \sum_e \ell_e w_e$, the mean path exposure of a positive unit flow. The exposure of a single path is $L_\tau = \sum_e \ell_e n_e(\tau)$. Appendix H expands the module objective and explains how exposure makes the optimum depend on throughput.

4.2 A CERTIFICATE FOR JOINT CHANGES

Suppose a positive unit flow $w_0 \in \mathcal{W}$ has been cached, with policy $\pi_0(e \mid s) = w_{0,e}/w_{0,s}$ and internal path law P_0. For new positive counts c' and throughput z, let w_1 be the optimal unit flow for $J_{c'}(zw)$, with path law P_1. The loss from retaining the cached policy is

$$\mathcal{E} = J_{c'}(zw_0) - J_{c'}(zw_1). \tag{9}$$

Although w_1 defines the reference loss, the certificate below uses only the cached flow and the new fitting problem. Exposures and $\lambda > 0$ remain fixed, with $k_e = \ell_e/\lambda$.

Let h assign a real value to each module state, with $h(t) = 0$. Then $(Dh)_e = h(s_e) - h(t_e)$ is a state-potential difference. Set $Q_h(e) = \log \pi_0(e \mid s_e) + h(s_e)$. The same value is added to every action at a state, so softmax normalization leaves the cached policy unchanged. We can therefore adjust h to tighten a fitting-gap bound while retaining that policy. Define the cached-flow residual θ_e, the nonnegative scalar function $\mathfrak{e}$, and the bound Ψ by

$$\theta_e = \log \frac{zw_{0,e}}{c'_e} + \frac{\log \pi_0(e \mid s_e)}{k_e}, \qquad \mathfrak{e}(x) = e^{-x} - 1 + x,$$
$$\Psi(h; \theta, w_0) = \sum_e k_e w_{0,e}\, \mathfrak{e}\left(\theta_e + \frac{(Dh)_e}{k_e}\right). \tag{10}$$

All inputs are known: the cached flow and policy, the new counts and throughput, and the chosen values h. Even $h = 0$ gives a valid bound. Optimizing h can tighten it while leaving the cached policy unchanged. More precisely, $k_e\theta_e$ is the derivative of the new internal cost per unit throughput, $J_{c'}(zw)/z$, at the cached flow. The potential differences account for flow conservation, so the arguments of $\mathfrak{e}$ measure mismatch with the constrained optimality conditions.

Theorem 4.1 (A joint-change reuse certificate)**.** *For the fixed module, positive exposures and $\lambda > 0$, every positive feasible cached flow and finite potential h satisfy*

$$\frac{\mathcal{E}}{z} + \mathrm{KL}(P_1 \| P_0) \le \Psi(h; \theta, w_0), \qquad \mathrm{KL}(P_0 \| P_1) \le \frac{\mathcal{E}}{z}. \tag{11}$$

Consequently $z\Psi(h; \theta, w_0)$ bounds reuse loss, and $\Psi(h; \theta, w_0)$ bounds the sum of the two path KLs. These conclusions allow internal cycles and arbitrary positive c', z.

To use the theorem, choose an allowed reuse loss $\varepsilon_{\mathrm{reuse}}$. Keep the cached policy if $z\Psi(h; \theta, w_0) \le \varepsilon_{\mathrm{reuse}}$. Otherwise, improve h or refit the module. The test also bounds how much the complete internal path distribution changes, including repeated visits. These KLs compare module policies; the full-policy gap and record-completion certificate concern different distributions.

The residual θ includes error in an inexact cache. Appendix H characterizes when the optimized bound vanishes and gives a simpler expression for an exact old fit. In particular, scaling both reference counts and throughput by the same factor preserves an exact old optimum. Changing their relative proportions can instead change the fitted policy.

Why the bound works. Evaluating the full fitting gap at Q_h and flow zw_0 gives $z\Psi(h; \theta, w_0)$. Because Q_h preserves the cached policy, the policy-mismatch term vanishes. The remaining gap decomposes into reuse loss, reverse path KL and nonnegative edge terms; expanding at the new optimum gives the other KL bound. Appendix H provides these identities and the exact zero-loss condition.

5 CONSEQUENCES FOR LEARNING AND NETWORK REDUCTION

The general certificate gives simpler rules when counts are fixed or path exposures have additional structure. We then combine these rules with the completion and learning checks.

5.1 CYCLIC REUSE FROM MEAN PATH EXPOSURE

Corollary 5.1 (A sharp cyclic bound from mean exposure)**.** *For an exact reference fit at* (c, z_0) *with* $c' = c$*, every* $z > 0$ *satisfies*

$$\boxed{\mathcal{E} \le \frac{L(w_0)}{\lambda} D_{\mathrm{g}}(z\|z_0).} \tag{12}$$

The coefficient one is uniformly sharp, even for a fixed $\lambda > 0$ *and a fixed flow ratio* $z/z_0 \neq 1$*.*

The bound is the product of two interpretable terms: the cached mean exposure $L(w_0)/\lambda$ and the discrepancy $D_{\mathrm{g}}(z\|z_0) = z\log(z/z_0) - z + z_0$ between throughputs. It vanishes when throughput is unchanged. The coefficient cannot be uniformly reduced: a three-state module with a repeated internal loop approaches equality. Thus a finite certificate is available without bounding the longest path or solving the new internal learning problem. The potential form also handles an inexact cached flow through its explicit residual θ. Appendix H gives the exact remainder and the sharp family.

Exact reuse. With counts fixed, the optimal policy is independent of throughput exactly when every complete module walk has the same exposure. Appendices I and J give this characterization, the exact network reduction, and an additional bound for acyclic modules.

5.2 THE COROUTE LEARNING PROCEDURE

We now combine the three certificates. Completion controls whether a proposed update improves the record objective; the full flow gap controls accuracy within the frozen learning block; the module bound decides whether an old internal policy is an adequate proposal.

Check	What is controlled	Reference for comparison
Record acceptance	Change in the record score	Current parameters
Full-flow gap	Accuracy of the fitted policy	Exact fit to current completed counts
Module reuse	Loss from retaining an internal policy	New module optimum at specified counts and throughput

These references are numerical learning targets, not the unknown expert reward or policy.

CoRoute: one certified iteration

1. Complete. Estimate backward values and expected visits for each record. Construct verified bounds for both approximation errors (Appendix F).

2. Propose with reuse. Aggregate positive balanced counts (Appendix K). In a tabular block, hold cached internal policies fixed while forming a candidate flow and its module throughputs. Check reuse bounds at those counts and throughputs; tighten potentials or refit modules when needed. Recheck affected bounds if throughputs change. Reconstruct Q^+ on the original graph and test the full-flow gap with Theorem G.1.

3. Certify. Sum the record-gain bounds $b_i(Q^+, Q)$ from equation (20). Accept a nonnegative total after numerical error allowances, committing a refreshed cache only with that update. For a tabular block, a failed gap or record test can trigger a full-flow fallback and the same acceptance check. An unresolved proposal keeps the previous Q and cache.

Return. The updated policy π_Q and inverse reward r_Q.

Local reuse budgets do not replace the full-flow and record-acceptance checks: the full gap is evaluated after reconstructing Q^+ on the original graph, and acceptance uses the original records. Appendix K gives the reconstruction and budget normalization. A failed bound is inconclusive and calls for refinement or refitting.

The record test accepts any finite Q class with common support, including shared features or neural models; the exact flow optimizer in Theorem G.1 remains the unrestricted tabular reference. Balanced count proposals, sparse response solves and module cache keys are specified in Appendix K.

Measured implementation. The implemented arithmetic guarantees concern the score of stored record kernels; kernel construction and flow reconstruction are audited numerically. Appendix K explains the additional error bounds needed to certify the ideal objective.

6 EXPERIMENTS

Our experiments ask three questions: how closely can examples approach the proved bounds, does the residual certificate reduce the cost of accepting an update, and when is retaining a cached policy computationally useful? The studies evaluate numerical accuracy and computation for the specified fitting objectives. They do not establish predictive superiority over other IRL methods. Appendix N specifies hardware, numerical precision, sampling, stopping and timing protocols. The eight-state learning tests use float64 margins with independent audits; the sparse and integrated acceptance tests use outward-rounded bounds for stored kernels. The generic module study audits flow gaps numerically.

Tightness of the path and reuse bounds. We compare each certificate with the corresponding independently computed error or loss, retaining cycles and repeated edge visits. Appendix N gives the reference-solve checks and workloads. Table 1 reports covariance response and reuse-gap checks on directed acyclic graphs (DAGs), cyclic modules and equal-exposure controls, together with joint-change checks. Realized loss reaches 0.9995 of the DAG range bound (37), 0.99999975 of the cyclic potential bound (11), and 0.99999775 of the cyclic mean-exposure bound (12). Each ratio is the actual loss divided by its bound; values close to one measure how closely the instances approach the proved sharp constants.

Why the completion law matters. This test isolates the choice of completion while keeping the intended regularized objective fixed. Among 16,384 parameter and record configurations on one complete seven-node DAG, 343 frozen behavioral-posterior updates decrease the intended regularized objective. At $\lambda = 0$, score and behavioral posteriors coincide; at positive regularization the path penalty changes the completion needed for a variational minorizer.

Refinement and learning. We first vary only the accuracy of completion, then compare acceptance rules on the same proposed learning updates. The cached-tight-score baseline retains an accurate old record score and solves for the new score; the direct-interval baseline refines score bounds until they decide acceptance. Our residual test instead uses the completion estimates already available. Weak absorption means that termination is unlikely at each step, making paths longer and the completion solves harder. Refinement certifies more of the same improving proposals (Figure 2a; counts in Appendix M). We also run 40 gradient updates on each of 256 cyclic models under regular and weak absorption. Cached tight scores, direct score intervals and the two-residual certificate accept all 10,240 proposals in each regime and return identical final Q. Independent dense audits of every accepted change find no decrease above 10^{-9}. Appendix N also reports tests with decreasing proposals and corrupted visit estimates.

Computation. In the eight-state tests, the certificate reuses the E-step estimates without additional fixed-point iterations. Under weak absorption, median warm loop times are 0.788 s for the certificate, 0.853 s for direct score intervals and 1.010 s for cached tight scores. Figure 2b and Table 3 retain all rules in both regimes. They also agree on all 960 updates over three sparse 64-state graphs with portal-only absorption, which require weighted drift witnesses (Table 4).

Reuse and potential refinement. Here the integrated learning runs may follow different parameter trajectories, whereas the later frozen-module queries hold each input problem fixed to isolate the reuse decision. Warm full solving starts from an available flow and allows it to change; a potential-refinement step adjusts only the state values used to certify the cached policy. At a common 0.001 full-gap target and record acceptance test, the integrated procedure with cache checks and full-solve

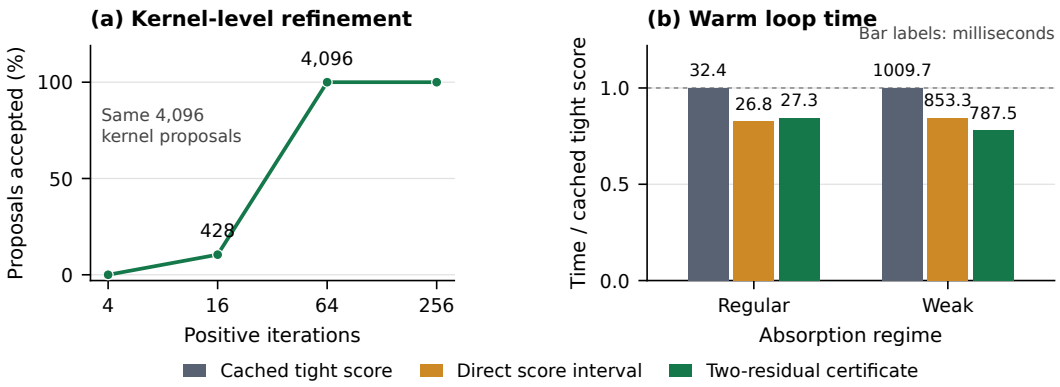

Figure 2: **(a)** Refinement certifies more of the same 4,096 improving kernel proposals; exact solves audit the bounds but are not used to construct them. **(b)** Warm cyclic-learning time relative to cached tight scores. All three rules use shared proposals and return identical final Q in each regime. Absolute times and sweep counts are in Table 3.

fallback accepts 187 updates, including 62 reuses after joint count and throughput changes. This comparison does not hold final objectives fixed. Full solving is fastest on this five-state fixture (Table 7). We also test 768 frozen cyclic-module queries whose record products have up to 225 states. At common per-query gap targets, one potential step retains 275 caches instead of 234; 34 of the 41 additional cases have unchanged counts and throughput. Median batch time is 1.2423 times warm full solving, and four steps retain no additional caches (Figure 4). Valid reuse and reduced computation are therefore measured separately.

7 Related work

Recursive route models marginalize missing decisions (Fosgerau et al., 2013; Mai et al., 2019; 2023), and IQ-Learn supplies the inverse Bellman parameterization (Garg et al., 2021). CoRoute builds on these formulations. Its completion law follows the standard variational argument for the specified penalized score; the additional results concern certified numerical approximation and reuse of the fitted policy.

Flow duality, constrained sensitivity (Fosgerau & Sørensen, 2026), policy-reuse/KL identities (Schlaginhaufen & Kamgarpour, 2026) and entropic cost-stability bounds (Eckstein & Nutz, 2022) supply related tools. Existing policy-reuse bounds can also avoid computing the new optimum. Our distinction is a certificate for positive feasible cached flows, including inexact caches, under joint count and throughput changes in cyclic modules. The sharp mean-exposure bound does not require a finite maximum path exposure. Appendix L gives detailed comparisons.

8 Discussion and conclusion

CoRoute makes policy reuse a verifiable decision in inverse learning from incomplete trajectories. Its main certificate uses a cached flow to bound fitting loss and both directions of path KL after counts and throughput change, including cycles and inexact caches. For fixed counts, the sharp mean-exposure bound remains applicable even with unbounded path lengths. Together with completion and full-flow checks, these results connect local reuse decisions to certified learning updates. The experiments validate the bounds, demonstrate lower acceptance costs on controlled learning workloads, and show that potential refinement can certify additional cache reuse. These findings provide a foundation for accuracy-controlled reuse, while consistent end-to-end speedups remain a target for further study. Appendix K.1 discusses the current scope and directions toward broader models, large-network evaluation, and end-to-end numerical certification.

STATEMENT OF USE OF AI

The research questions, the main ideas, the methodology, and the experimental protocols of this paper were chosen by the authors, who divided the work among themselves, supervised every stage of the project, and made all decisions on claims and conclusions. Under the authors' direction, we used generative AI tools as assistants for the following tasks with required disclosure: refining the conceptual framing, stress-testing mathematical claims and proof arguments, and implementing and running the experiments. In the theoretical component, generative AI was used in part as a "theory attacker": it assisted in implementing computational searches and large-scale testing over extensive collections of generated examples in order to identify potential counterexamples, edge cases, or weaknesses in candidate theorem statements and proof arguments. These searches were used to help the authors examine and refine the theoretical claims. The failure of such searches to identify a counterexample does not constitute a proof, and no theorem in this paper is considered established on the basis of AI-assisted simulation, search, or testing.

The authors independently verified all theorem statements and proofs, citations, experimental specifications, and AI-assisted text against their research artifacts. No experimental data or results were produced by generative AI tools themselves: every reported number comes from experiments that were actually executed, and translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools for literature discovery and for drafting and editing the manuscript, including checks of notation and of the consistency of reported numbers. The authors take full responsibility for every claim, proof, citation, and experimental result in this paper; no AI system is an author.

ETHICS STATEMENT

The motivating data construction uses a restricted archive of GPS travel diaries collected with participant consent under a data-use agreement. The numerical experiments also use synthetic graphs. The decrypted records, raw coordinates and participant identifiers are held only on an access-controlled analysis machine and are not part of this submission or its code archive; per-traveller losses are stored under hashed identifiers, and an automated gate rejects any exported artifact containing an identifier, a trip id, a coordinate or a trajectory. Every number in this paper is an aggregate. Route-choice models of this kind can in principle be used to profile individuals; the released artifacts are aggregate statistics and code, not fitted per-participant models.

REPRODUCIBILITY STATEMENT

Section 6 and Appendix N give the numerical protocols and network construction, including graph families, recording laws, proposal rules, solver tolerances, independent references and charged timing phases. The synthetic studies require no private records. The pilot network and module study additionally use the restricted archive under its data-use agreement. All proofs are in the appendices. The supplementary material provides the source code for all experiments and figures, together with the recorded outputs of the synthetic studies; it contains no records from the restricted archive.

REFERENCES

Kenneth Bogert and Prashant Doshi. Multi-robot inverse reinforcement learning under occlusion with interactions. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 173–180, 2014.

Raphaël Chetrite and Hugo Touchette. Nonequilibrium markov processes conditioned on large deviations. *Annales Henri Poincaré*, 16:2005–2057, 2015. doi: 10.1007/s00023-014-0375-8. URL https://arxiv.org/abs/1405.5157.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Stephan Eckstein and Marcel Nutz. Quantitative stability of regularized optimal transport and convergence of Sinkhorn's algorithm. *SIAM Journal on Mathematical Analysis*, 54(6):5922–5948, 2022. doi: 10.1137/21M145505X. URL https://www.math.columbia.edu/~mnutz/docs/quantStability.pdf.

Mogens Fosgerau and Jesper Riis-Vestergaard Sørensen. Convex duality in perturbed utility route choice, 2026. URL https://arxiv.org/abs/2604.20220. Preprint.

Mogens Fosgerau, Emma Frejinger, and Anders Karlström. A link based network route choice model with unrestricted choice set. *Transportation Research Part B: Methodological*, 56:70–80, 2013. doi: 10.1016/j.trb.2013.07.012.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, Matthieu Geist, and Stefano Ermon. IQ-Learn: Inverse soft-Q learning for imitation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4028–4039, 2021. URL https://proceedings.neurips.cc/paper/2021/file/210f760a89db30aa72ca258a3483cc7f-Paper.pdf.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4565–4573, 2016.

Kris M. Kitani, Brian D. Ziebart, J. Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision (ECCV)*, pp. 201–214, 2012.

Tien Mai and Emma Frejinger. Undiscounted recursive path choice models: Convergence properties and algorithms. *Transportation Science*, 56(6):1469–1482, 2022. doi: 10.1287/trsc.2022.1145.

Tien Mai and Thuy Anh Ta. Occupancy-based network reduction for recursive-logit route-choice estimation, 2026. URL https://ssrn.com/abstract=7372485. Working paper, posted 30 August 2026.

Tien Mai, Quoc Phong Nguyen, Kian Hsiang Low, and Patrick Jaillet. Inverse reinforcement learning with missing data, 2019. URL https://arxiv.org/abs/1911.06930. Preprint.

Tien Mai, The Viet Bui, Quoc Phong Nguyen, and Tho V. Le. Estimation of recursive route choice models with incomplete trip observations. *Transportation Research Part B: Methodological*, 173: 313–331, 2023. doi: 10.1016/j.trb.2023.05.004. URL https://arxiv.org/abs/2204.12992.

Radford M. Neal and Geoffrey E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan (ed.), *Learning in Graphical Models*, pp. 355–368. Kluwer Academic Publishers, 1998. doi: 10.1007/978-94-011-5014-9_12. URL https://www.cs.toronto.edu/~hinton/absps/emk.pdf.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pp. 663–670, 2000.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999. URL https://ai.stanford.edu/~ang/papers/shaping-icml99.pdf.

Andreas Schlaginhaufen and Maryam Kamgarpour. Fast rates for inverse reinforcement learning, 2026. URL https://arxiv.org/abs/2605.14599.

Joar Max Viktor Skalse, Matthew Farrugia-Roberts, Stuart Russell, Alessandro Abate, and Adam Gleave. Invariance in policy optimisation and partial identifiability in reward learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32033–32058, 2023. URL https://proceedings.mlr.press/v202/skalse23a.html.

Arjan van der Schaft, Bart Besselink, and Anne-Men Huijzer. Kron reduction of nonlinear networks. *IEEE Control Systems Letters*, 8:556–561, 2024. doi: 10.1109/LCSYS.2024.3398194. URL https://arxiv.org/abs/2403.00400.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. doi: 10.1561/2200000001. URL https://people.eecs.berkeley.edu/~jordan/sail/readings/wainwright-jordan-fnt.pdf.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

GUIDE TO THE APPENDICES

The appendices answer three different questions; they need not be read in sequence. For the main contribution, start with Appendix H: it proves the joint-change reuse bound and its cyclic mean-exposure consequence. For the learning procedure, read Appendix K, then consult Appendices F and G for the two supporting certificates. For the evidence, use Appendices M and N, which describe the data construction and experimental protocols.

Appendices A–E supply modeling background. Appendices I–J are optional fixed-count refinements: they explain throughput sensitivity and exact network reduction, but are not prerequisites for applying the joint-change certificate. Throughout, completed counts c are the fitting targets, flows f are optimization variables, and a module's unit flow w is scaled by throughput z. The distinction between completion and behavioral path laws is summarized in Appendix K.

A ABSORBING PATHS AND SCORE IDENTITIES

This appendix establishes that the behavioral policy defines a normalized distribution over complete routes, even when the graph contains cycles. We first show that routes terminate with finite expected length, then derive the reward–path identity used to construct the training score.

Fix a finite directed decision graph, a destination d with no outgoing decisions, and a finite number of legal actions at every other state. Every retained state can reach d. Q-values on all legal actions are finite; the temperature is one. Thus every legal action has positive probability under the softmax policy.

Proposition A.1 (Proper path model)**.** *Every finite Q on the stated graph gives an almost surely terminating route policy with finite expected route length, and*

$$\log p_Q(\tau) = \sum_{e \in \tau} r_Q(e) - V_Q(o). \tag{13}$$

Choose, for every nonterminal state, one path to d. Let L be their maximum length and $\epsilon > 0$ the smallest policy probability of following one of these paths. Finiteness of the graph and positivity of every factor give $\epsilon > 0$. Starting from any transient state, the probability of not reaching d during the next L steps is at most $1 - \epsilon$. The Markov property therefore gives $\Pr(T > kL) \le (1 - \epsilon)^k$. In particular, $T < \infty$ almost surely, $\mathbb{E}T < \infty$, and the transient policy matrix has spectral radius below one. This is a pointwise statement for each finite Q; it does not provide a uniform iteration bound as unbounded Q-values approach a trapping policy.

For a route $\tau = (s_0, a_0, \ldots, s_T = d)$, write $r_Q(s, a) = Q(s, a) - V_Q(T(s, a))$, with $V_Q(d) = 0$. Then

$$\begin{aligned} \sum_{t<T} \log \pi_Q(a_t \mid s_t) &= \sum_{t<T} \{Q(s_t, a_t) - V_Q(s_t)\} \\ &= \sum_{t<T} r_Q(s_t, a_t) - V_Q(s_0). \end{aligned}$$

The complete path distribution is normalized because the policy terminates almost surely. Consequently the reward path partition is exactly $\sum_{\tau: o \to d} \exp\{\sum_{e \in \tau} r_Q(e)\} = e^{V_Q(o)}$. This supplies the absorbing, undiscounted counterpart to the inverse-Bellman parameterization used by Garg et al. (2021), in the recursive route-choice setting of Mai & Frejinger (2022).

B REGULARIZED PATH SCORES AND THEIR MARGINALISATION

This appendix explains how the complete-data inverse-learning criterion extends to incomplete observations. It establishes that the resulting record score is finite and explains why it is a training score rather than a normalized behavioral likelihood.

Let ϕ be finite, concave and nondecreasing, with $\phi(r) \le r$, and set $\psi = r - \phi$. Then $\psi \ge 0$ is convex. For complete demonstrations, the episodic IRL criterion is

$$\sum_e d_E(e) r(e) - \sum_o \eta_E(o) V_r(o) - \sum_e d_E(e) \psi(r(e)),$$

where d_E is the mean undiscounted edge-count measure and η_E the empirical origin law. Substituting r_Q and using the preceding partition identity yields the expert average of $S_Q(\tau) = \sum_e n_e(\tau)\phi(r_Q(e)) - V_Q(o)$. Counts are per episode; no discounted occupancy normalization is inserted.

For $\phi_\lambda(r) = (1 - e^{-\lambda r})/\lambda$, $\phi'_\lambda = e^{-\lambda r} > 0$, $\phi''_\lambda = -\lambda e^{-\lambda r} < 0$, and $e^{-\lambda r} \geq 1 - \lambda r$ proves $\phi_\lambda(r) \leq r$. The zero-λ limit is r. These properties are the ones needed for the frozen learning objective below.

Fix a discrete recording channel $\kappa(y \mid \tau)$, independent of Q, with values in $[0, 1]$, and take an observation of positive support. Since $e^{S_Q(\tau)} = p_Q(\tau)e^{-\sum_e n_e \psi(r_Q(e))} \leq p_Q(\tau)$,

$$0 < C_Q(y) = \sum_\tau \kappa(y \mid \tau)e^{S_Q(\tau)} \leq p_Q(y) \leq 1.$$

Dividing by $p_Q(y)$ gives the conditional-expectation representation in the main text. At positive regularization, $\sum_y C_Q(y)$ is generally less than one. Normalising it would add a Q-dependent partition term and define a different estimator, which would not recover the stated complete IQ criterion. We optimize the displayed attenuated score and predict with the normalized behavioural policy π_Q.

C Exposure convention and deterministic subdivision

Exposure determines how the reward penalty scales with the physical size of an edge. This appendix explains that convention and proves that splitting a deterministic edge does not change the optimized score under the stated freedom to choose the inserted values.

Fix a positive edge exposure ℓ_e, in declared physical units, and use $\phi_{\lambda,e}(r) = \ell_e \phi_\lambda(r/\ell_e)$. It has all the monotonicity, concavity and domination properties used below. Its penalty vanishes at zero reward; we add no path-length constant.

The latent extension is sensitive to score conventions. Replacing ϕ_e by $\phi_e - c$ subtracts $cT(\tau)$ from a complete route score. That change is independent of Q on a fixed complete dataset. But on an incomplete record it reweights possible routes by e^{-cT}, and therefore changes the model's completion and objective when lengths differ. Complete-data equivalence does not select a unique latent extension.

Now replace a single action $s \to t$ by a chain with no new choices. Keep its first Q-value, all original-state Q-values, the physical recording law and all original origins/destinations fixed. Let internal Q-values be free finite tabular entries. Since internal states have one outgoing action, their values equal those action Q-values. The chain rewards telescope to the original reward r, while their allocation can be any finite $(r_1, \ldots, r_k)$ summing to r. For positive lengths summing to ℓ, Jensen's inequality gives

$$\sum_j \ell_j \phi_\lambda(r_j/\ell_j) \leq \ell \phi_\lambda(r/\ell).$$

Equality is attained by $r_j = \ell_j r/\ell$. Choose internal Q-values backwards from $V_Q(t)$ to realise those rewards. The behavioural path probabilities are unchanged, since all inserted transitions are deterministic. Every occurrence of this edge in every route receives the same optimal score. Thus the upper bound holds simultaneously for all route scores and is attained by one internal allocation; summing compatible paths and then profiling yields precisely the coarse objective. Repeated traversals do not change this conclusion. Shared neural constraints, new observations or origins at inserted states, or restrictive internal Q boxes can invalidate this profiling argument and are excluded from the statement.

Thus exposure weighting makes deterministic subdivision consistent when the inserted values are free. The remaining proofs apply separately to each edge's concave function ϕ_e.

D VARIATIONAL LEARNING

This appendix justifies the completion-and-learning step in Section 3. The variational identity shows when improving the frozen objective improves the record score; the subsequent arguments verify integrability on the countable route space and concavity of the tabular or affine learning block.

Here $H(q) = -\mathbb{E}_q \log q$ is Shannon entropy, $q_{Q,i} = q_Q(\cdot \mid y_i)$, and Q_0 is the current parameter vector. For fixed completion distributions q_i with finite displayed terms, write $\mathcal{L}(Q; q) = \sum_i \mathbb{E}_{q_i}[S_Q(\tau) + \log \kappa(y_i \mid \tau)] + \sum_i H(q_i)$. The variational identity gives

$$F(Q) - F(Q_0) = \mathcal{L}(Q; q) - \mathcal{L}(Q_0; q) + \sum_i \{\mathrm{KL}(q_i \| q_{Q,i}) - \mathrm{KL}(q_i \| q_{Q_0,i})\}. \tag{14}$$

For $q = q_{Q_0}$, the old-completion KL is zero and the new one is nonnegative. Increasing the frozen objective therefore increases F. This is the usual variational argument (Dempster et al., 1977; Neal & Hinton, 1998) applied to the regularized score, not to the behavioral likelihood.

Why the identity holds. Define $q_Q(\tau \mid y) = \kappa(y \mid \tau) e^{S_Q(\tau)} / C_Q(y)$. For a distribution q supported on compatible routes for which the displayed expectations and entropy are finite,

$$\mathrm{KL}(q \| q_Q) = -H(q) - \mathbb{E}_q\{\log \kappa(y \mid \tau) + S_Q(\tau)\} + \log C_Q(y).$$

Hence $\log C_Q(y) = \mathcal{L}_y(Q; q) + \mathrm{KL}(q \| q_Q)$. Choosing $q = q_{Q_0}$, subtracting the identity at Q_0, and summing observations gives the exact improvement identity. This is the usual variational minorization operation, with the CoRoute score in place of a likelihood (Dempster et al., 1977). The old completion must be the optimizer for this score; replacing it by a different distribution loses equality at Q_0.

Why cycles do not invalidate the calculation. The expectations above remain finite on the countable route space. For fixed Q, all edge scores are bounded in magnitude by a finite constant, so $|S_Q(\tau)| \le c_1 + c_2 T$. Also $q_Q \le p_Q / C_Q(y)$, implying finite expected length. Finally $t|\log t| \le 1/e$ on $[0, 1]$ implies $\mathbb{E}_{q_Q}|\log \kappa| \le 1/(eC_Q(y))$. The required entropies and expectations are therefore finite. Local differentiation in finite Q is justified by the same length bound and locally bounded edge derivatives.

What is concave, and what is not. For a frozen completion, expected counts are nonnegative constants. The map $Q \mapsto V_Q(s)$ is convex and $Q \mapsto r_Q(s, a) = Q(s, a) - V_Q(s')$ is concave. Composition with concave, nondecreasing ϕ remains concave. Thus

$$\sum_e \bar{n}_e \phi(r_Q(e)) - \sum_o \bar{\eta}_o V_Q(o)$$

is concave in tabular Q, and in parameters of an affine Q map. A convex box constraint can make the update domain compact without changing the argument. Concavity does not extend through a general neural parameterization, nor does it imply concavity of the full log-sum score. Monotone accepted updates give increasing objective values; global convergence of the full latent-variable estimator is not asserted.

E AN ANALYTIC COUNTEREXAMPLE TO THE ORDINARY-POSTERIOR UPDATE

The following two-route example isolates why completion must match the chosen training score. We construct a small parameter change that increases the surrogate based on the behavioral posterior while decreasing the regularized record objective, so the distinction is consequential even without cycles or a large graph.

Consider two routes $o \to a \to d$ and $o \to b \to d$, both compatible with the recorded endpoints. Let $\lambda = 1$, $Q(o, a) = Q(o, b) = Q(b, d) = 0$, and $Q(a, d) = \log 4$. Vary only $Q(o, a) = t$, holding the other values fixed. The two path scores are

$$S_a(t) = \frac{7}{4} - 4e^{-t} - \log(1 + e^t), \qquad S_b(t) = -\log(1 + e^t).$$

Thus

$$F(t) = \log\{1 + \exp(7/4 - 4e^{-t})\} - \log(1 + e^t), \qquad F'(0) = \frac{4}{1 + e^{9/4}} - \frac{1}{2} < 0.$$

The derivative is negative because $e^{9/4} > 7$.

At $t = 0$, the ordinary behavioural posterior assigns probability $1/2$ to each route. Its frozen surrogate is $\widetilde{\mathcal{L}}(t) = \frac{1}{2}(7/4 - 4e^{-t}) - \log(1 + e^t)$, up to constants, and $\widetilde{\mathcal{L}}'(0) = 3/2 > 0$. By continuity, every sufficiently small positive step increases that surrogate and decreases the intended score. The correct old completion instead gives mass $1/(1 + e^{9/4})$ to route a. The example uses an unobserved choice; its score change is regularization-driven, not new behavioural information.

F Proof of the completion certificate

This appendix supplies the computable record-gain bound used by CoRoute's acceptance test. We first define the numerical quantities and state the certificate, then prove it on the recording product graph and construct the error bounds needed to account for repeated visits.

F.1 Certificate and computational quantities

The exact completion requires a linear solve on the recording product graph. A local residual alone can be misleading because a cycle repeatedly visits the same error. We therefore bound the change in the whole record score.

For $e = (s, a)$, the score factors over decisions through

$$B_Q(s, t) = \sum_{a:T(s,a)=t} \pi_Q(a \mid s) e^{-\psi_e(r_Q(e))}. \tag{15}$$

For a channel with a finite recording realization, a state also records how many visible anchors have been matched, through a finite recording automaton that gives each compatible path its prescribed weight. Keep only states reachable from the record start that can also reach acceptance. Then let $A \geq 0$ be the transient kernel and $b \geq 0$ the terminal weights of one record, with start state o and terminal backward value one. Here $\rho(A)$ denotes the spectral radius. If $\rho(A) < 1$ and every state reaches a positive terminal weight, then

$$h = Ah + b > 0, \qquad C_Q(y) = h_o. \tag{16}$$

The recording construction and proper path model ensure these conditions for the kernels considered here. All sums below keep terminal transitions.

Here completion paths retain action labels and recording alignments, and q_Q denotes their joint posterior. Its physical-route marginal is (5). The KL identity below concerns these product paths; data processing gives the same upper bound for physical-route marginals. The recording transitions are specified in the proof below.

A path law from an approximate backward value. Given any $\widehat{h} > 0$, define $d_s = (A\widehat{h} + b)_s$ and

$$\widetilde{P}_{st} = \frac{A_{st}\widehat{h}_t}{d_s}, \qquad \widetilde{p}_{s\dagger} = \frac{b_s}{d_s}, \qquad a_s = \log \frac{d_s}{\widehat{h}_s}. \tag{17}$$

These rows sum to one and give a proper path law $\widetilde{q}$ with exact expected visit vector $\nu = e_o + \widetilde{P}^\top \nu$, where e_o is the unit vector at the start state and $\dagger$ denotes termination. Row normalization is the finite absorbing version of the positive-value transform for conditioned Markov processes (Chetrite & Touchette, 2015); what we need here is its error.

Theorem F.1 (A residual certificate for completion and learning)**.** *Let $U > 0$ satisfy $U \geq AU + b$. For the proposal in* (17)*,*

$$\mathrm{KL}(\widetilde{q} \| q_Q) = \log \frac{h_o}{\widehat{h}_o} - \nu^\top a \leq \log \frac{U_o}{\widehat{h}_o} - \nu^\top a. \tag{18}$$

For an approximate visit vector $\widehat{\nu}$, put $r_\nu = e_o + \widetilde{P}^\top \widehat{\nu} - \widehat{\nu}$. Choose $v > 0$ and $\alpha < 1$ with $\widetilde{P}v \le \alpha v$, and define

$$R_\nu(g) = \frac{\max_s |g_s|/v_s}{1-\alpha} \sum_s v_s |r_{\nu,s}|. \tag{19}$$

Consider any new parameters Q^+ with the same recording support. Let g_s be the proposal expectation of the one-step log ratio of the new and old score kernels, including the terminal transition. Then

$$\boxed{\log C_{Q^+}(y) - \log C_Q(y) \ge \log \frac{\widehat{h}_o}{U_o} + \widehat{\nu}^\top (a+g) - R_\nu(a+g).} \tag{20}$$

The bound adds over records. All quantities on its right are available without an exact backward value or an exact visit solve.

The bound is a conservative estimate of the improvement, not an estimate of the optimal parameters. It separates the quantities needed for a decision. The start-value ratio controls the backward-value error, $\widehat{\nu}^\top(a+g)$ estimates an accumulated path quantity, and $R_\nu(a+g)$ accounts for the visit error. The proof telescopes the path likelihood ratio and bounds the remaining expected sum with the weighted drift inequality, counting repeated visits on cycles. Summing the bound over records gives the acceptance test; if it is negative, the solves can be refined. The test uses only finite score values and therefore also applies to a neural Q with the same support.

Candidate vectors U and v are produced by positive iterations and accepted only after their componentwise inequalities are verified, with the solver's error intervals included in the acceptance margin. The explicit construction is given below.

Proposition F.2 (Small local residuals can hide large path error)**.** *There is a one-state absorbing family with $\|A\widehat{h} + b - \widehat{h}\|_\infty \to 0$ but $\mathrm{KL}(\widetilde{q}\|q_Q) \to \infty$.*

Repeated visits can therefore invalidate a fixed local residual tolerance. The proof below gives an inverse-Q example for every fixed positive λ; the weighted certificate accounts for this accumulation.

F.2 Proof and recording construction

The proof compares path probabilities under the exact and approximate completion laws. A telescoping identity isolates backward-value error, and a bound on accumulated visits controls the remaining expectation. We also specify how recording alignments enter the construction and give a one-state example showing why a local residual tolerance is insufficient.

All statements in this section concern one fixed record and its labelled product paths. Recording-alignment weights $\kappa_y(\tau,\alpha)$ satisfy $\sum_\alpha \kappa_y(\tau,\alpha) = \kappa(y \mid \tau)$, so summing $\kappa_y(\tau,\alpha)e^{S_Q(\tau)}$ gives the same $C_Q(y)$. The joint posterior has the physical-route marginal in (5). The matrices A, b sum labelled transition weights; when labels share a successor, the proposal splits their probabilities in proportion to those weights. In path products below, each factor retains its label. Write $A_{s\dagger} = b_s$ and $\widehat{h}_\dagger = h_\dagger = 1$. Every transient state can reach the terminal state, and row normalization preserves every positive edge. The resulting finite Markov chain is therefore proper. Its stopping time has a geometric tail, so expected visits and path-log expectations are finite.

A visit-level recording realization. Let $\rho_t \in [0,1]$ be the probability of recording a visit to state t. For interior anchors $y_1, \dots, y_m$, let (s,j) mean that j anchors have been recorded. For a physical action $e : s \to t$ with score weight $B_e = \pi_Q(e \mid s)e^{-\psi_e(r_Q(e))}$ and $t \ne d$, include

$$(s,j) \to (t,j) :\ B_e(1-\rho_t), \qquad (s,j) \to (t,j+1) :\ B_e\rho_t\, \mathbf{1}\{t = y_{j+1}\} \quad (j < m).$$

A transition to d is accepted only at $j = m$, with weight B_e; the origin and destination are forced observations. Missing a visit to the next anchor does not advance j, so hidden revisits and repeated anchors are included. Distinct choices of detected visits are distinct alignments and their weights add. Other finite-state channels add their recording state to this product. The computational certificate uses such finite realizations; the objective (5) also allows general discrete channels.

The telescoping identity. For any labelled terminal path $(e_0, \dots, e_{T-1})$ with states $(s_0 = o, \dots, s_T = \dagger)$, write $\widetilde{p}_e$ for its labelled transition probability. Then

$$\frac{\prod_{j<T} A_{e_j}}{\prod_{j<T} \widetilde{p}_{e_j}} = \widehat{h}_o \prod_{j<T} \frac{d_{s_j}}{\widehat{h}_{s_j}} = \widehat{h}_o \exp\left(\sum_{j<T} a_{s_j} \right).$$

The exact joint completion is the numerator divided by h_o. Taking the proposal expectation of the log ratio proves (18). For a supersolution U, iteration of $U \geq AU + b$ yields $U \geq \sum_{j=0}^{m-1} A^j b + A^m U$. Since $\rho(A) < 1$, taking $m \to \infty$ gives $U \geq h$. Marginalizing recording alignments gives $\mathrm{KL}(\widetilde{q}^{\mathrm{route}} \| q_Q^{\mathrm{route}}) \leq \mathrm{KL}(\widetilde{q} \| q_Q)$. The difference is the nonnegative conditional alignment KL. For example, (o, a, a, d) can produce the record (a) by detecting either visit; the two weights $\rho_a(1 - \rho_a)$ must both contribute.

The visit residual. Subtracting the exact visit equation from the residual equation gives $\nu - \widehat{\nu} = (I - \widetilde{P}^\top)^{-1} r_\nu$. For any finite vector g, put $c_g = \max_s |g_s|/v_s$. Positivity and $\widetilde{P} v \leq \alpha v$ imply

$$|(I - \widetilde{P})^{-1} g| \leq \sum_{j \geq 0} \widetilde{P}^j |g| \leq \frac{c_g}{1 - \alpha} v.$$

Consequently $|(\nu - \widehat{\nu})^\top g| \leq R_\nu(g)$. This holds for any finite $\widehat{\nu}$; nonnegativity can be imposed by the solver but is not needed for the residual identity.

A candidate learning update. Let A^+ include the terminal weights at Q^+. For each positive old labelled transition define $j_e = \log(A_e^+ / A_e)$ and $g_s = \sum_{e : s_e = s} \widetilde{p}_e j_e$, including terminal transitions. The variational inequality applied to the fixed proposal gives

$$\log C_{Q^+}(y) \geq \mathbb{E}_{\widetilde{q}} \log \frac{\prod_{e \in \tau} A_e^+}{\widetilde{q}(\tau)} = \log \widehat{h}_o + \nu^\top (a + g).$$

Subtract $\log C_Q(y) \leq \log U_o$ and use the visit-residual bound on the single vector $a + g$. This proves (20). Using one vector is at least as tight as separately bounding the old completion KL and the surrogate change, by the triangle inequality for R_ν. For exact $\widehat{h} = h$, $U = h$ and $\widehat{\nu} = \nu$, it reduces to the usual frozen-completion improvement.

An explicit upper solution and a drift vector. Choose $w > 0$ with $d_w = (I - A)w > 0$, and let

$$r_h = A\widehat{h} + b - \widehat{h}, \qquad t = \max\left\{ 0, \max_s \frac{r_{h,s}}{d_{w,s}} \right\}, \qquad U = \widehat{h} + tw.$$

Such a drift vector exists because $(I - A)^{-1}\mathbf{1} > 0$; positive partial sums also work once their drift is verified positive. This choice makes $(I - A)(\widehat{h} + tw) - b \geq 0$ componentwise. Positive $\widehat{h}, w$ give $U > 0$. Since the proposal is proper, $v = (I - \widetilde{P})^{-1}\mathbf{1} \geq \mathbf{1}$ is finite and satisfies $\widetilde{P} v = v - \mathbf{1}$. Taking $\alpha = \max_s(1 - 1/v_s) < 1$ proves existence of a drift certificate. Computation needs only a vector satisfying the verified inequality, not this exact inverse. If evaluation of any inequality is inexact, its certified interval is used in the conservative direction.

Why a local residual tolerance is insufficient. To prove Proposition F.2, we use a single state whose loop becomes increasingly hard to leave. This already shows the obstruction at positive regularization. Fix any $\lambda > 0$, use unit exposures and an OD-only record with $\kappa(y \mid \tau) = 1$. On one transient state with a loop and an exit, set $Q_{\mathrm{exit}} = 0$ and $Q_{\mathrm{loop}} = \log((1-\eta)/\eta)$. Then $V = -\log \eta$, $r_{\mathrm{loop}} = \log(1 - \eta)$ and $r_{\mathrm{exit}} = 0$. The actual score kernel is

$$A_\lambda = \exp\left(\frac{1 - (1 - \eta)^{-\lambda}}{\lambda} \right), \qquad b = \eta, \qquad h = \frac{\eta}{1 - A_\lambda}.$$

It is proper since $0 < A_\lambda < 1 - \eta$. With $\widehat{h} = \eta$, the residual is ηA_λ and the proposal exits with probability $1/(1 + A_\lambda)$. Hence

$$\mathrm{KL}(\widetilde{q} \| q_Q) = -\log(1 - A_\lambda) - (1 + A_\lambda)\log(1 + A_\lambda) = \log(1/\eta) - 2\log 2 + o(1),$$

because $A_\lambda = 1 - \eta - (\lambda/2)\eta^2 + O(\eta^3)$. Thus $h \to 1$ and the residual vanishes while the KL diverges at fixed positive regularization. Every member has finite Q and positive path support.

G PROOF OF THE FLOW AND PATH CERTIFICATE

This appendix proves that fitting the frozen tabular objective is equivalent to optimizing a unique positive flow. The resulting gap identity provides a computable error bound, which we then translate into guarantees for complete paths and recorded observations. The final argument justifies the balanced count projection used in numerical proposals.

Fix one absorbing destination, with no outgoing actions, and a nonempty finite action graph in which every nonterminal state can reach it. For finite tabular Q-values, define

$$V_Q(s) = \log \sum_{e:s\to\cdot} e^{Q(e)}, \quad V_Q(d) = 0, \qquad \pi_Q(e \mid s) = e^{Q(e)-V_Q(s)}, \quad r_Q(e) = Q(e) - V_Q(t_e).$$

Here s_e, t_e are the tail and head of an action. Let D be the edge-by-state incidence matrix, with the destination coordinate deleted, so $D^\top f$ is net outflow. Suppose completed edge counts $c_e > 0$ obey $D^\top c = b \geq 0$, where $b = b_0$ is the origin-mass vector from the main text. Fix exposures $\ell_e > 0$, $\lambda > 0$, and write $k_e = \ell_e/\lambda$. For scalar arguments define $D_{\mathrm{g}}(u\|v) = u\log(u/v) - u + v$, continuously at $u = 0$. Let $f_s = \sum_{e:s\to\cdot} f_e$, and set

$$L(Q) = \sum_e c_e k_e (1 - e^{-r_Q(e)/k_e}) - b^\top V_Q,$$
$$J(f) = \sum_e k_e D_{\mathrm{g}}(f_e\|c_e) + \sum_e f_e \log(f_e/f_{s_e}).$$

Zero-flow state entropy contributes zero.

Theorem G.1 (Unique flow update and path certificate)**.** *For unrestricted finite tabular Q-values,*

$$\boxed{\max_Q L(Q) = \min_{f\geq 0:\, D^\top f = b} J(f).} \tag{21}$$

Both optimizers are unique and attained; $f^* > 0$, and $\pi_{Q^*}(e \mid s) = f^*_e/f^*_s$. For every feasible f and finite Q,

$$J(f) - L(Q) = \sum_e k_e D_{\mathrm{g}}\Big(f_e \Big\| c_e e^{-r_Q(e)/k_e}\Big) + \sum_s f_s \,\mathrm{KL}(\pi_f(\cdot \mid s)\|\pi_Q(\cdot \mid s)). \tag{22}$$

For positive total origin mass $B_0 = \sum_s b_s$, this gap also bounds complete-path KL as in (8)*, with origins drawn from b/B_0.*

This formula specializes IQ-Learn's occupancy duality to an absorbing network with exposure weights (Garg et al., 2021). General flow/potential duality is established route-choice methodology (Fosgerau & Sørensen, 2026). The constructive update and its certificate are the role of the theorem here.

Existence and uniqueness of the flow. The feasible set is nonempty because it contains c, and is closed and convex. The weighted generalized KL is strictly convex. Conditional negative entropy is convex as a sum of perspectives of negative Shannon entropy. Let $M = \sum_e f_e$, $C = \sum_e c_e$, $k_{\min} = \min k_e$, and $A_{\max}$ be the largest action count. Nonnegativity of scalar KL and the log-sum inequality give

$$J(f) \geq k_{\min}\{M\log(M/C) - M + C\} - M\log A_{\max} \longrightarrow \infty.$$

Thus sublevel sets are compact, and continuity gives an attained minimizer; strict convexity makes it unique.

It is positive. Otherwise use the feasible perturbation $f(t) = (1-t)f^* + tc$. Each zero edge contributes $k_e c_e t\log t + O(t)$ to the KL change, with positive coefficient. At a state of positive original outflow, any zero-action entropy terms add further nonpositive $t\log t$ terms. At a zero-outflow state the entropy block is homogeneous and changes by $O(t)$. Positive coordinates also change by $O(t)$. The total change is strictly negative for small $t > 0$, contradicting optimality.

Recover a finite inverse-Q optimizer. Destination reachability makes D full column rank: $Dh = 0$, with $h_d = 0$, forces every potential to vanish along a path to the destination. The first-order conditions at the positive optimum therefore supply a unique potential vector v such that

$$k_e \log(f_e^*/c_e) + \log(f_e^*/f_{s_e}^*) + (Dv)_e = 0.$$

Set $\pi_e^* = f_e^*/f_{s_e}^*$ and $Q^*(e) = \log \pi_e^* + v_{s_e}$. Its soft values are v_s, and

$$r_{Q^*}(e) = -k_e \log(f_e^*/c_e), \qquad c_e e^{-r_{Q^*}(e)/k_e} = f_e^*.$$

All quantities are finite.

Exact duality and uniqueness of Q. Writing $g_e = c_e e^{-r_Q(e)/k_e}$, direct scalar expansion gives

$$k_e D_{\mathrm{g}}(f_e \| c_e) + f_e r_Q(e) - c_e k_e (1 - e^{-r_Q(e)/k_e}) = k_e D_{\mathrm{g}}(f_e \| g_e).$$

Also $r_Q = \log \pi_Q + DV_Q$, so feasible flow implies $\sum_e f_e r_Q(e) = \sum_e f_e \log \pi_Q(e) + b^\top V_Q$. Summation proves (22). Both terms are nonnegative, and both vanish at the constructed (f^*, Q^*), proving equality and attainment without an unjustified exchange of extrema.

Any other maximizing Q must also make both certificate terms vanish at $f = f^*$. Positivity gives the same rewards and policy as Q^*. Equal policies imply Q differences are state potentials; equal rewards make their incidence differences zero. Full rank of D then gives $Q = Q^*$. Finally, π^* has full support and is proper. Flow balance becomes $f_S^* = b + (P_T^*)^\top f_S^*$, whose unique solution is its actual expected visit flow. This includes cyclic returns.

Algorithmic and boundary consequences. Any feasible flow and finite Q bracket the optimum: $L(Q) \le L(Q^*) = J(f^*) \le J(f)$. The certificate is a block-optimization error, not a global certificate for the alternating latent-variable learner. It remains a comparison with the unrestricted block when a neural class is used; it does not make that restricted block equivalent to the flow program. Feasibility must be established independently of a small numerical residual.

For one junction, unit exposure and counts summing to one, $\pi_a^* \propto c_a^{1/(1+\lambda)}$, an immediate check of the constants. Positive counts are essential: with two terminal actions and $c = (1, 0)$, the supremum is zero, approached by $Q_1 = 0, Q_2 \to -\infty$, and no finite Q attains it. A positive reference flow can repair support only by changing the fitted criterion; arbitrary edgewise pseudocounts need not preserve balance.

From the optimal flow to full paths. Let $\Delta = J(f) - L(Q)$ for a feasible comparator f. The key observation is that a path log-probability ratio is a sum of action log-probability ratios. Expected edge counts therefore convert the optimization gap into a bound on complete paths. For any common recording channel, with output laws M^*, M_Q,

$$\mathrm{KL}(M^* \| M_Q) + \mathbb{E}_{Y \sim M^*} \mathrm{KL}(P^*(\cdot \mid Y) \| P_Q(\cdot \mid Y)) \le \frac{\Delta}{B_0}. \tag{23}$$

With origins drawn from b/B_0, the policy of f^* has expected edge counts f^*/B_0. All paths stop at their first goal arrival. Hence

$$B_0 \,\mathrm{KL}(P^* \| P_Q) = \sum_e f_e^* \log \frac{\pi_{Q^*}(e \mid s_e)}{\pi_Q(e \mid s_e)}.$$

Apply the gap identity at f^* and use $J_c(f^*) \le J_c(f)$ for any feasible comparator flow. This proves (8). Generate Y from the same channel conditionally on each complete path. The channel term cancels in the joint KL, and the chain rule in the other order gives

$$\mathrm{KL}(P^* \| P_Q) = \mathrm{KL}(M^* \| M_Q) + \mathbb{E}_{M^*} \mathrm{KL}(P^*(\cdot \mid Y) \| P_Q(\cdot \mid Y)).$$

This also covers a channel that records the origin. All terms are finite: policies have positive support, log ratios per edge are bounded on the finite graph, and the reference stopping time has finite expectation. For a bounded path statistic $a(\tau) \in [0, 1]$, Pinsker's inequality gives

$$|\mathbb{E}_{P^*} a - \mathbb{E}_{P_Q} a| \le \sqrt{\Delta/(2B_0)}.$$

For a positive feasible flow f, define $\pi_f(e \mid s) = f_e/f_s$ and let P_f be its complete-path law. The following two-sided bound therefore checks both the reconstructed flow policy and the fitted Q policy. For flow vectors that have zero entries, the gap uses continuous entropy values; the path argument is applied only to the positive optimal flow.

The two-sided certificate. At the positive constrained optimum f^*, the first-order term of $J_c(f) - J_c(f^*)$ vanishes along every feasible difference. Its Bregman remainder is

$$J_c(f) - J_c(f^*) = \sum_e k_e D_{\mathrm{g}}(f_e \| f_e^*) + B_0 \,\mathrm{KL}(P_f \| P^*).$$

For $f > 0$, flow conservation and properness justify its path law exactly as for f^*. Add this identity to the dual gap at f^* and discard the two nonnegative edge-divergence terms to obtain the two-sided certificate

$$\mathrm{KL}(P_f \| P^*) + \mathrm{KL}(P^* \| P_Q) \le \frac{\Delta}{B_0}. \tag{24}$$

Positive balanced count projection. In (41), the reference $q_e = \widehat{c}_e + \epsilon c_e^{\mathrm{ref}}$ is strictly positive by the declared choice of ϵ. The feasible set is closed and contains $c^{\mathrm{ref}} > 0$. The generalized KL objective is continuous, strictly convex and coercive, so it has a unique minimizer. If any coordinate of that minimizer were zero, a small feasible move toward c^{ref} would produce a negative $t|\log t|$ term on its zero coordinates, whereas the other changes are $O(t)$. This contradicts optimality. Thus the projection is positive and balanced. It defines a numerical proposal criterion; the original-record acceptance certificate is applied afterward.

H Proof of the joint-change reuse certificate

This appendix proves the main reuse theorem by expressing the computable potential bound as the actual reuse loss plus nonnegative divergence terms. It then characterizes zero-loss reuse, separates changes in the fitting problem from error in the old cache, and derives the sharp mean-exposure bound for cyclic modules.

H.1 What the bound measures and when reuse is exact

This subsection records the throughput reparameterization and the additional conclusions of the reuse theorem. The expansion relates the direct loss definition in the main text to the internal objective used in the proof; the optimized potential bound identifies exact reuse.

Fix positive reference counts c_e, exposures ℓ_e, and $\lambda > 0$, and retain $k_e = \ell_e/\lambda$. To separate the internal route choice from total throughput, define the internal cost $K(w)$ and mean exposure $L(w)$:

$$K(w) = \sum_e k_e w_e \log \frac{w_e}{c_e} + \sum_e w_e \log \frac{w_e}{w_{s_e}}, \qquad L(w) = \sum_e \ell_e w_e. \tag{25}$$

Expanding $J_c(zw)$ shows that the internal optimum depends on throughput through the scalar $u = (\log z - 1)/\lambda$:

$$w_u = \arg\min_{w \in \mathcal{W}} \{K(w) + uL(w)\}. \tag{26}$$

Thus higher throughput places more weight on mean exposure in the internal fitting objective. This dependence follows from our exposure-weighted fitting criterion; it is not a general property of policy evaluation. The optimizer w_u is positive and unique. Normalizing outgoing flow gives its internal policy and path law P_u. If $n(\tau) = (n_e(\tau))_e$ is the vector of edge traversal counts, then $\mathbb{E}_{P_u} n = w_u$ and path exposure is $L_\tau = \sum_e \ell_e n_e(\tau)$. The full network objective also retains the entry-choice entropy $z \log(z/M_s)$, where M_s is total outgoing flow at the entry, including competing external actions. This term is unchanged when comparing internal policies at the same throughput.

Write K_c when the reference counts need to be explicit. At new counts c' and throughput z, the optimum w_1 minimizes $K_{c'}(w) + uL(w)$. Expanding the two terms in (9) gives

$$\mathcal{E} = z\{K_{c'}(w_0) + uL(w_0) - K_{c'}(w_1) - uL(w_1)\}.$$

Define $\mathcal{B}(\theta; w_0) = \min_h \Psi(h; \theta, w_0)$. Here the minimum is over finite state potentials with $h(t) = 0$. It is attained uniquely, and the stronger form of the main bound is

$$\frac{\mathcal{E}}{z} + \mathrm{KL}(P_1 \| P_0) \le \mathcal{B}(\theta; w_0) \le \Psi(h; \theta, w_0).$$
$$\mathcal{B}(\theta; w_0) = 0 \iff w_0 = w_1 \iff (k_e \theta_e)_e \in \operatorname{range} D. \tag{27}$$

If w_0 is the exact old optimum at counts c and throughput z_0, the certificate can equivalently use the joint shift

$$\zeta_e = \log \frac{z c_e}{z_0 c'_e}, \qquad \mathcal{B}(\theta; w_0) = \mathcal{B}(\zeta; w_0). \tag{28}$$

Here $\operatorname{range} D = \{Dh : h(t) = 0\}$ is the set of state-potential differences.

The zero-loss condition says that the new objective's gradient at the cached flow is a state-potential difference. Its inner product with any feasible flow change is zero, so the cached flow is already optimal. For an exact old fit, $\zeta_e = \log(z/z_0) - \log(c'_e/c_e)$ makes the cancellation under proportional changes explicit. Inexact caches must use θ unless their error is separately accounted for. For two parallel terminal actions with unit exposure and $\lambda = 1$, swapping counts $(0.8, 0.2)$ to $(0.2, 0.8)$ changes the fitted probabilities from $(2/3, 1/3)$ to $(1/3, 2/3)$ at fixed throughput.

H.2 Proof: loss, potential bound, and cyclic consequence

We first express reuse loss through edge and path divergences, then derive the potential bound and its exact zero set. The first two steps establish the bound used by the algorithm; the remaining steps characterize its zero set and fixed-count specialization. A repeated-loop family establishes sharpness.

Fix the module, exposures and λ, and put $H_1(w) = K_{c'}(w) + uL(w)$. The Hessian of K_c is independent of c: changing reference counts adds a linear term. A positive feasible flow induces its proper internal policy and has the expected edge counts w, as proved below for cyclic modules. Therefore its entropy Bregman remainder is a complete-path KL. Here a Bregman remainder means the difference between a function and its first-order approximation: $D_{K_c}(w, v) = K_c(w) - K_c(v) - \nabla K_c(v)^\top (w - v)$. For any positive feasible w, v,

$$D_{K_c}(w, v) = \sum_e k_e D_{\mathrm{g}}(w_e \| v_e) + \mathrm{KL}(P_w \| P_v). \tag{29}$$

This identity holds with c' as well as c.

Step 1: express reuse loss as divergences. Differentiation gives $\nabla H_1(w_0)_e = k_e \log(z w_{0,e}/c'_e) + \log \pi_0(e \mid s_e) = k_e \theta_e$. Expand H_1 at w_0 and use its minimizer w_1:

$$\frac{\mathcal{E}}{z} = -D_{K_c}(w_1, w_0) - \sum_e k_e \theta_e (w_{1,e} - w_{0,e}). \tag{30}$$

The first-order term at w_1 vanishes on feasible differences, so an expansion at that point also gives

$$\frac{\mathcal{E}}{z} = \sum_e k_e D_{\mathrm{g}}(w_{0,e} \| w_{1,e}) + \mathrm{KL}(P_0 \| P_1).$$

This proves the forward-KL part of (11).

Step 2: obtain the computable potential bound. For a finite potential h define $\widetilde{w}_{h,e} = w_{0,e} \exp\{-\theta_e - (Dh)_e/k_e\}$. The scalar weighted-KL conjugate, together with $D^\top(w_1 - w_0) = 0$, gives the exact identity

$$\boxed{\Psi(h; \theta, w_0) - \frac{\mathcal{E}}{z} = \sum_e k_e D_{\mathrm{g}}(w_{1,e} \| \widetilde{w}_{h,e}) + \mathrm{KL}(P_1 \| P_0).} \tag{31}$$

One can verify it by expanding each scalar divergence: the potential term sums to $h^\top D^\top (w_1 - w_0) = 0$, and the remaining linear term is exactly (30). Dropping the nonnegative edge terms proves the reverse-KL bound for every finite h. This is the policy-preserving specialization of the flow-gap identity. On the isolated module with origin mass z, choose $Q_h(e) = \log \pi_0(e \mid s_e) + h(s_e)$, so $V_{Q_h} = h$ and $r_{Q_h}(e) = \log \pi_0(e \mid s_e) + (Dh)_e$. Substitution into the gap at flow $z w_0$ gives exactly $z\Psi(h; \theta, w_0)$; the policy-divergence term of that gap is zero. The certificate uses the KL conjugate, while the joint shift and its sharp scalar consequence specify how it controls cached inverse learning.

Step 3: characterize the optimized bound and exact reuse. Consider the strictly convex problem

$$\min_{w\in\mathcal{W}}\left\{\sum_e k_e D_{\mathrm{g}}(w_e\|w_{0,e})+\sum_e k_e\theta_e(w_e-w_{0,e})\right\}.$$

Its objective is coercive, and $\mathcal{W}$ is closed with a positive member. The minimizer exists and is positive by the same $t\log t$ boundary argument used for the flow theorem. Its equality-constrained optimality equations give a finite potential h such that the minimizer is $\widetilde{w}_h$. Thus $D^\top\widetilde{w}_h = D^\top w_0$, which is precisely $\nabla_h\Psi = 0$. Moreover

$$\nabla_h^2\Psi = D^\top \operatorname{diag}(\widetilde{w}_{h,e}/k_e)D > 0,$$

since the exit-normalized incidence matrix has full column rank. The potential minimum is attained uniquely. Every scalar $\mathfrak{e}(x)$ is nonnegative and vanishes only at zero, so its minimum is zero exactly when $k\theta + Dh = 0$ for some h. This is the constrained first-order condition for w_0 to minimize H_1. Strict convexity gives (27).

Separating count change from the old optimization residual. For an exact old optimum at (c, z_0), put $H_0(w) = K_c(w) + ((\log z_0 - 1)/\lambda)L(w)$. The difference of the new and old gradients at w_0 is

$$\nabla H_1(w_0) - \nabla H_0(w_0) = (k_e\log(zc_e/(z_0c'_e)))_e = k\zeta.$$

The old gradient is a state-potential difference. Hence $k(\theta-\zeta)\in\operatorname{range}D$, and translating the potential variable leaves the minimum of Ψ unchanged. This proves (28). For an inexact cache, the directly computed θ already includes the old optimization residual and needs no such simplification.

The cyclic mean-exposure bound. If $c' = c$, every $\zeta_e = a = \log(z/z_0)$. Evaluate the shifted certificate at zero potential:

$$z\Psi(0;\zeta,w_0) = \frac{zL(w_0)}{\lambda}(e^{-a}-1+a) = \frac{L(w_0)}{\lambda}\{z\log(z/z_0) - z + z_0\}.$$

Equation (11) proves (12). The expected exposure is finite for a positive feasible unit flow, even though the set of complete walks can be infinite.

Sharpness from a repeated loop. Use three states s, v, t, with edges $s\to v$, $v\to v$ and $v\to t$, each of exposure one. Fix $\lambda > 0$, let $z_0 = 1$, and take reference counts $c = (1, M(M/(M+1))^\lambda, 1)$ for $M > 0$. They are positive and balanced. The optimal unit flow at z_0 is $w_0 = (1, M, 1)$: the derivative with respect to loop flow is zero exactly when $\log(M/c_{\mathrm{loop}})/\lambda + \log(M/(M+1)) = 0$. At a fixed new ratio $z = r \neq 1$, the optimal loop count m is the unique positive solution of

$$rm\left(\frac{m}{m+1}\right)^\lambda = M\left(\frac{M}{M+1}\right)^\lambda.$$

As $M\to\infty$, $m = M/r + O(1)$. The path laws have geometric numbers of loop traversals, and

$$\mathrm{KL}(P_M\|P_m) = \log\frac{m+1}{M+1} + M\log\frac{M(m+1)}{m(M+1)} = O(1).$$

The exact reuse loss is therefore

$$\mathcal{E} = \frac{rM}{\lambda}(\log r - 1 + 1/r) + O(1).$$

The bound in (12) is the same leading term with M replaced by $M+2$. Their ratio tends to one for every fixed $\lambda > 0$ and $r \neq 1$. The potential certificate can cancel both non-loop edge shifts, leaving $rM\mathfrak{e}(\log r)/\lambda$; its ratio also tends to one. This proves the claimed uniform sharpness. For each finite M, the mean exposure $L(w_0) = M+2$ is finite, but the maximum complete-walk exposure is infinite. At $r\neq 1$ the corresponding path-cost change $(\log r)L_\tau/\lambda$ is unbounded. The finite-horizon comparison of Schlaginhaufen & Kamgarpour (2026, Lemma D.3) and the bounded-cost bound of Eckstein & Nutz (2022, Proposition 3.12) therefore do not apply directly to this family, while (12) remains finite and asymptotically sharp. Their more general Lemma 4.5 allows unbounded costs under a growth condition but uses both optimizers; our bound uses the cached flow.

I When throughput changes the policy

This appendix explains how the optimal internal policy responds to throughput when reference counts stay fixed. It is a refinement of the reuse analysis, not an extra condition for the main certificate. The practical conclusion is simple: exposure differences make a policy sensitive to throughput; equal exposure makes reuse exact. The response formula quantifies this sensitivity and supports the DAG bound in Appendix J.

I.1 Statement and interpretation

With counts fixed, the optimal policy is reusable at every throughput exactly when all complete walks have the same exposure. The theorem below also quantifies the response when exposures differ, including the contribution of repeated visits on cycles.

Let $n(\tau)$ be the vector of edge traversal counts, let $\ell_{\min} = \min_e \ell_e$, and write $\Sigma_u = \mathrm{Cov}_{P_u}(n)$ for the covariance matrix of edge counts and $A_u = \mathrm{diag}(\ell_e/(\lambda w_{u,e}))$ for the indicated diagonal matrix. The covariance records how route choices change edge usage together; A_u is the additional curvature from regularization. With I the identity matrix, define

$$\chi_u = \ell^\top \Sigma_u (I + A_u \Sigma_u)^{-1} \ell. \tag{32}$$

Theorem I.1 (Exact response and local-to-global rigidity)**.** *For every $z > 0$, including modules with cycles, $d\,\mathbb{E}_{P_u} L_\tau / d\log z = -\chi_u/\lambda$ and $0 \le \chi_u \le \mathrm{Var}_{P_u}(L_\tau)/(1 + \ell_{\min}/\lambda)$. The following are equivalent: (i) $\chi_u = 0$ at one finite u; (ii) every finite s–t walk has the same total exposure; (iii) the internal optimal policy is independent of z for all $z > 0$. A module containing a directed cycle has $\chi_u > 0$ at every finite u.*

The statement is local and its conclusion is global: an exactly zero response at one operating point implies that no flow change can alter the internal policy, and a nonzero response measures how much higher throughput favors lower-exposure paths. The covariance counts repeated edge visits, so the same formula describes loops. It arises because the path-entropy Hessian and the path-count covariance are inverses on the feasible flow directions (Fosgerau & Sørensen, 2026; Wainwright & Jordan, 2008); the edge penalty adds A_u and produces the factor $(I + A_u\Sigma_u)^{-1}$.

I.2 Proof

We differentiate the module's optimality conditions along feasible flow directions. Relating the curvature of path entropy to edge-count covariance then yields the response formula and its variance bound. Finally, the covariance nullspace identifies the condition for exact reuse.

Let D be the module's edge-by-state incidence matrix with the exit coordinate removed. Every state is reachable from the entry and can reach the exit. Thus D has full column rank, and $\mathcal{W} = \{w \ge 0 : D^\top w = e_s\}$ has a strictly positive member: average finitely many entry-to-exit walks covering every edge. Fix an orthonormal basis U for $\mathcal{T} = \ker D^\top$.

The constrained sensitivity formula. Differentiating the optimality equations of (26) on the feasible directions gives

$$\frac{dw_u}{du} = -U(U^\top H U)^{-1} U^\top \ell, \tag{33}$$

where H is the Hessian of K at w_u. This is the constrained form of exponential-family mean/parameter duality (Wainwright & Jordan, 2008; Fosgerau & Sørensen, 2026) for absorbing walk counts; (32) follows once the edge penalty contributes A_u.

Existence, positivity, and smoothness on cyclic modules. Existence, uniqueness and positivity follow by the same argument as Appendix G: the edge penalty is strictly convex and grows faster than the linear throughput term, even on an unbounded cyclic flow set. Moving toward a positive feasible flow excludes a boundary minimum. It remains to justify differentiating the optimum.

Write $T(w) = \sum_e w_e \log(w_e/w_{s_e})$. Its Hessian is

$$H_T = \mathrm{diag}(1/w_e) - R\,\mathrm{diag}(1/w_s) R^\top, \tag{34}$$

where $R_{es} = 1$ if edge e leaves state s, and zero otherwise. The Hessian of K is $H = A_u + H_T$, with $A_u > 0$. Thus $U^\top HU$ is positive definite. The implicit function theorem applied to the constrained first-order conditions makes w_u smooth in u and gives (33).

A positive unit flow is the occupancy of its own policy. Normalize w by outgoing sums. The resulting policy has full support and is proper by goal reachability. Flow conservation reads $w_S = e_s + P_T^\top w_S$, whose unique solution is the expected visit vector. Its expected edge counts are therefore w. This argument holds in a neighborhood of every positive w_u, including feasible tangent perturbations. Geometric tails justify the differentiations below.

Why covariance appears. This step links a change in feasible flow to a change in path probabilities. It supplies the covariance form of the response; the sparse implementation can use the constrained Hessian directly. Take a tangent direction $v \in \mathcal{T}$, write $v_s = \sum_{e:s_e=s} v_e$, and differentiate the log path probability:

$$Z_v(\tau) = \sum_e n_e(\tau) \left(\frac{v_e}{w_e} - \frac{v_{s_e}}{w_{s_e}} \right) = n(\tau)^\top H_T v.$$

The derivative of a normalized path law has mean zero. Differentiating its mean edge counts gives

$$\Sigma H_T v = v \quad (v \in \mathcal{T}). \tag{35}$$

Moreover, $D^\top n(\tau) = e_s$ pathwise, so $\Sigma D = 0$ and the range of Σ is contained in $\mathcal{T}$.

To identify its nullspace exactly, suppose $a^\top \Sigma a = 0$. Every finite legal terminal walk has positive probability, so $a^\top n$ is constant over those walks. At each vertex v, fix a prefix from s to v. Concatenating it with any two v–t suffixes shows that the a-sum on a suffix is independent of the suffix. Define that sum as $h(v)$ and put $h(t) = 0$. Then $a_e = h(s_e) - h(t_e)$ on every edge, including cycle edges. Conversely, such a potential has a constant sum along every terminal walk. Therefore

$$\ker \Sigma = \operatorname{range} D, \qquad \operatorname{range} \Sigma = \mathcal{T}. \tag{36}$$

It follows from (35) that H_T and Σ are inverse positive forms on $\mathcal{T}$.

Response formula. Differentiate the optimality equations and multiply by Σ. Since potential multipliers vanish under Σ, the derivative obeys

$$(I + \Sigma A_u) \frac{dw_u}{du} = -\Sigma \ell.$$

The matrix $I + \Sigma A_u$ is invertible: it is similar to $I + A_u^{1/2} \Sigma A_u^{1/2} > 0$. The response matrix is

$$(I + \Sigma A_u)^{-1} \Sigma = \Sigma (I + A_u \Sigma)^{-1} = U (U^\top H U)^{-1} U^\top,$$

which is symmetric positive semidefinite. Taking the inner product with ℓ and using $du/d \log z = 1/\lambda$ proves the exact response.

Variance damping. Since $\operatorname{diag}(1/w_e) - H_T$ is positive semidefinite, $A_u \succeq (\ell_{\min}/\lambda) \operatorname{diag}(1/w_e) \succeq (\ell_{\min}/\lambda) H_T$. On $\mathcal{T}$ this gives $H \succeq (1 + \ell_{\min}/\lambda) H_T$. Invert the positive forms and use (35) to get $\chi_u \le \ell^\top \Sigma \ell / (1 + \ell_{\min}/\lambda)$. The numerator is exactly $\operatorname{Var}(L_\tau)$.

Local-to-global rigidity. The positive constrained inverse shows $\chi_u = 0$ if and only if $U^\top \ell = 0$, independently of u. By (36), this is equivalent to constant exposure on all complete walks. In that case $L(w)$ is constant throughout $\mathcal{W}$, so the optimizer of $K + uL$ does not depend on u. Conversely, a policy independent of u has the same unit occupancy at each u, and differentiation gives $\chi_u = 0$. A directed cycle can be repeated in a terminal walk and has strictly positive exposure; it therefore rules out constant complete-walk exposure. This proves every part of Theorem I.1.

J FIXED-COUNT REUSE BOUNDS AND EXACT REDUCTION

This appendix develops the fixed-count consequences of the response law. It gives the sharp exposure-range bound for acyclic modules, derives exact reuse-loss identities, and shows when a module can be replaced by one effective edge without changing the frozen fitting problem.

Interpretation: exact and approximate reuse. With counts fixed, a module's optimal policy can be reused at every throughput if and only if all complete entry-to-exit walks have the same exposure (Theorem I.1). In that case, changing throughput adds the same cost to every possible walk, so their relative preferences do not change. The module can then be represented by one effective edge in the frozen flow problem, while retaining the entry-choice entropy and original origin constraints.

Positive-exposure cycles rule out this exact invariance: repeating a cycle changes total exposure. They do not rule out accurate approximate reuse, which the general certificate can still establish. For a directed acyclic graph (DAG), the shortest and longest path exposures give an additional sharp bound proportional to $z \log^2(z/z_0)$ (equation (37)). The range bound below is an alternative to the general certificate when the module is acyclic and its exposure range is easy to compute.

J.1 Range bound and exact reduction

For an acyclic module, shortest and longest path exposures provide a finite range that bounds the cost of reuse. We state that bound first and describe its zero-range case, where a single effective edge represents the module exactly.

For fixed counts c, put $C_0 = \sum_e k_e c_e$. The minimized internal contribution at throughput z is

$$G(z) = C_0 + z\{K(w_u) + uL(w_u)\}, \qquad u = (\log z - 1)/\lambda.$$

The response law also gives a quantitative rule for approximate reuse. On a DAG, let ΔL be the largest minus smallest complete-path exposure and $\ell_{\min} = \min_e \ell_e$. Define $G_{\text{reuse}}(z) = C_0 + z\{K(w_{u_0}) + uL(w_{u_0})\}$, where $u_0 = (\log z_0 - 1)/\lambda$. Reusing w_{u_0} fitted at z_0 costs at most

$$\boxed{0 \le G_{\text{reuse}}(z) - G(z) \le \frac{z(\Delta L)^2}{8\lambda^2(1 + \ell_{\min}/\lambda)} \log^2 \frac{z}{z_0},} \tag{37}$$

with the coefficient $1/8$ uniformly sharp; ΔL costs two backward passes and no path enumeration. When all path exposures agree, the whole module is represented exactly at every throughput by one weighted-KL edge with reference $c_{\text{eff}} = \exp(-\lambda K_*/L)$, where L is that common exposure and $K_* = \min_{w \in \mathcal{W}} K(w)$. This supplies an exact reduced proposal for a frozen flow block. The derivation below retains the additive constant, entry-choice entropy and original origin constraints, and establishes the loss identities and sharpness.

J.2 Loss identities and proof

Two expressions for the same reuse loss connect the optimization problem to the stated bound: one uses edge and path divergences, and the other integrates the throughput response. We use these identities to prove the range bound, exhibit a sharp family, and establish the exact single-edge reduction.

Set $\delta = \log(z/z_0)/\lambda$. The exact reuse loss satisfies

$$\frac{G_{\text{reuse}}(z) - G(z)}{z} = \sum_e \frac{\ell_e}{\lambda} D_{\text{g}}(w_{u_0,e} \| w_{u,e}) + \text{KL}(P_{u_0} \| P_u) \tag{38}$$

$$= \delta^2 \int_0^1 (1 - v) \chi_{u_0 + v\delta}\, dv. \tag{39}$$

When every module path has the same exposure L, set $K_* = \min_{w \in \mathcal{W}} K(w)$ and $c_{\text{eff}} = \exp(-\lambda K_*/L)$; then

$$G(z) = \frac{L}{\lambda} D_{\text{g}}(z \| c_{\text{eff}}) + C_0 - \frac{L}{\lambda} c_{\text{eff}}, \tag{40}$$

so on a DAG one weighted-KL edge represents G at every throughput if and only if all path exposures agree. The effective reference c_{eff} is used with the original origin constraints and entry-choice entropy; after the reduced block is solved, the saved unit flow times the macro-edge flow recovers fine-edge counts. This reduces the frozen learning problem only; inference keeps its recording states. It complements occupancy-based reduction for recursive-logit estimation (Mai & Ta, 2026).

The edge-and-path decomposition in (29) already gives the Bregman remainder of K; no additional path identity is needed here. First-order optimality at w_u makes $\langle \nabla K(w_u) + u\ell, w_{u_0} - w_u \rangle = 0$. Subtracting the two module objectives now yields (38).

Let $\Phi(u) = \min_{w \in \mathcal{W}} \{K(w) + uL(w)\}$. The envelope theorem and the response formula give $\Phi'(u) = L(w_u)$ and $\Phi''(u) = -\chi_u$. Taylor's formula with integral remainder, valid for either sign of $\delta = u - u_0$, gives

$$\Phi(u_0) + \delta\Phi'(u_0) - \Phi(u) = \delta^2 \int_0^1 (1 - v)\chi_{u_0 + v\delta}\, dv.$$

This proves (39). All quantities are finite on each compact u interval; positivity and smoothness have already been proved. For a DAG, the exposure variable lies in an interval of width ΔL, so its variance is at most $(\Delta L)^2/4$. Insert the damped variance bound into the integral and integrate $1 - v$ to obtain (37).

Sharpness at every positive damping level. Fix any $m > 0$, set $\lambda = 1$, and use two parallel edges with exposures m and $m + \epsilon$. Choose positive reference counts so that the unit optimum at $u_0 = 0$ is $(1/2, 1/2)$; for instance $c_e = \frac{1}{2}\exp(1 - \log 2/\ell_e)$ makes the two unconstrained gradients zero there. The corresponding throughput is $z_0 = e$. Writing $w = (x, 1 - x)$, the second derivative of K at $x = 1/2$ is $4(1 + m) + 2\epsilon$, while $dL/dx = -\epsilon$. Hence

$$\chi_{u_0} = \frac{\epsilon^2}{4(1 + m) + 2\epsilon}.$$

Let $z = z_0 e^{\delta}$ and first take $\delta \to 0$. The ratio of the actual reuse loss to the right side of (37) tends to $4(1 + m)/(4(1 + m) + 2\epsilon)$, which tends to one as $\epsilon \downarrow 0$. The two paths have exposure range ϵ and $\ell_{\min}/\lambda = m$ throughout. Thus no coefficient smaller than $1/8$ works uniformly even at a fixed positive damping level.

Single-edge closure on a DAG. Every feasible unit flow on a DAG is a convex combination of path flows. If every path has exposure L, then $L(w) = L$ throughout $\mathcal{W}$, and direct substitution gives (40). For necessity, suppose $G(z) = (L_{\mathrm{eff}}/\lambda) D_{\mathrm{g}}(z \| c_{\mathrm{eff}}) + B$ for every $z > 0$. Compactness of the DAG flow polytope implies $G(z) \to C_0$ as $z \downarrow 0$. Subtract C_0, divide by z, and write $u = (\log z - 1)/\lambda$. Then $\Phi(u)$ is affine on the entire real line. Each affine function $K(w) + uL(w)$ lies above Φ for every real u. Their slopes must agree, so every path has exposure L_{eff}. The effective count is a KL reference, and the origin vector is retained from the fine graph. Expansion of the entry entropy gives

$$\sum_{e : s_e = s} z w_e \log \frac{z w_e}{M_s} = z \log \frac{z}{M_s} + z \sum_{e : s_e = s} w_e \log w_e,$$

which verifies that outside competition is preserved.

K Implementation of the certified iteration

This appendix connects the theoretical certificates to the numerical iteration. It distinguishes the path laws being compared, specifies the implemented acceptance margin and balanced count proposals, and explains how to evaluate response and manage module caches. These details define the reference procedure tested in Section 6.

Path laws used by the certificates. The behavioral law p_Q generates complete physical trajectories. Its score-induced completion $q_Q(\cdot \mid y)$ weights trajectories by both the recording channel and the regularized score. In the finite recording product, this law is lifted to include action labels and recording alignments; $\widetilde{q}$ is the approximation induced by $\widehat{h}$. The laws P^*, P_Q in Theorem G.1 are behavioral laws for a frozen block, including its origin distribution. The laws P_0, P_1 in Theorem 4.1 are internal module laws at unit throughput. Thus completion KL, fitted-policy KL and module KL refer to different laws.

The implemented acceptance margin. Each record kernel is a finite matrix of transition weights whose path sum computes that record's score. The stored kernel contains the floating-point coefficients actually used by the solver. Outward-rounded arithmetic encloses calculation errors and certifies a lower gain for the score $\bar{F}$ of those stored coefficients. Certifying the ideal score F additionally requires bounds on errors made when constructing the coefficients, as described below. Our experiments audit that construction, count projection, flow optimization and Q reconstruction numerically; they do not provide an arithmetic certificate for the entire construction pipeline.

Let $\bar{F}$ be the log-partition score of the stored finite, proper record systems. A directed lower bound b certifies $\bar{F}(Q^+) - \bar{F}(Q) \geq b$. If kernel-construction enclosures also give $|F(Q) - \bar{F}(Q)| \leq \epsilon_Q$ and $|F(Q^+) - \bar{F}(Q^+)| \leq \epsilon_{Q^+}$, then

$$F(Q^+) - F(Q) \geq b - \epsilon_Q - \epsilon_{Q^+}.$$

This follows by adding the two construction errors to the stored-system inequality. Our directed experiments report b for the stored systems; construction and flow feasibility/reconstruction are audited separately, rather than assigned uncomputed ϵ bounds. The earlier float64-margin tests remain numerical validations. A gap for projected counts concerns that projected block, while acceptance always uses the original record kernels.

Balanced numerical proposals. For nonnegative numerical counts $\widehat{c} \geq 0$, one explicit proposal rule is

$$\overline{c} = \arg \min_{c \geq 0 : D^\top c = b_0} \sum_e D_{\mathrm{g}}(c_e \| \widehat{c}_e + \epsilon c_e^{\mathrm{ref}}). \tag{41}$$

Here $c^{\mathrm{ref}} > 0$ is a fixed feasible flow for the same origin mass. We take $\epsilon = 0$ when $\widehat{c} > 0$; a declared $\epsilon > 0$ handles zero numerical counts. The KL projection is unique and positive on a graph admitting c^{ref}. Its block certificate refers exactly to $\overline{c}$. The proposed Q^+ is then tested on the original record objective by (20). This procedure preserves the origin constraints and gives a defined proposal at finite precision.

Computing the response without a dense covariance. The covariance formula is an interpretation, not a requirement to store an edge-by-edge matrix. Equation (33) can be evaluated by the sparse saddle system

$$\begin{pmatrix} H & D \\ D^\top & 0 \end{pmatrix} \begin{pmatrix} v \\ \xi \end{pmatrix} = \begin{pmatrix} -\ell \\ 0 \end{pmatrix}, \qquad \chi = -\ell^\top v.$$

Here H is a positive diagonal matrix minus one rank-one block per state. A Hessian-vector product needs edge operations and outgoing sums, so its cost is linear in the number of edges. This provides an independent way to measure response while retaining the path-covariance meaning. The covariance itself is computed only on small modules in the numerical identity tests.

Reconstructing the policy and interpreting budgets. For a positive candidate flow, normalize outgoing flows to obtain π_f, then choose state values V and set $Q^+(e) = \log \pi_f(e \mid s_e) + V(s_e)$, with goal value zero. Choosing these values affects the reward and fitting gap while preserving π_f. The full gap is evaluated after this reconstruction; the record test uses the original records.

A local budget $\varepsilon_{\mathrm{reuse}}$ measures excess module objective at the candidate throughput. Dividing it by z gives a per-entry loss budget and, for a passing certificate, a bound on the sum of module path KLs. The global tolerance Δ / B_0 instead measures full-path KL per episode. Loss budgets add for disjoint modules at a fixed outer flow, but do not cover errors in that outer flow or in value reconstruction. Thus local checks do not replace the global check. Budgets are chosen accuracy requirements; a failed bound calls for refinement or refitting, and does not itself prove that the candidate is inaccurate.

Choosing modules and error budgets. We first check the two-terminal boundary condition and exclude internal origins. A DAG traversal computes its shortest and longest exposure. Exact equality of the exposure potential differences certifies replacement, while a nonzero range supplies the bound in (37). Reference internal policies are recomputed when their certified reuse budget is exceeded. Disjoint modules contribute additive frozen objective errors at a fixed macro flow. The final flow gap is evaluated after reconstruction, including entry competition, rather than inferred from the number of removed edges. The cache stores the graph, exposures, λ, reference counts, throughput

and feasible internal flow. A change in graph or exposures or λ starts a new cache. Count and throughput changes are tested by θ in (10); an exact reference fit can instead use ζ. This includes changes between completion steps. Evaluating $\Psi(0)$ takes one edge pass, and any finite potential update supplies another valid certificate. Cyclic modules use (11) or its mean-exposure corollary; DAGs can also use their exposure range. In floating-point arithmetic, a small response is a diagnostic. Exact contraction uses verified potential equalities; an interval upper bound on the exposure range is used for approximate contraction.

K.1 Scope and future directions

The present work establishes when cached policies can be reused with a quantified error, and how that decision can be checked within a learning iteration. This focus separates the validity of reuse from its eventual computational benefit. The following directions build on that foundation.

Broader models and module structures. The reuse guarantees assume finite deterministic absorbing MDPs, positive feasible flows, and two-terminal modules without internal origins. The exact flow optimizer is an unrestricted tabular reference; the record acceptance test also accommodates finite neural or shared-feature proposals with common support. Extending the reuse analysis to stochastic dynamics, multi-boundary modules, and constrained function classes would broaden its applicability. The certificates concern the specified regularized objective and frozen fitting blocks, not identification of a unique true reward or global convergence of the alternating learner.

From certified reuse to computational savings. The controlled studies show lower acceptance costs and additional valid cache retention, but potential refinement does not consistently outperform warm full solving, and full solving is fastest on the small integrated fixture. These results motivate cost-aware choices of modules, refinement budgets, and fallback rules. A useful next test is whether such choices reduce total learning time at matched accuracy on larger workloads, including all completion, reconstruction, and certification costs.

Large-network learning and predictive evaluation. The MOBIS-derived network supplies a real-world setting, while the reported pilot validates 19 small modules from 542 records. The large sparse-solver benchmarks assess a computational component rather than end-to-end learning on the full network. Training on larger portions of that network, evaluating held-out route predictions, and comparing with alternative IL/IRL methods would test practical generalization and scalability beyond the current numerical validation. Predictive superiority is not established here.

End-to-end numerical certification. The implemented outward-rounded acceptance tests certify the scores of stored record kernels. Kernel construction, flow feasibility, and policy reconstruction are audited numerically. Propagating verified error bounds through these stages, as outlined in Appendix K, would extend the arithmetic guarantee to the ideal learning objective.

L Extended related work

This appendix places the paper's results alongside the methods most directly related to its construction. We first discuss incomplete-data inverse learning, then compare the reuse certificate with flow sensitivity, policy-reuse bounds, entropic stability and network reduction. The comparisons identify both shared tools and the scope of the present guarantees.

Inverse learning and incomplete demonstrations. Inverse reinforcement learning recovers a reward from behavior (Ng & Russell, 2000); the maximum-entropy formulation (Ziebart et al., 2008) makes the demonstration law a Gibbs measure over paths and is the ancestor of the route models above. Adversarial imitation replaces that likelihood with a discriminator (Ho & Ermon, 2016; Fu et al., 2018), and IQ-Learn removes the inner solve by parameterizing the soft action value and reading the reward off the inverse Bellman map (Garg et al., 2021). Partly observed demonstrations are usually handled by marginalizing the hidden decisions (Kitani et al., 2012; Bogert & Doshi, 2014; Mai et al., 2019; 2023) — all under the *behavioral* law. For our specified regularized inverse-Q score, that posterior need not give an ascent step. We use the completion consistent with this score and

certify the resulting update. Variational completion follows EM and its free-energy view (Dempster et al., 1977; Neal & Hinton, 1998); the positive-value path transform is related to conditioned Markov processes (Chetrite & Touchette, 2015).

Flow sensitivity, policy reuse, and reduction. Fosgerau & Sørensen (2026, Theorem 7, equation (16)) derive constrained inverse-Hessian sensitivity for separable link perturbations; our learning cost (7) also carries state conditional entropy, and Theorem 4.1 controls instead the finite loss and two-way path KL from reusing an old policy after its completed counts change, sharply in the cyclic case. Covariance/entropy duality (Wainwright & Jordan, 2008) supplies the convex tools. Two reductions are adjacent but distinct: nonlinear Kron elimination keeps a nonlinear boundary relation (van der Schaft et al., 2024), and occupancy reduction drops low-use edges with a controlled excluded-route error (Mai & Ta, 2026). Ours asks when one *stored internal policy* stays valid and otherwise bounds the cost of using it, keeping the module's action support.

Finite cost changes also admit two-way KL bounds in entropic optimal transport: Eckstein & Nutz (2022, Lemma 4.5) pair the cost change with the difference of the two optimizers. In finite-horizon inverse learning, Schlaginhaufen & Kamgarpour (2026, Corollary A.11 and Lemma D.3) identify policy reuse loss with trajectory KL. Their equations (162)–(163) already give upper bounds using the old Hessian, a known parameter shift and a bounded cumulative-advantage scale, without computing the new optimizer. Avoiding that optimizer alone is therefore not our distinction. We work with positive feasible cached cyclic flows, including inexact caches, after joint count and throughput changes. Corollary 5.1 needs only the cached mean exposure and remains sharp when maximum complete-path exposure is infinite.

With an exact old cache, the count and throughput change adds the linear flow cost $\langle k\zeta, w\rangle$. When $k\zeta = Dh$, flow conservation makes this term constant: it has the boundary cancellation familiar from potential-based reward shaping (Ng et al., 1999; Skalse et al., 2023). Equation (11) controls the finite reuse loss and both path KLs away from this invariant set.

M Network construction and additional checks

This appendix describes the aggregate network used to construct the pilot module problems and collects additional numerical checks. It separates validation of the score, response and reduction calculations from the data-processing choices used to build the cell-transition graph.

Checks not tabulated in the main text. The equal-exposure reduction is exact to 2.8×10^{-13} over 2,922 cases; all 4,096 count-only changes have strictly positive reuse loss at unchanged throughput; the zero-response control stays at 3.7×10^{-32}; and the completion engine agrees with the compiled `index_add` reference to 4.4×10^{-16}.

Refinement and acceptance. With 4, 16, 64 and 256 positive iterations the constructive certificate accepts 0, 428, 4,096 and 4,096 of the same improving proposals: the bound is conservative when the completion is coarse and becomes tight well before the solve is exact, which is what makes an approximate E-step usable rather than merely cheap.

MOBIS-Covid19. 268,013 car journeys of 5–20 km from 2,339 participants inside a Zürich bounding box, 69 M ordered GPS vertices. Vertices are collapsed to 110 m cells and the support graph keeps the directed 8-neighbour transitions observed by at least two participants, giving 56,121 cells and 189,294 directed edges. Of the 228,707 usable records only 7.5% (17,064) are strict 8-neighbour walks from origin to destination; the rest have a median of six gaps each, and a gap spans a median Chebyshev distance of two cells. The recording model is a visit-level Bernoulli channel with cell-dependent probabilities, calibrated per fold on non-evaluation travellers only. These constructions supply the aggregate network and the pilot module problems used in the numerical validation.

N Experimental protocol and reproducibility

This appendix gives the workloads, solver settings, reference calculations and timing protocols behind Section 6. The studies progress from checks of individual identities to controlled learning, integrated

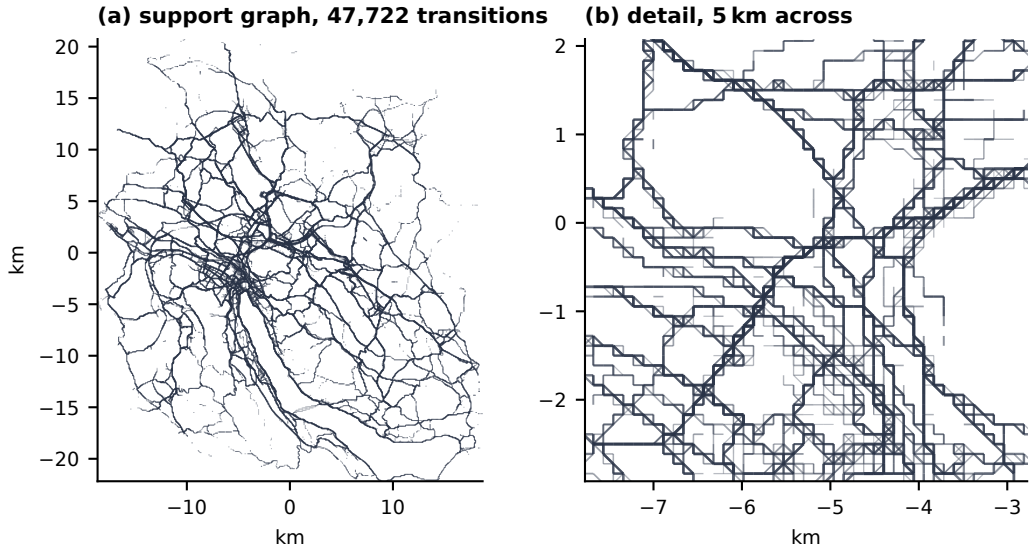

Figure 3: The network the MOBIS study is specified on: 110 m cells joined by the 8-neighbour transitions the archive contains, with **(b)** a detail at a scale where the staircase geometry shows why we call this a cell-transition graph and not a map-matched road network. Only transitions carrying ≥ 20 distinct participants are drawn, axes are kilometres from the bounding-box centre. This is the aggregate network, not a predictive benchmark result.

Table 1: Executed theory checks; maxima over all draws in each family. Relative errors divide by one plus the magnitude of the reference.

Check	DAG	Cyclic	Equal exposure
Covariance response error	3.46×10^{-15}	4.67×10^{-15}	1.14×10^{-15}
Finite-difference error	4.65×10^{-10}	5.69×10^{-9}	2.78×10^{-13}
Reuse-gap identity error	6.40×10^{-15}	1.60×10^{-14}	3.62×10^{-15}
Largest response χ	1.3996	50.6587	3.70×10^{-32}
Joint changes: 20,480 cases in five families			
Remainder identity error			4.71×10^{-14}
Largest realized loss / bound in sharp families			
DAG range bound (37)			0.9995
Cyclic potential bound (11)			0.99999975
Cyclic mean bound (12)			0.99999775

reuse and generic module queries. For each study, we specify what is measured and distinguish numerical audits from arithmetic certification of the stored record kernels.

The theory checks use CUDA float64 on an NVIDIA RTX 4080 SUPER, PyTorch 2.11.0+cu128 and Triton 3.6.0. Dense linear solves provide separate small-graph references; path enumeration is used only on finite DAGs.

N.1 Checks of the score and certificate identities

These small-graph studies compare the formulas with direct enumeration, independent solves, or finite differences. They establish numerical agreement for the stated workloads; timing comparisons begin in the next subsection.

Behavioral-posterior update test. All 16,384 configurations use the same graph on vertices $0, \ldots, 6$, with every edge $i \to j$ for $i < j$, unit exposures, origin zero and goal six. Its 21 edges

Table 2: Sparse linear algebra on physical node-state graphs, before link-state or recording-product expansion: float64, median of five repeats. SpMM timings exclude channel construction and witnesses. Iteration counts use the graphs' own attenuation at maximum row sum 0.999–0.9999.

		SpMM ms per 1,024 RHS			iterations, $\varepsilon = 10^{-8}$		
Graph	states × actions	Triton	torch	`index_add`	Richardson	BiCGSTAB	peak GB
G900	900 × 3,480	0.15	0.21	0.45	90	18	0.17
P3077	3,077 × 7,459	0.24	0.17	0.54	120	21	0.55
G10k	10,000 × 39,600	0.45	0.84	1.79	90	19	0.97
G90k	90,000 × 358,800	3.68	7.42	–	90	18	2.60

give 32 complete paths. Initial Q entries are independent $\mathcal{N}(0, 1.25^2)$. A record retains each path independently with probability 0.35, always retaining the direct path; its recording weight is one for retained paths and zero otherwise. Six values $\lambda \in \{0, 0.02, 0.1, 0.3, 0.6, 1\}$ are cycled over sixteen batches of 1,024 draws. At frozen ordinary completion weights, compute the exact enumerated surrogate gradient g and take $Q^+ = Q + 0.001g/\max(10^{-12}, \|g\|_2)$, with no optimizer stopping test. The 343 counted cases have surrogate gain above 10^{-8} and original score gain below -10^{-8}. An independent $\mathcal{N}(0, 0.2^2)$ Q perturbation also checks the variational identity in each draw.

Sparse completion checks. The 3,308 record comparisons use 48 random graphs crossed with three $(\lambda, \rho_{\min})$ settings: $(0, 0.05)$, $(0.35, 0.35)$ and $(1, 0.7)$. Graph sizes cycle from 6 to 12 states; DAG and cyclic graphs alternate, and each consecutive pair alternates single and parallel actions. Each allowed non-self edge is included with probability 0.7 on a DAG (forward edges only), or 0.55 on a cyclic graph. Empty rows receive a forward edge; graphs are redrawn unless all states reach the goal. Parallel-action cases duplicate each edge independently with probability 0.25. Exposures are uniform on $[0.5, 1.5]$ and Q entries are $\mathcal{N}(0, 1.2^2)$. Detection probabilities are uniform on $[\rho_{\min}, \rho_{\min} + 0.2]$. For each setting, six policy walks are sampled from origin zero, stopping at the goal or 30 vertices; only completed walks contribute records. Records retain the first zero to three interior visits after removing origin and goal visits. Sparse Richardson and BiCGSTAB solves use tolerance 10^{-11} and are compared with dense resolvents at 10^{-9}, including support agreement and isolated-state padding. This is a numerical engine check on fixed generated workloads.

Constructive response tests. Each of the three response families has 2,048 draws. The DAG has edges $(01, 02, 12, 13, 23, 24, 34)$; the cyclic graph has $(01, 02, 12, 21, 13, 23, 31, 34, 14, 24)$, with origin zero and goal four. Policy logits are independent $\mathcal{N}(0, 1.2^2)$ variables, normalized within rows. Solve for unit expected counts w. Draw exposures uniformly on $[0.5, 2]$ and λ uniformly on $[0.2, 2]$; the equal-exposure control uses the DAG with $\ell_{ij} = j - i$. Choose $\log c_e = \log w_e + 1 + \log \pi_e / k_e$. The first-order conditions then hold exactly at $u = 0$. Finite-difference responses use independently solved optima at $u = \pm 0.0001$. Separate draws use $u \in [-0.5, 0.5]$ to compare the reuse gap with its edge-KL and path-KL decomposition. For covariance, use the fundamental matrix $N = (I - P_T)^{-1}$ and

$$\mathbb{E}[n_e n_f] = \mathbf{1}_{e=f} w_e + w_e N_{t_e, s_f} \pi_f + w_f N_{t_f, s_e} \pi_e,$$

where a row at the absorbing exit is zero. Subtract $w_e w_f$ to obtain Σ. This calculation retains repeated visits on cycles. Equality-control exposures are differences of a decreasing state potential on a DAG. We record all constraint residuals and do not discard difficult draws on the basis of their response or error.

Joint-change families. We generate 4,096 cases each for a five-state, seven-action DAG with joint changes and a five-state, ten-action cyclic graph with joint changes, count-only changes, potential-invariant changes, or an inexact cache. Policy logits are independent $\mathcal{N}(0, 1.1^2)$ variables; $\ell_e \sim \mathrm{Unif}[0.5, 2]$ and $\lambda \sim \mathrm{Unif}[0.2, 2]$. Reference counts are chosen as $\log c_e = \log w_e + 1 + \log \pi_e / k_e$, so w is optimal at $u = 0$ and $z_0 = e$. Independently, $a = \log(z/z_0)$ is uniform on $[-0.7, 0.7]$ and $\log(c'_e/c_e) \sim \mathcal{N}(0, 0.6^2)$. Count-only cases set $a = 0$. Invariant cases instead use $\log(c'_e/c_e) = a + (Dh)_e/k_e$ with independent $h_s \sim \mathcal{N}(0, 0.4^2)$, making $\zeta = -Dh/k$. Inexact caches mix $0.8w$ with 0.2 times an independent unit flow whose policy logits have standard deviation 0.8, and use the resulting residual θ. The identities are checked per unit throughput, using $\mathcal{E}/z$, with no discarded

draws. These are positive module reference counts for Theorem 4.1; no recording-data interpretation is needed for this algebraic test.

Finite sharpness instances. The largest DAG ratio in Table 1 uses the two-edge family of Appendix J, with $m = 10$, $\epsilon = 0.01$, $\lambda = 1$ and $\delta = \log(z/z_0) = 0.03$. The loss and bound per unit throughput are $1.0222626058 \times 10^{-9}$ and $1.0227272727 \times 10^{-9}$. The repeated-loop family of Appendix H, at $M = 10^6$, $\lambda = 0.25$ and $z/z_0 = 4$, has loss per unit throughput 2545176.8082; the potential and mean-exposure bounds are 2545177.4445 and 2545182.5348. An independent 80-digit scalar root solve confirms the displayed ratios 0.9995, 0.99999975 and 0.99999775; its stationarity residuals are below 10^{-60}. These finite checks accompany the analytic limiting proofs.

N.2 Acceptance checks with shared learning proposals

These experiments isolate the cost of deciding whether to accept an update. All acceptance rules receive the same proposals, so differences in runtime do not come from following different learning trajectories.

Learning with certified acceptance. Each test runs 40 gradient updates on 256 cyclic models with eight transient states and a terminal action at every state. Records specify the initial state and absorption. Every transient state has actions to all eight transient states (including itself) and to the goal. Initial Q entries are independent $\mathcal{N}(0, 0.15^2)$ variables. The regular case uses $\lambda = 0.125$; the weak absorption case uses $\lambda = 0.0005$ and subtracts 5 from the initial terminal logits. Shared proposals use 64 backward and visit iterations in the first case and 16,384 in the second. In every timed loop, form the frozen-score gradient G from these approximate completion counts and propose $Q^+ = Q + 0.5G/\max(1, \|G\|_\infty)$. The norm is taken separately for each model.

We compare three acceptance rules. The tight-score control caches the previous score and warm-starts the new solve, with relative value error below 10^{-9} and an acceptance margin of 2×10^{-8}. A second control adaptively compares direct score intervals from residual and row-drift bounds. The residual rule evaluates Theorem F.1 using the E-step estimates and checks the upper solution and drift inequalities with float64 margins. These experiments numerically validate the rule at the stated audit tolerances. All three rules accept 10,240 proposals in each case and return identical final Q. Independent dense CUDA solves audit every accepted change after timing, finding no decrease above 10^{-9}. A further 3,072 checks include decreasing proposals and corrupted visit estimates, with no bound violation or false acceptance.

The corruption controls draw fresh initial Q entries as 0.15 times independent standard normals, with the same terminal shifts and iteration budgets as above. Both signs of the shared gradient step $\Delta Q = 0.5G/\max(1, \|G\|_\infty)$ are tested, where G is the frozen-score gradient. Visit estimates are the computed vector, zero, or independent $\mathcal{N}(0, 100^2)$ entries, including negative entries. The $2 \times 2 \times 3 \times 256$ checks use a 10^{-8} bound-violation tolerance and count a false acceptance when the bound is nonnegative but the independent score change is below -10^{-9}.

Table 3 reports fixed-point sweeps and synchronized wall time for the complete loop. Extra sweeps are those used by acceptance; total sweeps also include the shared proposals. Residual checks are not counted as fixed-point sweeps. Each time reports the median and full range of three warmed runs. The tight control uses the fastest tested check interval: 64 from $\{16, 64, 256\}$ in the regular case, and 1,024 from $\{16, 512, 1024, 2048, 4096, 8192\}$ in the weak case. The Triton iterations and tensor computations compile with `fullgraph=True`. Reusing the E-step estimates verifies these updates without additional fixed-point iterations.

Weighted certificates on sparse cyclic graphs. Three 64-state topology instances, labeled A–C, join a bidirectional ring to two permutation shortcuts, with distinct four-way outgoing actions and four portal states that can exit. The origin is chosen farthest from the portals. Each graph uses eight independent initial Q vectors and twenty updates in each of the same two regularization/terminal-shift regimes. The proposal budgets, 10^{-9} tight-score accuracy and 2×10^{-8} acceptance margin are unchanged. A uniform visitation drift vector cannot satisfy strict contraction on nonportal rows, so these tests require the weighted construction in Theorem F.1.

Table 3: Controlled learning with shared proposals and identical final Q. The controls cache previous scores or adapt direct score intervals; the two-residual certificate reuses the E-step estimates. Time cells give median [minimum, maximum] over three warmed runs. A sweep advances one linear iteration for the whole batch of 256 models.

Acceptance test	Extra sweeps	Total sweeps	Warm time (ms)
Regular absorption			
Cached tight score	2,560	7,680	32.4 [30.4, 32.9]
Direct score interval	160	5,280	26.8 [25.9, 27.0]
Two-residual certificate	0	5,120	27.3 [25.6, 27.7]
Weak absorption			
Cached tight score	286,720	1,597,440	1009.7 [1007.8, 1013.8]
Direct score interval	20,320	1,331,040	853.3 [849.4, 861.6]
Two-residual certificate	0	1,310,720	787.5 [786.3, 789.9]

Table 4: Sparse 64-state learning: milliseconds, median [minimum, maximum] of three warmed runs. All rules share the same proposals and final Q.

Topology	Cached tight score	Direct interval	Certificate
Regular absorption			
A	84.5 [70.6, 85.4]	125.1 [124.6, 145.0]	43.1 [42.3, 44.4]
B	82.1 [72.2, 110.7]	121.2 [108.6, 123.7]	55.6 [44.3, 63.2]
C	70.6 [69.3, 75.8]	113.8 [106.9, 126.8]	42.8 [42.2, 44.9]
Weak absorption			
A	485.1 [479.4, 485.8]	411.0 [406.2, 413.5]	168.2 [166.8, 171.1]
B	498.9 [456.0, 526.8]	442.7 [432.4, 447.5]	176.3 [170.7, 183.3]
C	470.3 [470.0, 483.0]	417.4 [412.2, 435.4]	175.7 [166.0, 179.5]

Truncated unit-right-hand-side resolvent iterations build the score-kernel witness w, stopping only after directed arithmetic verifies $(I - A)w \geq \frac{1}{2}\mathbf{1}$. Value and visitation refinement each have a fixed 32,768-sweep cap. Directed products, sums and logarithm intervals bound the exact stored record kernels; independent dense CUDA solves audit the scores and kernel construction. Small exact-rational references check the directed primitives, including signed visit estimates and subnormal logarithm inputs. The tight control caches its old value interval and constructs only the witnesses it needs for the new decision.

All three rules accept all 960 distinct proposals and follow identical trajectories, with no audited score decrease or bound violation. All 288 checks using both proposal directions and computed, zero or signed visit estimates pass, including 144 decreasing cases. A further 48 checks reject the invalid uniform drift witness. Table 4 reports every topology and regime. Timings charge the shared proposals, witness construction, refinement and checks; dense audits are separate. The certificate uses no further value or visit iterations, but its witness construction uses 232–240 batched sweeps in the regular regime and 25,008–26,544 in the weak regime. The fullgraph CUDA/Triton runs have no graph breaks.

N.3 Combining completion, reuse, and full-flow checks

We first test partial cache-based proposals on a small cyclic fixture, then add the full-flow stopping rule and fallback. This distinction matters: a local reuse budget alone does not certify the quality of the whole fitted policy.

Integrated completion and reuse. The graph has origin 0, exit 4 and actions $(01, 04, 12, 24, 22, 23, 32)$. All exposures are one and $\lambda = 0.125$. The equally weighted interior records are $\varnothing$, (1), $(2, 2)$ and (3), with detection probabilities $\rho_1 = \rho_2 = 0.5$ and $\rho_3 = 1$; endpoints are forced. Thus completion must sum hidden loops and the alignments of repeated anchors. Eight initial Q perturbations of standard deviation 0.03 are drawn for each central policy

Table 5: Integrated partial proposals without full-flow fallback: all sixteen initializations and twelve attempts. Accepted columns split refreshed and reused modules. Rejected columns split negative and positive independently audited score gains, using a 10^{-10} sign allowance.

Initial $(\pi_{24}, \pi_{22}, \pi_{23})$	Attempts	Accepted		Rejected	
		Refresh	Reuse	Negative	Positive
$(0.50, 0.25, 0.25)$	96	26	34	21	15
$(0.10, 0.65, 0.25)$	96	0	0	96	0

$(\pi_{24}, \pi_{22}, \pi_{23}) = (0.50, 0.25, 0.25)$ and $(0.10, 0.65, 0.25)$. The origin enters the module with probability 0.7. Every initialization receives twelve proposal attempts.

Each round uses sixteen backward and visitation sweeps. With raw physical counts $\widetilde{c}$, the explicit balanced projection is

$$c = (z_E, 1 - z_E, z_E, z_E, x_E, y_E, y_E), \qquad z_E = \widetilde{c}_{01}, \quad x_E = \widetilde{c}_{22}, \quad y_E = (\widetilde{c}_{23} + \widetilde{c}_{32})/2.$$

Positive coarse flows have the same form $f(z, x, y)$. The cached module from 1 to 4 has unit flow $(1, 1, a, b, b)$ on $(12, 24, 22, 23, 32)$, so $x = za$, $y = zb$. We solve for z at fixed (a, b) and check its reuse bound. The exact potential minimum in (11) is

$$z \min_h \Psi(h; \theta, w_0) = \frac{z}{\lambda} \left[a\mathfrak{e}(\theta_{22}) + 2b\mathfrak{e}((\theta_{23} + \theta_{32})/2) \right].$$

Directed logarithm intervals and $\mathfrak{e}(t) \le |t|^2/[2(1 - |t|/3)]$ for $|t| < 3$ give an upper bound. If it exceeds 0.001 or cannot be evaluated, one module solve refreshes (a, b) at fixed z. Four node potentials then reconstruct $Q_e = \log \pi_f(e) + V_{s_e}$. The original-record certificate alone decides whether to apply this partial proposal; a refreshed cache is committed only with an accepted update. The initial cache need not be optimal.

The original record scores use product chains with three, three, five and five productive states. Directed Triton operations verify the acceptance bound for their stored coefficients. Kernel construction, count projection, flow optimization and potential reconstruction are audited numerically. For an independent score oracle, write the seven action weights as (s, d, t, e, l, p, q) and $D_0 = 1 - l/2$. Summing the hidden loops gives

$$(C_\varnothing, C_{(1)}, C_{(2,2)}, C_{(3)}) = \left(d + \frac{ste}{4D_0}, \frac{ste}{4D_0}, \frac{stel}{8D_0^3}, \frac{stpqe}{8D_0^2} \right).$$

These expressions agree with separate dense CUDA solves. The maximum count-projection correction is 8.8408×10^{-6} in infinity norm. Full three-variable flow solves, used only after the production loop, audit the original fine-graph gap and its path-KL decomposition; the maximum decomposition error is 3.5528×10^{-15}.

All 60 accepted updates improve the independently evaluated original score. All 34 accepted reuses follow a previous accepted refresh and change both the cached reference count vector and throughput by more than 10^{-10}. Their largest conditional reuse loss is 0.0008386924, and the largest directed upper bound is 0.0009151127. Both module path-KL bounds hold. Their largest full fine-graph gap is 0.0009437096; across all accepted updates it is 0.0376698048, so the module budget is distinct from a full-flow stopping tolerance. The second regime retains its initial model: all 96 partial proposals decrease the original score and are rejected. The fifteen rejected positive gains in the first regime are at most 1.0083×10^{-7}, illustrating that rejection need not mean non-ascent.

Full-gap stopping and fallback. We next extend the same fixture with a full-flow fallback and compare three proposal rules: a joint full solve, the certified cache proposal above, and an unchecked cache proposal that keeps (a, b) without a local module test or conditional refresh. All three require the same full fine-graph gap $\Delta \le 0.001$ and the same original-record acceptance bound. Thus "unchecked" refers only to the local cache proposal. The two final checks remain in every arm.

The full routine starts from (c_{01}, c_{22}, c_{23}), reconstructs the four potentials, and checks the gap before any joint Newton step. It then checks again after each single accepted joint step, stopping at the first

Table 6: Charged costs of the twelve-round partial-proposal validation loop. Times include remaining compilation after primitive checks; independent audits take a further $17.0242\,\mathrm{s}$. Every rejected proposal is charged.

Phase	Seconds
Completion	1.4754
Count projection	0.3953
Coarse flow solve	5.1767
Cache certificate	0.8378
Module refresh	3.3293
Fine-Q reconstruction	2.3291
Fine-graph gap	3.2219
Original-record acceptance	2.2242
Complete loop	19.0059

Table 7: Common-gap learning with a full-flow fallback. Accepted updates are split between the regular and difficult initial regimes (96 opportunities each). Fallback and reuse counts pool both regimes. Times are milliseconds, median [minimum, maximum] of three warmed loops.

Proposal	Accepted regular / difficult	Fallback	Reuse	Warm time (ms)
Joint full solve	65 / 56	0	0	323.3 [321.1, 324.6]
Certified cache	91 / 96	106	62	1736.4 [1715.8, 1740.1]
Unchecked cache	96 / 96	128	64	1023.3 [1007.1, 1048.0]

successful gap, with at most 32 joint steps. Potential reconstruction has a 32-step cap; the step cap is 3, the Armijo coefficient is 0.0001, and the line search allows 40 backtracks. The gradient tolerance 10^{-10} is a secondary stopping diagnostic and does not require solving past an achieved gap. A cache proposal that fails either final check calls this same full routine and is tested again on the original records. An unresolved result keeps the previous model and cache.

We retain all sixteen initializations and twelve outer rounds in every arm. The common gap is a tolerance for the current frozen counts; the learners' later counts and trajectories can differ. One warm-up and three warmed runs are recorded per arm. Timing includes completion, projection, every proposal, gap/witness check, rejection and fallback inside the loop. Initialization, primitive controls and independent audits are separate. The current batched fallback recomputes record certificates for all sixteen models, and that work is charged. Repeated runs follow identical trajectories within each arm.

Every difficult-regime start now accepts updates in all three arms. All accepted gaps meet 0.001, and independent record solves find no accepted decrease or bound violation. A separate conditional optimization checks every actual cache reuse and both module path KLs. For the 62 certified reuses, the largest loss is 0.0008254476 and the largest directed upper bound is 0.0008877513. The joint full solve is fastest on this small graph. After twelve rounds, mean record scores are -2.8708920, -2.8709544 and -2.8709266 for full, certified-cache and unchecked-cache proposals, respectively. These are common-tolerance learning comparisons, not identical final objectives or a time-to-global-optimum comparison.

N.4 Potential refinement on generic cyclic modules

These frozen queries measure whether tightening the certificate retains more caches and whether that additional work saves time. They isolate the reuse decision from an outer learning loop.

Workload and controllers. We isolate the cost of choosing a useful potential on a fixed set of module queries. Each module has one entry, one exit and $m \in \{8, 32\}$ internal states, with a bidirectional ring, one distinct permutation shortcut per state and two or four exit portals, respectively. For each size, two independently drawn shortcut/portal configurations are labeled A and B. There are 27 or 101 physical actions. Four initial policies per setting use within-row Gaussian logits of standard

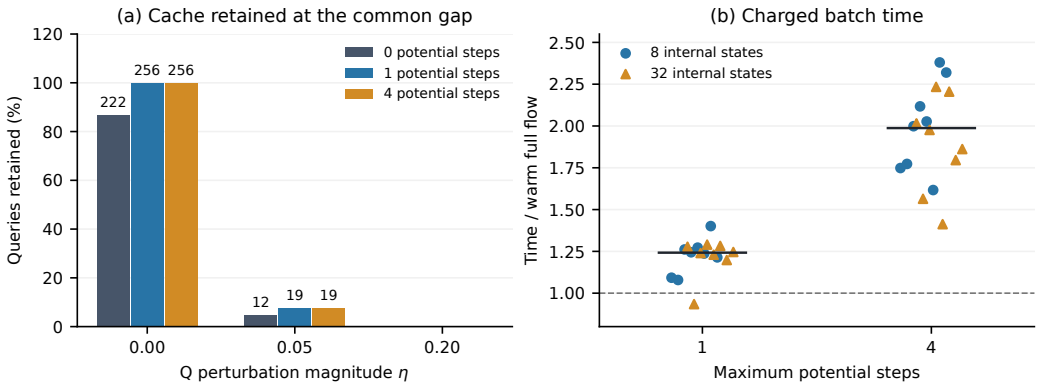

Figure 4: Generic frozen module queries. **(a)** Cache retention among 256 queries at each perturbation magnitude; labels give counts. The zero-step control also checks the initial gap before solving. **(b)** Every point is a paired ratio of batch medians over three warmed runs, relative to that control. Horizontal bars give the median over all sixteen fixed batches; they are not confidence intervals.

deviation 0.35, with portal exit probability 0.4 or 0.05. Exposure is one and $\lambda = 0.125$. This gives sixteen batches after crossing topology, size, exit probability and record length.

Each input count vector is obtained from a Bernoulli recording product with three or six interior anchors, including repeated anchors. Detection probabilities are uniform draws in $[0.15, 0.7]$, fixed within a setting. Writing $a = \lfloor m/2 \rfloor$, $b = \lfloor m/3 \rfloor$ and $c = m - 1$, the anchor lists are (a, b, a) and (a, b, a, c, a, b). The productive products have 33, 57, 129 or 225 transient states. Perturbing each base Q by ηG, with fixed independent standard-normal edge entries G and $\eta \in \{0, 0.05, 0.2\}$, generates the new count vectors. Throughput is $z = \exp(\eta u)$ for a fixed independent $u \sim \mathrm{Unif}[-1, 1]$. The old throughput is one. Thus nonzero perturbations change both counts and throughput, without assuming that they are iterates of an outer learner.

Two caches are supplied: a tight numerical old fit and one positive feasible Newton step from the old count flow. Each of the 48 queries per batch uses one of these caches and a gap target $\varepsilon = \epsilon z L(w_0)/\lambda$, $\epsilon \in \{0.0001, 0.001\}$. All controllers start with the same cache and recover a potential by least-squares fitting of Dh to the negative current flow gradient. They evaluate the same full module gap $z\Psi(h; \theta, w)$. The warm full control updates the feasible flow when that check fails. The other controls first allow at most one or four Newton steps in h while holding the cached flow fixed; each stops if its gap target is met, and otherwise calls the same warm full-flow routine. Refinement retains its improved potential on fallback. Each query starts afresh from the specified cache; this is not a sequential cache-update or end-to-end learning study.

Both Newton solvers use Armijo coefficient 0.0001 and at most 32 line-search trials. Full-flow Newton has a 64-step cap and a nullspace parameterization that preserves balance numerically. Solved queries are removed from the active batch in every controller. A fused Triton edge kernel evaluates the potential value without constructing unused gradients or Hessians. Initial potential recovery, gap checks, refinement, indexing, scattering, line searches and all fallbacks are timed. Graph/count/cache construction and tightened reference solves are common input preparation outside that timing. Every batch has one warm-up and three timed repetitions. The arrays, labels and budgets are identical across controllers and repetitions.

All 768 queries in each controller meet their final gap target, with no unresolved query; the largest final gap divided by its target is below 0.9995. The zero-, one- and four-step controls retain 234, 275 and 275 of 768 caches. At $\eta = 0$ the counts are 222, 256 and 256; at $\eta = 0.05$ they are 12, 19 and 19; at $\eta = 0.2$ none is retained. Median paired batch-time ratios are 1, 1.2423 and 1.9875. One-step refinement is faster in one of the sixteen batches, while four-step refinement is slower in all of them (Table 8). The additional retained caches therefore do not yield a consistent time reduction in this design. All topologies, settings, queries and timing repetitions are retained.

Table 8: All frozen-query batches. Each cell gives milliseconds, median [minimum, maximum] over three warmed runs of 48 queries. m counts internal states, p_{exit} is the initial portal exit probability, and L is the number of anchors. Columns give the maximum potential-refinement steps

m	Topology	p_{exit}	L	0 (warm full)	1	4
8	A	0.40	3	9.72 [9.11, 10.37]	10.61 [9.91, 10.76]	16.99 [16.20, 17.46]
8	A	0.40	6	12.49 [11.82, 12.97]	13.47 [13.24, 15.71]	22.14 [22.06, 22.36]
8	A	0.05	3	7.77 [7.75, 9.15]	9.80 [9.75, 9.99]	15.52 [15.03, 15.56]
8	A	0.05	6	7.06 [6.91, 7.31]	8.79 [8.74, 10.09]	14.95 [14.83, 14.95]
8	B	0.40	3	8.75 [8.72, 8.94]	11.13 [11.03, 11.26]	17.73 [17.56, 17.79]
8	B	0.40	6	9.06 [8.97, 9.38]	11.21 [11.11, 11.30]	14.65 [14.23, 14.93]
8	B	0.05	3	6.05 [5.94, 6.13]	8.48 [8.46, 8.49]	14.40 [14.36, 15.58]
8	B	0.05	6	6.29 [5.62, 6.43]	7.64 [7.63, 7.69]	14.60 [14.44, 15.05]
32	A	0.40	3	9.95 [9.85, 9.97]	12.71 [12.44, 13.21]	20.05 [19.78, 20.60]
32	A	0.40	6	9.61 [9.57, 10.62]	8.98 [8.62, 9.01]	15.04 [14.96, 15.35]
32	A	0.05	3	9.56 [9.50, 9.57]	11.85 [10.79, 12.05]	18.89 [18.51, 18.89]
32	A	0.05	6	7.22 [7.09, 7.31]	9.31 [9.27, 9.47]	16.12 [16.11, 16.67]
32	B	0.40	3	13.32 [13.17, 13.63]	16.39 [15.62, 17.11]	18.82 [18.34, 19.07]
32	B	0.40	6	7.39 [7.15, 7.47]	9.48 [9.23, 10.86]	16.30 [16.16, 16.70]
32	B	0.05	3	10.54 [10.25, 10.57]	12.63 [12.48, 13.47]	18.93 [17.53, 19.42]
32	B	0.05	6	9.52 [9.43, 9.78]	11.86 [11.48, 11.94]	17.72 [17.61, 17.92]

For each fixed cache we also measure the slack above $\mathcal{E} + z\,\mathrm{KL}(P_1\|P_0)$, using a tighter numerical reference. Normalized by $zL(w_0)/\lambda$, median slack falls from 0.00031094 to 0.00001136 after one potential step and 0.00001126 after four. The reference uses the same flow solver at relative gap 10^{-11}; it is not an independent optimizer. Complementary checks use an independent scalar-objective differentiation, direct primal–dual expansion, flow balance and a separate physical-state resolvent for the recording law. Differentiating that resolvent's log normalizer with respect to physical log weights checks all 192 distinct record-count inputs: maximum log-normalizer error is 7.2×10^{-15} and scaled count error is 1.1×10^{-15}. The largest flow-balance residual is 2.8×10^{-15} and relative gap identity error is 2.2×10^{-15}. These are float64 numerical audits; the component study does not extend the stored-record arithmetic certificate to the complete construction pipeline.

As a separate optimizer check, we solve the inverse-Q dual directly from $Q = 0$ on all 192 distinct count/throughput problems. CUDA automatic differentiation supplies its gradient and Hessian; damped Newton steps use an infinity-norm cap of 3, Armijo coefficient 0.0001, at most 40 trials per step and a 100-step cap. All problems reach a maximum gradient divided by $1 + \|c\|_\infty + z$ below 5×10^{-11}. An absorbing-policy resolvent then recovers unit flow. Against all 768 supplied reference flows, the largest objective difference divided by one plus its magnitude is 6.5×10^{-11}, and each directional path KL is below 1.2×10^{-10}. This implementation uses neither the production nullspace parameterization nor its flow derivatives.

N.5 Additional audit details and the pilot recording model

This subsection records the constructive kernel-refinement workload and the pilot-specific recording and module-selection settings. It completes the reproducibility details for the checks reported above; these are separate from the generic frozen-query timing experiment.

Constructive kernel refinement and audit targets. The constructive completion test draws 4,096 kernels on eight cyclic states. Each row has a self-loop and a transition to the next state modulo eight; only row seven can exit. Draw three independent standard-normal logits per row and normalize them by softmax. On rows zero to six, remove the exit and renormalize the two remaining entries. Multiply each positive weight independently by $\exp(-U)$, $U \sim \mathrm{Unif}[0.2, 0.7]$. Multiplication by 1.02 preserves properness because the resulting total row weights are at most $1.02e^{-0.2} < 1$. For a transient kernel P, its drift candidate is $\sum_{j=0}^{63} P^j\mathbf{1}$; the strict drift inequality is checked componentwise. The upper solution is constructed from the backward residual and a positive drift of the score kernel. Backward values start at one and visit vectors start at zero. Both use 4, 16, 64 or 256 positive iterations. The candidate kernel multiplies every positive old score weight by 1.02; the

resulting kernels remain proper. Exact backward solves evaluate the true change but do not enter the certificates. The largest drift factor among the tests is 0.9997 after rounding.

Completion is checked through path KL, the certified upper radius, and observed objective change. The flow certificate is checked against a high-accuracy tabular optimum and its full-path KL. Module response is checked through finite differences, the covariance formula and the constant-exposure control. Compression is checked through reconstructed fine flows, objective error, policy KL and charged computational cost.

Pilot recording model. Detection is calibrated per fold in four usage-density quartiles of the 110 m cells, on non-evaluation travellers only: the fold-T table is $\rho = 0.38, 0.45, 0.66, 0.78$ from the sparsest quartile to the densest, against a pooled mean of 0.75. A cell entered during a gap contributes $1 - \rho$ and an anchor contributes ρ, so a record's weight is a product over its segments between successive detections, allowing undetected revisits; the sensitivity arms halve and double the odds and replace the table by its pooled mean.

Numerical identities are checked at absolute or scaled tolerances of 10^{-9}; finite differences use 2×10^{-5} because they also contain a truncation error. Values at the roundoff scale are treated as signed numerical residuals. Exact module closure is established from exposure potentials, not from thresholding a measured response. The integrated studies use the explicit balancing map above and report its correction and cost separately.

The pilot module check uses $\lambda = 0.25$ on the first eight nonempty destination zones in the fixed ordering of the pilot training subset, containing 542 records. Lengths and distances are in kilometres. Initialize $Q_0(e) = -4\ell_e + 4[d(s_e, d) - d(t_e, d)]$, where $d(v, d)$ is Euclidean distance to the destination-zone centre. For each observed origin o, include cells v with $d(o, v) + d(v, d) \leq 1.3d(o, d) + 1$, take their union with the visible anchors, and retain only cells that reach the destination within this set. At this fixed Q, retain positive-count edges that strictly decrease hop distance to the destination. Consecutive cut nodes carrying the full destination throughput define candidate two-terminal blocks; forward and backward reachability remove dead ends. Keep blocks with an internal choice, at most 24 states and 48 edges. The 19 retained modules have 2–19 states and 2–28 actions. Positive reference counts are floored at 10^{-12}, and throughput ratios are 0.5, 1 and 2. Feasible-flow Newton uses a 300-step cap and relative first-order tolerance 10^{-10}.

Module validation. The pilot-corridor workload uses one completion at fixed initial parameters. All 19 modules pass their original checks after CUDA float64 solves in feasible flow coordinates; all 9 unequal-exposure modules reach the required optimum. The largest relative first-order residual is 3.3×10^{-11}, and the largest realized-loss ratio to the damped DAG bound is 0.941. This validates the numerical reuse calculation on these modules; it is not a measured training speedup. The Newton step and parallel line search compile with `fullgraph=True`; a fused Triton kernel evaluates objective, gradient and Hessian-diagonal terms.

Recorded numerical evidence. The reported results comprise completion and sparse-solver checks, constructive response and sharpness tests, controlled cyclic learning, and frozen module validation. Accuracy is assessed against the stated equations, independent references and numerical tolerances; timing charges the phases specified for each experiment. These experiments assess numerical accuracy and computation at fixed learning objectives.