WHEN IS DELETION ORDERING TRACTABLE? FROM UPDATE DYNAMICS TO PERMUTATION STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Given a fixed set of pending deletion requests, retraining from scratch after each request is prohibitive, so a prescribed request-wise policy processes them sequentially. The resulting terminal model can depend on their order. Rather than prescribing an ordering rule, we study the permutation objective induced by the fixed policy and ask when it admits simpler structure. We identify two independent reductions: position additivity represents the objective by request–position costs, reducing optimization to assignment and, with a shared positional profile, sorting; suffix localization removes dependence on the distant prefix while retaining interactions among the surviving requests. Under shared affine updates, we characterize the quadratic interactions that obstruct additivity, prove the reductions' independence, and show that suffix-conditioned assignment improves the approximation rate from $O(\rho^L)$ to $O(\rho^{2L})$. Experiments recover both structures in executed objectives. A controlled damped-Newton sweep shows that stronger contraction shifts the objective toward shorter, more suffix-specific dependence, while two full-network policies exhibit distinct positional and within-suffix structure. Structures identified from compact execution sets also predict unseen orders. These results frame deletion ordering as identifying the computational structure induced by the executed updates.

1 INTRODUCTION

A deployed model is often required to remove specific training data after the fact, and retraining from scratch on every request is prohibitive, so approximate unlearning edits the trained model in place (Guo et al., 2020; Kurmanji et al., 2023; Koloskova et al., 2025). When a prescribed request-wise policy processes a pending set, each request acts on the model produced by its predecessors. The same pending requests can therefore lead to different terminal models under different orders; with the request set, policy, and per-request budget fixed, the order is the decision variable.

Existing work has shown that deletion order matters and has proposed schedules based on uncertainty, difficulty, or forgettability (Zhao et al., 2024; Kumar et al., 2026). These rules choose an order, but do not characterize the optimization problem induced by the sequential updates. The objective over the $k!$ schedules of k pending requests may reduce to independent request scores, request–position costs, dependence on only the most recent requests, or none of these. Unlike classical scheduling and training-order problems, where the cost structure is prescribed (Smith, 1956; Rajput et al., 2022; Li & Hiratani, 2025), here it is created by composing the deletion updates and evaluating the terminal model.

We fix the deletion requests, update policy, and terminal objective, and vary only their order, defining an objective $J(\pi)$ over permutations $\pi \in S_k$, where S_k is the set of all $k!$ request orders. We study two ways in which this objective can be simplified (Figure 1). *Position additivity* represents $J(\pi)$ by request–position costs, reducing order selection to linear assignment and, under a shared positional profile, to sorting. *Suffix localization* makes the distant prefix irrelevant once the final L requests are fixed, reducing the search to ordered suffixes. The two reductions discard different information. Position additivity removes cross-request interactions across positions, whereas suffix localization removes dependence on the distant prefix. They can therefore hold separately or together.

The update dynamics determine which reduction, if any, is available. Even when request responses combine additively at the endpoint, a squared terminal objective introduces pairwise request–

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.

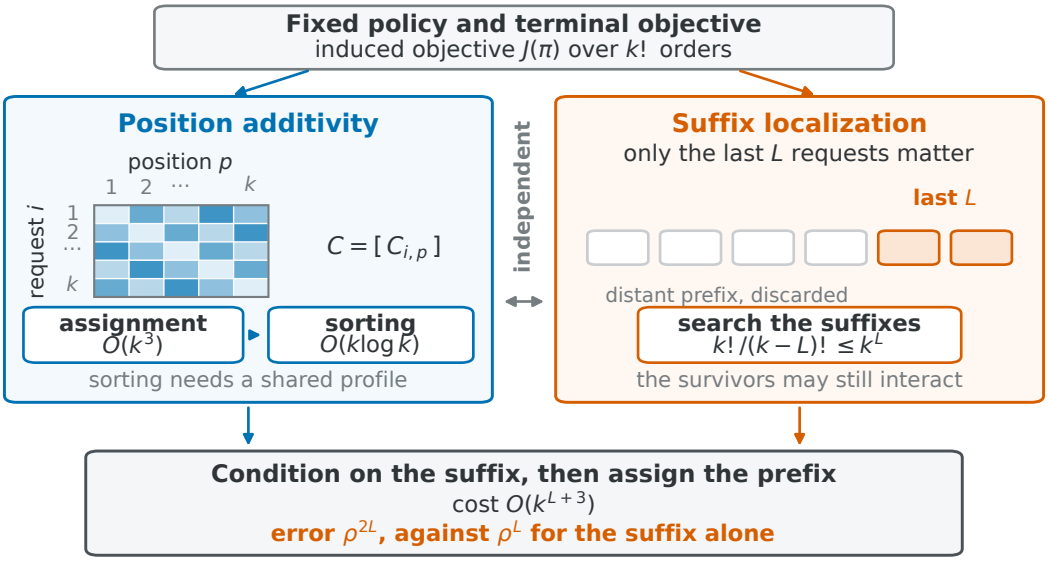


Figure 1: **Two independent routes to a simpler ordering problem.** Left, position additivity represents the objective by request–position costs. Right, suffix localization discards the distant prefix. Below, the two compose, at the rate Proposition 2(c) gives for shared affine updates.

position interactions, so endpoint additivity need not imply position additivity. Conversely, contractive updates suppress dependence on the distant prefix without eliminating interactions among the surviving requests. We characterize both mechanisms, prove that the two reductions are independent and composable, and show that contraction can yield a useful approximation without making exact optimization tractable.

We then test these structures in executed unlearning objectives. Across frozen-head and full-network settings, the induced objectives occupy different structural regimes. A deterministic damped-Newton intervention, used here because its Jacobian is analytic, shows that stronger contraction shifts dependence toward shorter, more suffix-specific representations, while held-out studies show that structures identified from compact execution sets transfer to unseen orders. The cost of identifying a representation from executions is measured separately from the cost of optimizing once that representation is available.

Contributions. We make three contributions: **(i)** we formulate deletion ordering as an induced permutation objective and identify position additivity and suffix localization as two independent structural reductions; **(ii)** we derive assignment, sorting, and finite-suffix reductions with approximation guarantees, characterize their composition, and establish a boundary for exact optimization under contraction; and **(iii)** we connect these structures to executed frozen-head and full-network dynamics and test their transfer beyond exhaustive small-k tables.

2 Related Work

Approximate unlearning uses gradients, curvature, distillation, or parameter saliency (Guo et al., 2020; Kurmanji et al., 2023; Fan et al., 2024; Foster et al., 2024; Koloskova et al., 2025). Reference proximity and behavioral forgetting provide different evidence about an unlearning policy (Mavrothalassitis et al., 2025). We use reference distance as a dense terminal ordering objective and report behavioral gates and deletion trajectories separately. Guo et al. use a Newton correction, with loss perturbation for the general case, whereas Koloskova et al. use noisy fine-tuning on retained data. Our deterministic damped-Newton rule is an intervention chosen for its known Jacobian.

Zhao et al. compare one-shot unlearning with sequential processing of difficulty-refined forget subsets, and Kumar et al. study noncommutativity and uncertainty-based forget-set ordering (Zhao

et al., 2024; Kumar et al., 2026). These studies establish that request composition and order affect sequential unlearning. Our complementary question is which permutation objective those executions induce and when that objective admits sorting, assignment, or suffix reduction. Request-level scoring corresponds to the scalar-sorting specialization of this framework.

Adaptive and continual unlearning address deletion requests that arrive over time, including validity under adaptively chosen requests and utility across repeated language-model updates (Gupta et al., 2021; Gao et al., 2025). Our setting conditions on a finite pending batch and studies the objective over its execution orders, isolating schedule dependence under a fixed sequential policy.

Yu et al. study how the order of multi-stage training changes subsequent local unlearning that uses forget-set gradients (Yu et al., 2025). We hold the trained starting state fixed and vary the later deletion history; common deletion suffixes, rather than common training histories, define our memory measure. The changed history and the information available to the unlearning rule are therefore different in the two settings.

Classical scheduling derives sorting rules from supplied job costs and processing times, as in Smith's rule for weighted completion time (Smith, 1956). Research on task and training order studies how curricula, optimization stability, and function composition create history dependence (Li & Hiratani, 2025; Hardt et al., 2016; Singh et al., 2023; Cooper et al., 2023; Kubo et al., 2024; Rajput et al., 2022; Lu et al., 2022). Our tools draw on quadratic assignment and weak-sum linearizability (Wang et al., 2021; Punnen & Kabadi, 2013; Çela et al., 2014). In our setting, the costs are induced by a deletion policy and terminal objective, and each permutation unit is a stateful request that may contain many examples and internal updates.

Kubo et al. prove target-composition hardness for general increasing linear maps (Kubo et al., 2024). Our PARTITION construction instead isolates the shared strict-contraction subclass: every request has linear part ρI and is realized by a positive, non-overshooting gradient step, while the dimension grows with the instance and ρ approaches one. The detailed comparison is in Appendix A.2.

3 DELETION ORDERING AS AN INDUCED COMPUTATIONAL PROBLEM

Let the training set be the disjoint union of a final retain set R and deletion requests $F_1, \dots, F_k$. Training on all data initializes the execution state w_0. We fix an unlearning policy $\mathcal{U}$ and vary only the request permutation. All stochastic choices are bound to request identity, so changing the order changes composition rather than the random draw assigned to a request.

Request F_i induces a transition U_i, where the state w includes both the model and any policy state carried across requests. For $\pi \in S_k$,

$$w_\pi = U_{\pi_k} \circ \cdots \circ U_{\pi_1}(w_0). \tag{1}$$

Terminal objective. Let Φ be a declared comparison map and z_R a retained-data reference. Define the state objective $j(w) = \|\Phi(w) - z_R\|_2^2$. Our primary permutation objective is

$$J(\pi) = j(w_\pi) = \|\Phi(w_\pi) - z_R\|_2^2, \tag{2}$$

with lower values preferred. The comparison map lets one formulation cover parameter-, output-, and representation-level terminal objectives. The concrete instantiations appear in Section 5. We define

$$J^\star = \min_{\pi \in S_k} J(\pi), \qquad \Delta_J = \mathbb{E}_{\pi \sim \mathrm{Unif}(S_k)} J(\pi) - J^\star.$$

Thus Δ_J is the improvement available over a uniformly random order. We normalize structural errors only when $\Delta_J > 0$.

Two structural reductions. For *position additivity*, we approximate the executed objective J by an assignment-form surrogate

$$f(\pi) = c + \sum_{p=1}^{k} C_{\pi_p, p}, \tag{3}$$

where $c \in \mathbb{R}$ and $C \in \mathbb{R}^{k \times k}$ are free coefficients, so $C_{i,p}$ is the cost charged for serving request i at position p. Let $\mathcal{A}$ denote the family of all such surrogates. Exact position additivity means that some $f \in \mathcal{A}$ satisfies $f(\pi) = J(\pi)$ for every $\pi \in S_k$.

For *suffix localization*, write $\pi = \sigma s$, where s is an ordered length-L suffix and σ its prefix. For example, for four requests w, x, y, z, if $\pi = (w, x, y, z)$ and $L = 2$, then $\sigma = (w, x)$ and $s = (y, z)$. We define

$$W_L = \max_s \left\{ \max_\sigma J(\sigma s) - \min_\sigma J(\sigma s) \right\}. \tag{4}$$

The width W_L measures how much the objective can still vary with the prefix after the final L requests are fixed. Thus $W_L = 0$ means that J depends only on the ordered length-L suffix, while interactions within that suffix may still be arbitrary.

Uniform structural error. For position additivity, define the normalized assignment error

$$D_{\mathcal{A}} = \frac{2}{\Delta_J} \min_{f \in \mathcal{A}} \max_{\pi \in S_k} |J(\pi) - f(\pi)|. \tag{5}$$

Thus $D_{\mathcal{A}} = 0$ under exact position additivity.

For suffix localization, let $\mathcal{S}_L$ be the family of functions that depend only on the ordered length-L suffix. The best constant for each fixed suffix is the midpoint of its objective range, so $\inf_{f \in \mathcal{S}_L} \|J - f\|_\infty = W_L/2$. We therefore define

$$D_{S,L} = \frac{W_L}{\Delta_J}. \tag{6}$$

Thus $D_{S,L} = 0$ under exact length-L suffix localization. Throughout, *width* denotes the raw range W_L, while *error* denotes the corresponding quantity normalized by Δ_J.

The uniform norm gives a direct decision guarantee. If a surrogate f satisfies $\|J - f\|_\infty \le \varepsilon$ and $\widehat{\pi} \in \arg\min_\pi f(\pi)$, then

$$J(\widehat{\pi}) - J^\star \le 2\varepsilon,$$

which we call the *uniform regret bound*. We therefore normalize uniform structural errors by $2/\Delta_J$. An error below one is sufficient to guarantee that the selected order improves on the uniformly random-order mean.

4 Update Dynamics and Ordering Structure

A fixed unlearning policy induces a permutation objective through its update dynamics. We begin with the simplest analyzable case, where all requests share the same linear response, and ask when the resulting ordering problem reduces below exhaustive search over S_k. Throughout this section, w denotes the representation in which terminal distance is measured, so Equation 2 has $\Phi = I$. Proposition 2(a) separately allows a Lipschitz nonlinear comparison map.

4.1 Squared terminal distance creates interactions

Suppose every request has the shared affine transition $U_i(w) = Mw + b_i$. Repeated application gives the additive endpoint

$$w_\pi = M^k w_0 + \sum_{p=1}^{k} M^{k-p} b_{\pi_p}. \tag{7}$$

Had the terminal cost been linear, substituting Equation 7 would preserve this additivity and give an assignment objective $\text{const} + \sum_p C_{\pi_p,p}$ with $C_{i,p} = c^\top M^{k-p} b_i$. The squared terminal objective instead introduces pairwise interactions. Expanding Equation 2 gives

$$J(\pi) = c_0 + \sum_p C_{\pi_p,p} + 2 \sum_{p<q} \mathcal{K}_{\pi_p,\pi_q,p,q}, \tag{8}$$

where $\mathcal{K}_{ijpq} = b_i^\top (M^{k-p})^\top M^{k-q} b_j$ and the unary terms absorb the reference and diagonal contributions. Thus endpoint additivity does not imply position additivity of the induced objective. Squared terminal distance couples request pairs with position pairs. For $M = \lambda I$, the norm term is a Koopmans–Beckmann quadratic assignment problem with rank-one positional coefficient matrix $P_{pq} = \lambda^{2k-p-q}$. Linear assignment is therefore a special, rather than generic, induced structure. The full expansion, the Gram-matrix reading, and the perturbation bounds that carry a surrogate guarantee over to bounded deviations from the shared response are in Appendix A.1.

4.2 Position additivity yields assignment and sorting

Equation 8 locates the obstruction: the coupling between a request pair and a position pair. Ordering reduces to assignment exactly when that coupling can be absorbed into unary request–position costs, on the linearizable boundary characterized by Punnen & Kabadi (2013) and Çela et al. (2014). Exact linearization is a strong requirement, so we instead measure distance from it. Fix a position pair (p, q) with $p < q$. The off-diagonal coefficients of Equation 8 at that pair form a request-pair interaction matrix

$$T^{pq} := [2\mathcal{K}_{ijpq}]_{i,j} \in \mathbb{R}^{k\times k},$$

and this interaction is assignment-compatible when it separates into one term for the request at position p and one for the request at position q. We therefore decompose each entry as

$$T^{pq}_{ij} = a^{pq}_i + \beta^{pq}_j + \Xi^{pq}_{ij}, \qquad i \neq j,$$

where the first two terms are absorbed into request–position costs and Ξ is the remaining non-additive interaction. Collecting the additive terms over all position pairs yields a projected assignment matrix $\widehat{C}$.

Assignment collapses further to sorting when its costs share a common monotone positional profile,

$$C_{i,p} = a_i + s_i h_p.$$

For any fixed monotone profile h, the rearrangement inequality makes sorting the request scores s_i optimal. To quantify approximate collapse with a simple closed-form certificate, Proposition 1(b) specializes this profile to the centered linear choice $q_p = p - (k+1)/2$ and measures the residual from that structure.

Proposition 1 (Position reduction)**.** *(a) Suppose the executed objective differs uniformly from the shared quadratic surrogate by at most $\varepsilon_{\mathrm{surr}}$. Let $\widehat{\pi}$ minimize the projected assignment matrix and set $\|\Xi\|_F^2 = \sum_{p<q}\sum_{i\neq j}(\Xi^{pq}_{ij})^2$. Then*

$$J(\widehat{\pi}) - \min_{\pi} J(\pi) \leq 2\varepsilon_{\mathrm{surr}} + 2\sqrt{\binom{k}{2}}\,\|\Xi\|_F. \tag{9}$$

(b) Suppose instead that an assignment objective with cost matrix C approximates J uniformly within $\varepsilon_{\mathcal{A}}$. Every permutation pays a request-only and a position-only effect once, so double-center, $\bar{C} = HCH$ with $H = I - \frac{1}{k}\mathbf{1}\mathbf{1}^\top$. For the centered linear profile $q_p = p - (k+1)/2$, write

$$\bar{C} = sq^\top + \bar{\Xi}_q$$

as the projection of $\bar{C}$ onto the corresponding sorting structure, and let π_s sort s. Then

$$J(\pi_s) - \min_{\pi} J(\pi) \leq 2\varepsilon_{\mathcal{A}} + \sqrt{2k}\,\|\bar{\Xi}_q\|_F. \tag{10}$$

The two parts chain. The assignment matrix that part (a) projects out satisfies $\varepsilon_{\mathcal{A}} \leq \varepsilon_{\mathrm{surr}} + \sqrt{\binom{k}{2}}\,\|\Xi\|_F$, so substituting this into part (b) bounds the sorted order's regret by the two residuals together. A small interaction residual therefore certifies approximate position additivity, while a small linear-profile residual certifies that the corresponding scalar sorting rule suffices. The retained-gradient construction is in Appendix A.1, the linearizability details in Appendix A.2, and both projection proofs, together with the scope of the slice projection, in Appendix A.4.

4.3 Contraction localizes prefix dependence and composes with assignment

Memory reduction. Contraction provides a different simplification from additivity. A common contractive suffix attenuates differences between states produced by different prefixes. If the terminal objective is Lipschitz on the resulting states, its dependence on the distant history contracts accordingly, while interactions within the suffix remain unrestricted. Proposition 2(a,b) quantifies this, first for Lipschitz updates and then for the affine case.

Composition. Simply truncating the prefix also discards its interactions with the retained suffix, leaving an $O(\rho^L)$ error term. A sharper construction fixes the ordered suffix and absorbs these prefix–suffix interactions into unary request–position costs for the remaining requests. The only discarded terms are then prefix–prefix interactions, whose two factors have each been attenuated by at least L subsequent updates. This yields the $O(\rho^{2L})$ bound in Proposition 2(c).

Proposition 2 (Memory reduction and composition)**.** *(a) Fix an ordered length-L suffix s. Let $\mathcal{X}_s$ be its reachable pre-suffix states with diameter d_s, and suppose the suffix map U_s, the composition of the final L updates, is κ_s-Lipschitz on $\mathcal{X}_s$. If j is $L_{j,s}$-Lipschitz on $U_s(\mathcal{X}_s)$, then*

$$\max_{u,v\in\mathcal{X}_s} |j(U_s(u)) - j(U_s(v))| \le L_{j,s}\kappa_s d_s. \tag{11}$$

For the squared objective of Equation 2 with L_Φ-Lipschitz Φ, one may take $L_{j,s} = 2R_sL_\Phi$ with $R_s = \sup_{w\in U_s(\mathcal{X}_s)} \|\Phi(w) - z_R\|$. (b) Under the shared affine transition with $\|M\| \le \rho < 1$ and $\max_i \|b_i\| \le B$, retaining only the final L response terms approximates J uniformly over orders to $O(\rho^L)$, so the minimizer of the truncated objective has regret of the same order. (c) Under the hypotheses of (b), write $\pi = \sigma s$ and define the pointwise surrogate $\widehat{J}_L(\sigma s) = \|r_s\|^2 + \sum_{p\le k-L} C^{(s)}_{\sigma_p,p}$, where r_s is the residual fixed by s and $C^{(s)}$ contains the suffix-conditioned unary prefix costs. Then

$$\sup_\pi |J(\pi) - \widehat{J}_L(\pi)| \le \frac{B^2\rho^{2L}}{(1-\rho)^2}, \tag{12}$$

so its minimizer has regret at most $2B^2\rho^{2L}/(1-\rho)^2$. Minimizing $\widehat{J}_L$ enumerates the ordered suffixes and solves one assignment problem per suffix, at cost $O(k^{L+3})$.

At a fixed suffix depth, conditioning on the suffix before assigning the prefix therefore sharpens the affine rate from ρ^L to ρ^{2L}. Part (a), separately, establishes suffix localization with no additivity assumption on the surviving requests. For fixed contraction, uniformly bounded response constants, and a last-position decision signal that does not vanish, the depth a fixed tolerance requires is $L = O(\log(1/\tau))$, independent of k. Direct truncation to depth $2L$ has the same affine rate at optimization cost $O(k^{2L})$, so the conditioned construction has a smaller exponent in k only when $L > 3$. These costs assume that the structural representation is already available; constructing or identifying it is accounted for separately. The non-asymptotic constants behind (b), the Lipschitz constants for gradient-descent updates, the proofs, the queue-length result, and the contrasting refinement regime are in Appendix A.3.

4.4 Independence and exact complexity

Independence. Position additivity and suffix localization capture distinct properties of the induced permutation objective, and neither subsumes the other. A nilpotent contraction yields exact suffix localization while retaining non-additive interactions, whereas an exact assignment objective can retain positive suffix width.

Proposition 3 (Independence)**.** *Even under shared strict-contraction updates, position additivity and suffix localization do not imply one another, and each separation can cross the sufficient threshold of Section 3.*

Explicit low-dimensional witnesses, the queue lengths and suffix depths they cover, and their exact constants are in Appendix A.3.

Exact complexity. The approximation structure induced by contraction does not determine the complexity of exact optimization. Even a positive, non-overshooting gradient step can induce an NP-hard ordering problem.

Proposition 4 (Hardness under arbitrarily weak strict contraction)**.** *Deciding whether an order attains a given terminal objective is NP-complete even when every request is one positive, non-overshooting gradient step on a strictly convex quadratic, and hence a strict contraction with factor $\rho < 1$ that approaches one as the instance grows.*

The weak NP-hardness reduction from Partition has $\rho \to 1$ as the instance grows. It separates strict contraction from exact tractability but leaves the complexity for ρ bounded away from one open. The construction, proof, and fixed-contraction approximation are in Appendix A.2.

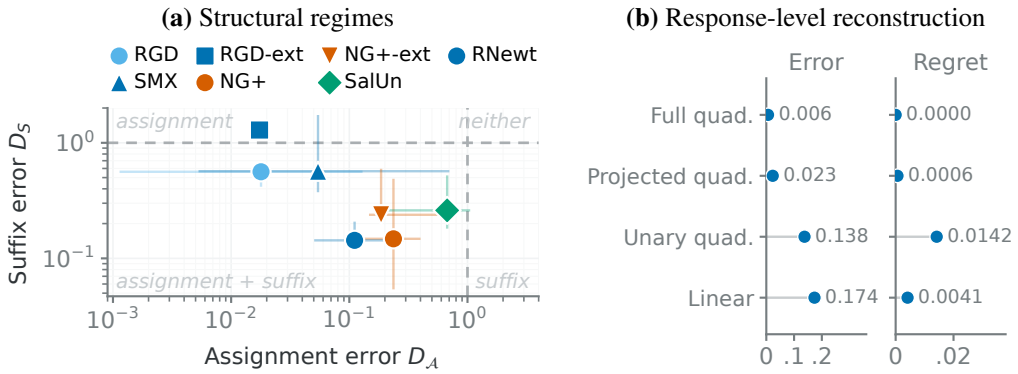


Figure 2: **Structural regimes and response-level reconstruction. (a)** Assignment and suffix errors across executed settings. Points are medians across instances, whiskers the observed ranges, and the dashed lines mark $D = 1$. Quadrant labels refer to the median point, and *-ext* marks a queue-extension setting. **(b)** Reconstruction error $2\varepsilon/\Delta_J$ and selection regret for four singleton-response constructions, over 45 complete $k = 4$ fresh-decision surfaces.

5 Empirical Structure of Executed Ordering Objectives

We ask which structural forms executed objectives take, which update dynamics produce them, and whether the structure can be identified from limited executions and transferred to unseen orders.

Settings. We study frozen-head and full-network unlearning. Frozen heads include ridge and softmax classifiers on CIFAR-10 features from an ImageNet-pretrained ResNet-18 (Krizhevsky, 2009; He et al., 2016; Deng et al., 2009), and ridge classifiers on AG News and 20 Newsgroups embeddings from DistilBERT (Zhang et al., 2015; Lang, 1995; Sanh et al., 2019). Full-network experiments delete CIFAR-100 superclasses from a ResNet-18 using NegGrad+ (Kurmanji et al., 2023) and SalUn (Fan et al., 2024). Queue lengths range from four to fifteen; RGD, SMX, NG+, SalUn, and RNewt abbreviate ridge gradient descent, softmax, NegGrad+, SalUn, and ridge Newton, with *-ext* denoting a queue extension. Frozen heads use squared parameter distance. The full-network objective averages, over a fixed 2,000-example test audit set, squared Euclidean distance between centered 20-superclass-logit vectors and a fixed two-run retain-only reference. It sums coordinates within each example; update and audit data are disjoint.

Protocol. Small queues are enumerated exhaustively, whereas longer-queue studies use disjoint identification and evaluation orders. To distinguish suffix-specific dependence from generic low-dimensional positional structure, we compare $D_{S,L}$ with equal-capacity prefix and middle controls $D_{P,L}$ and $D_{M,L}$, defined by conditioning on the first L positions or on a fixed contiguous middle block. Unless suffix depth is under study, $L = 2$ and we write D_S, D_P, and D_M. All structural errors are normalized by the random-order gap Δ_J. Full execution protocols and validity criteria are in Appendix B.

Objective anchoring. Reference distance compares completed sequences; utility and forgetting remain separate. On complete full-network surfaces it tracks retained more closely than deleted-group accuracy (Appendix E.2). Two fixed reference-free objectives on three auxiliary replays retain assignment and suffix-2 while matched controls fail (Appendix E.5); this retrospective result is not a prospective pipeline.

5.1 Executed objectives exhibit different structural forms

Executed objectives occupy different regimes (Figure 2a). The exhaustive ridge extension is assignment-only. Complete full-network surfaces support suffix localization; NegGrad+ also supports assignment throughout, while SalUn is more heterogeneous on that axis. Proposition 3 supplies the complementary suffix-only separation.

Position axis. A learned shared monotone profile brings NegGrad+ close to unrestricted assignment, while SalUn remains more heterogeneous; a fixed linear profile can therefore understate proximity to sorting.

Suffix axis. A low suffix error alone does not establish suffix-specific dependence. For frozen-head gradient objectives, an equal-capacity prefix fit is comparably accurate. On full-network objectives, matched prefix and middle controls separate the suffix structure. On the complete $k = 4$ surfaces, NegGrad+ is localized to the final request, whereas SalUn generally requires the final two (Figure 3b); its additive two-position suffix is also looser, showing residual interactions (Appendix C.3).

5.2 Update dynamics shape the induced structure

Quadratic interactions. Using the same singleton responses, the quadratic terminal score reconstructs the executed objective substantially more accurately than either its linear part or unary quadratic corrections carrying no interaction term (Figure 2b). Projecting those interactions onto request–position costs retains most of that accuracy and produces a tractable assignment objective. All constructions use no terminal-order observations, and the pattern holds across vision and text representations. A matched step-size intervention independently recovers the predicted second-order scaling, with a fitted slope of 1.97 against a predicted 2.

Controlled contraction. Varying Newton damping shifts the objective toward shorter, more suffix-specific representations (Figure 3a). For damped Newton the retained Hessian cancels, leaving a Jacobian of $(1 - \delta)I$ and hence known contraction $\kappa_1 = 1 - \delta$. On all five held-out instances, both suffix errors fall as contraction strengthens; the equal-capacity prefix error rises monotonically, and the middle control rises on four of five. Assignment is non-monotone, so the intervention changes where positional memory resides rather than simplifying the objective uniformly.

Beyond affine updates. Nonlinear full-network policies show the same localization. After a common two-request suffix, histories from different prefixes retain only 6.9% of their median objective difference under NegGrad+ and 10.2% under SalUn. At $k = 4$, effective suffix depth is one request for NegGrad+ and generally two for SalUn (Figure 3b). These are finite-execution observations; the causal claim rests on the controlled damping intervention above (Appendix C.2).

Composition. On the three complete $k = 5$ NegGrad+ surfaces, we compare rank-50 assignment models conditioned on the first, third, or final request. Four design-only folds hold out 30 cyclic-relabeling-balanced orders each; every 90-order training design has full rank, and fitting uses training-fold normalization.

Instance	Prefix-cond.	Middle-cond.	Suffix-cond.
seed0	$0.477\ [0.433, 0.522]$	$0.370\ [0.347, 0.501]$	$0.308\ [0.285, 0.323]$
seed1	$1.579\ [1.325, 1.941]$	$1.547\ [1.215, 2.053]$	$0.724\ [0.630, 0.990]$
seed2	$0.381\ [0.365, 0.409]$	$0.349\ [0.299, 0.357]$	$0.170\ [0.157, 0.190]$

Table 1: **Matched-rank position conditioning.** Entries are the median held-out D^{test} with observed $[\min, \max]$ over four design-only folds.

Final-request conditioning wins every instance–fold comparison, separating suffix specificity from capacity. The folds are not independent instances, and this one-depth test does not test the nonlinear ρ^{2L} exponent.

5.3 Identified structure transfers to unseen orders

The ridge identification orders are fixed from the family design matrix before objective values are observed. With eight identification executions, sorting recovers the unique optimum among all 40,320 orders in each of the three $k = 8$ ridge instances. Assignment fits more tightly and recovers the same optima, but requires 50 executions (Figure 4a). At $k = 15$, both families remain accurate on held-out orders. Newly executed minimizers of the fitted assignment objectives outperform every identification and held-out order in all three instances (Appendix D.2), although their global ranks remain unknown.

For full-network CIFAR-100, each instance uses 112 identification and 336 disjoint held-out orders. Assignment and suffix fits remain below threshold on held-out orders, whereas matched prefix and middle controls exceed it (Figure 4b). Executed unseen candidates pass the original screens, with mixed realized ranks (Appendix D.4); this tests structural extrapolation, not search efficiency.

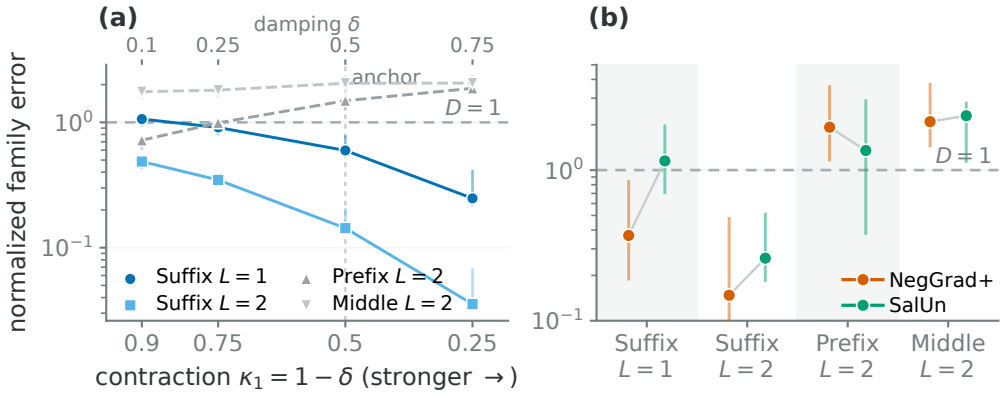


Figure 3: **Update dynamics and positional memory.** Points show medians across instances and whiskers the observed ranges, and the dashed line marks the normalized-error threshold $D = 1$. **(a)** Damped-Newton sweep, plotted against the analytic contraction $\kappa_1 = 1 - \delta$, with the damping that sets it on the upper axis. **(b)** NegGrad+ and SalUn compared across the suffix families and their equal-capacity prefix and middle controls.

(a) Ridge identification and transfer

Requests	Family	Fit executions	Held-out error
8	Sorting	8	0.068
8	Assignment	50	0.055
15	Sorting	15	0.107
15	Assignment	197	0.110

(b) Full-network held-out error

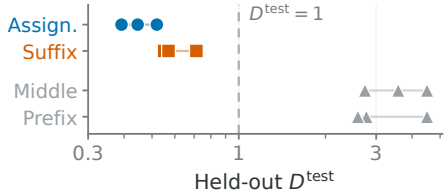


Figure 4: **Structural identification transfers to unseen orders.** **(a)** Sorting and assignment fits identified from compact execution sets and scored on orders held out of the fit, exhaustively at eight requests and on a disjoint 500-order sample at fifteen. **(b)** Held-out structural error D^{test} for the full-network prefix, middle, suffix, and assignment families, one point per instance, with the dashed line at $D^{\text{test}} = 1$.

6 Discussion and Conclusion

Deletion ordering need not admit a universal rule because its computational structure is induced jointly by the update dynamics and terminal objective. Position additivity and suffix localization expose two distinct simplifications: one removes cross-request positional interactions, while the other removes distant history without eliminating interactions among recent requests. This distinction explains why contraction can shorten memory without making the objective additive, and why suffix conditioning can compose with assignment. This also separates approximate from exact tractability: contraction can make recent-history reductions accurate, while strict contraction alone does not guarantee easy exact optimization. Across the executed settings, these structural regimes remain distinguishable under matched controls and can transfer to orders outside the identification set.

The resulting perspective is to treat deletion ordering as a structure identification problem before choosing a solver. A shared positional profile supports sorting, request–position costs support assignment, and localized history reduces the search to recent suffixes. Which reduction is appropriate therefore depends on what information the executed dynamics preserve in the terminal objective. A richer family is not automatically preferable: on the complete ridge surfaces, assignment fits more tightly, but sorting reaches the same optimum from fewer identification executions. Useful reductions must also remain predictive beyond their identification executions.

REPRODUCIBILITY STATEMENT

The anonymous supplementary repository contains the frozen protocols, the raw execution outputs, and the scripts needed to regenerate every reported table and figure and to verify the numerical claims. Analyses that use outcomes retrospectively are identified at first use. Held-out and pre-outcome constructions are versioned in the accompanying protocols, together with the partitions and checkpoints each run is bound to.

USE OF AI ASSISTANTS

AI assistants were used for research discussion, code development and debugging, experimental analysis, and manuscript editing. The authors verified all code, results, and claims and take responsibility for the content.

REFERENCES

Eranda Çela, Vladimir G. Deineko, and Gerhard J. Woeginger. Linearizable special cases of the QAP. *arXiv preprint arXiv:1409.6510*, 2014.

A. Feder Cooper, Wentao Guo, Duc Khiem Pham, Tiancheng Yuan, Charlie Ruan, Yucheng Lu, and Christopher M. De Sa. Coordinating distributed example orders for provably accelerated training. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, 2024.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI Conference on Artificial Intelligence*, 2024.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. On large language model continual unlearning. In *International Conference on Learning Representations*, 2025.

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning*, 2020.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Advances in Neural Information Processing Systems*, 2021.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1225–1234, 2016.

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

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31275–31298, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Susumu Kubo, Kazuhisa Makino, and Souta Sakamoto. Composition orderings for linear functions and matrix multiplication orderings. In *35th International Symposium on Algorithms and Computation*, volume 322 of *Leibniz International Proceedings in Informatics*, pp. 44:1–44:14, 2024. doi: 10.4230/LIPIcs.ISAAC.2024.44.

Varun Sampath Kumar, Esmaeil S. Nadimi, and Vinay Chakravarthi Gogineni. Controlling path dependence in gradient ascent unlearning through forget-set ordering. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 3224–3236, 2026.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, 2023.

Ken Lang. NewsWeeder: Learning to filter netnews. In *International Conference on Machine Learning*, 1995.

Ziyan Li and Naoki Hiratani. Optimal task order for continual learning of multiple tasks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34578–34603, 2025.

Yucheng Lu, Si Yi Meng, and Christopher De Sa. A general analysis of example-selection for stochastic gradient descent. In *International Conference on Learning Representations*, 2022. URL `https://dblp.org/rec/conf/iclr/LuMS22`.

Ioannis Mavrothalassitis, Pol Puigdemont, Noam Itzhak Levi, and Volkan Cevher. Ascent fails to forget. In *Advances in Neural Information Processing Systems*, 2025.

Abraham P. Punnen and Santosh N. Kabadi. A linear time algorithm for the Koopmans–Beckmann QAP linearization and related problems. *Discrete Optimization*, 10(3):200–209, 2013.

Shashank Rajput, Kangwook Lee, and Dimitris Papailiopoulos. Permutation-based SGD: Is random optimal? In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=YiBa9HKTyXE`.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Parantak Singh, You Li, Ankur Sikarwar, Weixian Lei, Difei Gao, Morgan B. Talbot, Ying Sun, Mike Zheng Shou, Gabriel Kreiman, and Mengmi Zhang. Learning to learn: How to continuously teach humans and machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11674–11685, 2023.

Wayne E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1–2):59–66, 1956. doi: 10.1002/nav.3800030106.

Yang Wang, Wei Yang, Abraham P. Punnen, Jingbo Tian, Aihua Yin, and Zhipeng Lü. The rank-one quadratic assignment problem. *INFORMS Journal on Computing*, 33(3):979–996, 2021.

Jiatong Yu, Yinghui He, Anirudh Goyal, and Sanjeev Arora. On the impossibility of retrain equivalence in machine unlearning. *arXiv preprint arXiv:2510.16629*, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *Advances in Neural Information Processing Systems*, 2024.

A PROOFS AND THEORETICAL DETAILS

A.1 EXACT AND APPROXIMATE STRUCTURE UNDER SHARED RESPONSES

Exact assignment under retained-gradient updates. A retained-gradient least-squares instance realizes exact position additivity. Let

$$\mathcal{L}_R(w) = \tfrac{1}{2}h_R\|w\|^2 - b_R^\top w, \qquad \mathcal{L}_i(w) = \tfrac{1}{2}h\|w\|^2 - b_i^\top w,$$

with $h_R, h > 0$ and $b_i^\top b_j = 0$ for $i \neq j$, take the terminal reference to be the retained-data minimizer $z_R = h_R^{-1}b_R$, and starting at the full-data minimizer take one fixed-size gradient step on the current retained loss after each deletion. Expanding the recurrence gives $w_\pi = \gamma_R b_R + \sum_p a_p b_{\pi_p}$ with a_p depending only on position, so $w_\pi - z_R = v + \sum_p a_p b_{\pi_p}$ for $v = (\gamma_R - h_R^{-1})b_R$, and orthogonality removes the cross-request terms to leave

$$J(\pi) = c_0 + \sum_{p=1}^{k} C_{\pi_p,p}, \qquad C_{i,p} = a_p^2\|b_i\|^2 + 2a_p v^\top b_i.$$

Linear assignment returns a global optimum. Writing $r_t = 1-\eta\{h_R+(k-t)h\}$ and $P_t = \prod_{u=t}^{k} r_u$ with an empty product equal to one, the coefficients are $a_p = P_1/(h_R + kh) + \eta\sum_{t<p} P_{t+1}$ and $\gamma_R = P_1/(h_R + kh) + \eta\sum_{t\le k} P_{t+1}$. These quadratics are ordinary least-squares blocks. Taking $X_i = \sqrt{h}I$ and $y_i = b_i/\sqrt{h}$ gives $X_i^\top X_i = hI$ and $X_i^\top y_i = b_i$, with the retain block analogous.

Interactions under general shared responses. Substituting Equation 7 into the squared terminal objective gives Equation 8. Specifically, let $\Psi_p = M^{k-p}$, $r = M^k w_0 - z_R$,

$$c_0 = \|r\|^2, \qquad C_{i,p} = 2r^\top \Psi_p b_i + b_i^\top \Psi_p^\top \Psi_p b_i, \qquad \mathcal{K}_{ijpq} = b_i^\top \Psi_p^\top \Psi_q b_j.$$

The squared terminal objective therefore couples request pairs (i, j) with position pairs (p, q) through the four-index tensor $\mathcal{K}$. For the scalar response $M = \rho I$ it factors as $\mathcal{K}_{ijpq} = \rho^{2k-p-q}\langle b_i, b_j\rangle$, separating a rank-one positional matrix from the request Gram matrix, which is the form Appendix A.2 uses to discuss linearizability.

Robustness to response mismatch. The following uniform bounds quantify the additional surrogate error when the executed responses deviate from the shared-response model. They require the stated bounds to hold uniformly over reachable states, so estimating the same quantities only along realized trajectories gives a finite-execution diagnostic rather than a uniform guarantee.

Linear mismatch. Let $U_i(w) = (M + E_i)w + b_i$ and $\bar{U}_i(w) = Mw + b_i$, with $\|M\| \le a$, $\|E_i\| \le \delta$, $\|b_i\| \le B$, and $\bar{a} = a + \delta$. Writing $\widetilde{M}_t = M + E_{\pi_t}$ for the executed linear parts, the telescoping identity $\widetilde{M}_m \cdots \widetilde{M}_1 - M^m = \sum_\ell \widetilde{M}_m \cdots \widetilde{M}_{\ell+1} E_{\pi_\ell} M^{\ell-1}$ with submultiplicativity gives $\|\widetilde{M}_m \cdots \widetilde{M}_1 - M^m\| \le m\delta\bar{a}^{\,m-1}$. Applying it to the initial state, which is multiplied by a product of length k, and to a displacement inserted at position p, which is multiplied by the $k - p$ matrices after it, gives

$$\|w_\pi - \bar{w}_\pi\| \le \Gamma_k := k\delta\bar{a}^{\,k-1}\|w_0\| + B\delta\sum_{m=1}^{k-1} m\bar{a}^{\,m-1}. \tag{13}$$

Expanding $\|w_\pi - z_R\|^2 - \|\bar{w}_\pi - z_R\|^2$ in $e_\pi = w_\pi - \bar{w}_\pi$ then bounds the terminal objectives uniformly by $2R_k\Gamma_k + \Gamma_k^2$, where $R_k = \max_\pi \|\bar{w}_\pi - z_R\| \le \|M^k w_0 - z_R\| + B\sum_{r<k} a^r$.

Nonlinear mismatch. Fix a common initial state, write $v = w - w_0$, and suppose that, uniformly over every reachable state v, deletion history S, and next request i,

$$U_{S,i}(w_0 + v) - w_0 = Mv + b_i + E_{S,i}v + d_{S,i} + r_{S,i}(v), \tag{14}$$

with $\|M\| \le a$, $\|E_{S,i}\| \le \delta$, $\|d_{S,i}\| \le \zeta$, and $\|r_{S,i}(v)\| \le K\|v\|^2/2$. If $\|v_t\| \le Q_t$, subtracting the shared surrogate recurrence $\bar{v}_{t+1} = M\bar{v}_t + b_{\pi_{t+1}}$ and inducting on

$$\Gamma_{t+1} = a\Gamma_t + \delta Q_t + \zeta + \frac{K}{2}Q_t^2, \qquad \Gamma_0 = 0, \tag{15}$$

gives $\|v_t - \bar{v}_t\| \le \Gamma_t$, and the same expansion as above bounds the terminal objectives by $2\bar{R}_k\Gamma_k + \Gamma_k^2$ for $\bar{R}_k = \max_\pi \|w_0 + \bar{v}_k - z_R\|$.

Write J_{aff} for the shared-response objective and let ε_{dyn} be the bound just obtained, so $\|J - J_{\text{aff}}\|_\infty \le \varepsilon_{\text{dyn}}$. If a surrogate $\widehat{J}$ satisfies $\|\widehat{J} - J_{\text{aff}}\|_\infty \le \varepsilon_{\text{surr}}$, then $\|J - \widehat{J}\|_\infty \le \varepsilon_{\text{dyn}} + \varepsilon_{\text{surr}}$, so the uniform regret bound gives any minimizer of $\widehat{J}$ regret at most $2(\varepsilon_{\text{dyn}} + \varepsilon_{\text{surr}})$.

A.2 Assignment reductions and exact hardness

Scalar-response linearizability. For $M = \rho I$, let $\alpha_p = \rho^{k-p}$, $P = \alpha\alpha^\top$, and $G_{ij} = \langle b_i, b_j \rangle$. Equivalently grouping the diagonal terms back into the quadratic term, the norm-generated part has the Koopmans–Beckmann form

$$\sum_{p,q} P_{pq} G_{\pi_p, \pi_q}.$$

The rank-one qualifier here applies to the positional coefficient matrix $P = \alpha\alpha^\top$. The request Gram matrix G may have higher rank and the request vectors may have arbitrary dimension. Related work studies QAPs with rank-one cost matrices (Wang et al., 2021). Here we state the restriction directly in terms of P and G, and what the hardness construction adds is the realization of such an instance by positive, non-overshooting gradient steps. Linearizability is then a known characterization rather than something we establish. A symmetric Koopmans–Beckmann QAP is linearizable as a linear assignment objective if and only if at least one coefficient matrix is weak-sum, meaning $H_{ij} = u_i + u_j$ off diagonal (Punnen & Kabadi, 2013; Çela et al., 2014). What is specific to our instances is where that condition fails. For the position product matrix,

$$P_{pq} + P_{rs} - P_{pr} - P_{qs} = (\alpha_p - \alpha_s)(\alpha_q - \alpha_r).$$

Four distinct geometric position weights therefore violate the condition. The same criterion identifies the exceptional request geometries through their Gram matrix G.

Proposition 1 measures distance from assignment without assuming a scalar response, and applies here too. Using $T_{ij}^{pq} = 2\mathcal{K}_{ijpq}$, independently least-squares project every off-diagonal slice onto $a_i^{pq} + \beta_j^{pq}$. Define

$$\widehat{C}_{i,p} = C_{i,p} + \sum_{q>p} a_i^{pq} + \sum_{q<p} \beta_i^{qp}.$$

Then, up to the order-independent constant,

$$J(\pi) - \sum_p \widehat{C}_{\pi_p, p} = \sum_{p<q} \Xi^{pq}_{\pi_p, \pi_q}.$$

Every permutation selects exactly $\binom{k}{2}$ residual entries, hence

$$\left| \sum_{p<q} \Xi^{pq}_{\pi_p,\pi_q} \right| \le \sqrt{\binom{k}{2}} \left(\sum_{p<q} \sum_{i \ne j} (\Xi_{ij}^{pq})^2 \right)^{1/2}.$$

Combining this uniform error with $\varepsilon_{\text{surr}}$ and applying the uniform regret bound proves the proposition. More generally, for predicted responses $e_{i,p}$ and a quadratic terminal score $\langle g, \delta \rangle + \frac{1}{2}\langle \delta, H\delta \rangle$, use

$$C_{i,p} = \langle g, e_{i,p} \rangle + \tfrac{1}{2}\langle e_{i,p}, He_{i,p} \rangle, \qquad T_{ij}^{pq} = \langle e_{i,p}, He_{j,q} \rangle.$$

This is the form used by the projected retained-loss realization. Slice additivity is only sufficient. Cancellations across position pairs can make a tensor linearizable even when this residual is nonzero, and a large bound does not exclude another tractable representation.

NP-completeness under positive small steps. To prove Proposition 4, reduce from Partition (Garey & Johnson, 1979). Given positive integers $a_1, \dots, a_N$, set $s_i = 2a_i$, $k = 2N$, and $S = \sum_i s_i$. Let

$$\mathcal{C} = \{-(2N-1), \dots, -3, -1, 1, 3, \dots, 2N-1\}$$

be the centered odd position weights. In dimension $N+1$, with standard basis $e_0, e_1, \ldots, e_N$, create two requests for each item:

$$b_{X_i} = \Omega e_i + s_i e_0, \qquad b_{Y_i} = -\Omega e_i - s_i e_0, \qquad \Omega = 4NS + 1.$$

Their sum is zero. If X_i and Y_i receive weights c_{X_i} and c_{Y_i}, write $d_i = c_{X_i} - c_{Y_i}$. For $v_\pi = \sum_p c_p b_{\pi_p}$,

$$\|v_\pi\|^2 = \Omega^2 \sum_{i=1}^{N} d_i^2 + \left(\sum_{i=1}^{N} s_i d_i \right)^2 . \tag{16}$$

Every d_i is a nonzero even integer, so $\sum_i d_i^2 \geq 4N$ with equality exactly when $|d_i| = 2$. Weights in $\mathcal{C}$ differing by two form a path on $2N$ vertices whose perfect matching is unique by endpoint forcing, so equality means adjacent weights are paired and $d_i = 2\sigma_i$ with $\sigma_i \in \{-1, 1\}$. Which pair goes to which item does not enter the objective. A proper pairing therefore has objective $4N\Omega^2 + 4(\sum_i \sigma_i s_i)^2 \leq 4N\Omega^2 + 4S^2$, while an improper one has some $|d_i| \geq 4$ and anchor term at least $(4N + 12)\Omega^2$. Since $12\Omega^2 > 4S^2$, every global optimum is properly paired and

$$\min_\pi \|v_\pi\|^2 = 4N\Omega^2 + 4 \min_{\sigma \in \{-1,1\}^N} \left(\sum_i \sigma_i s_i \right)^2 . \tag{17}$$

It remains to embed these centered linear weights in a positive geometric response. Set $B = \Omega + S$ and

$$\eta = \frac{1}{100B^2k^6}, \qquad \rho = 1 - \eta.$$

For position $p = 1, \ldots, k$, let $c'_p = 2(k-p) - (k-1) = k + 1 - 2p$, which traverses $\mathcal{C}$. Because $\sum_i (b_{X_i} + b_{Y_i}) = 0$, the constant term in $(1-\eta)^{k-p}$ cancels. Taylor's theorem, with $(1-\eta)^j = 1 - j\eta + r_j$ and $0 \leq r_j \leq j(j-1)\eta^2/2$, writes the exact endpoint under $U_i(w) = \rho w + b_i$ as $w_\pi = -\frac{\eta}{2} v_\pi + \xi_\pi$. Since $\sum_p |c'_p| = k^2/2$ and $\|b_i\| \leq B$, this gives $\|v_\pi\| \leq Bk^2/2$ and $\|\xi_\pi\| \leq Bk^3\eta^2/6$. Uniformly over permutations,

$$\begin{aligned} \left| \|w_\pi\|^2 - \frac{\eta^2}{4} \|v_\pi\|^2 \right| &\leq \eta \|v_\pi\| \|\xi_\pi\| + \|\xi_\pi\|^2 \\ &\leq \frac{B^2k^5}{12}\eta^3 + \frac{B^2k^6}{36}\eta^4 < \frac{\eta^2}{100}. \end{aligned} \tag{18}$$

If the Partition instance is positive, Equation 17 has minimum $4N\Omega^2$. Otherwise, every signed sum of the even s_i has magnitude at least two, so the minimum is at least $4N\Omega^2 + 16$. The rational threshold

$$\theta = \eta^2(N\Omega^2 + 2)$$

and Equation 18 strictly separate the two cases. All constructed rationals have polynomial bit length. Given a permutation, its rational endpoint and squared norm can be evaluated exactly with polynomially many bit operations, proving membership in NP.

Finally, the opposite request pairs give $\sum_i b_i = 0$, so with $L_R(w) = \frac{1}{2}\|w\|^2$ and $L_{F_i}(w) = \frac{1}{2}\|w + b_i/\eta\|^2$ both the full-data objective $L_R + \sum_i L_{F_i}$ and the final retain objective L_R have unique minimizer zero, while $w - \eta\nabla(2L_R(w) - L_{F_i}(w)) = (1-\eta)w + b_i$. Each quadratic is an identity-design least-squares block, so the hard instance has an explicit fixed-anchor retain–forget implementation whose update objective has Hessian I, positive response, and step size strictly between zero and one. The construction has $d = N + 1 = k/2 + 1$ and an instance-dependent ρ approaching one, and its fixed final-retain anchor is distinct from the prefix-changing retain loader used by NegGrad+.

This boundary differs from the target-composition hardness of Kubo et al. (2024). Their strong-hardness construction for increasing linear functions uses translations $x \mapsto x + a_i$ together with maps whose slope is larger than one. It therefore does not directly cover the shared strict-contraction subclass here. Conversely, our PARTITION reduction is only weakly NP-hard, uses growing dimension, and lets ρ approach one; it isolates realizability by shared positive gradient steps rather than superseding their broader result.

The construction and the small-step tractable regime are two sides of one expansion. A nonzero terminal anchor lets the first-order positional term dominate the $O(\eta^2)$ interactions, which is what makes small steps tractable there. Taking $w_0 = z_R = 0$ removes that anchor, so a positive non-overshooting step alone does not imply easy exact optimization.

A.3 Memory bounds and structural separation

Nonlinear common-suffix proof. For $u, v \in \mathcal{X}_s$, composition of the suffix maps gives

$$\|U_s(u) - U_s(v)\| \le \kappa_s \|u - v\| \le \kappa_s d_s.$$

Applying the state objective's Lipschitz constant on $U_s(\mathcal{X}_s)$ gives the general bound $L_{j,s}\kappa_s d_s$. For the squared objective the output difference is at most $L_\Phi \kappa_s d_s$, and writing $a = \Phi(U_s(u)) - z_R$ and $b = \Phi(U_s(v)) - z_R$,

$$\big|\|a\|^2 - \|b\|^2\big| = |\langle a + b, a - b\rangle| \le (\|a\| + \|b\|)\|a - b\| \le 2R_s L_\Phi \kappa_s d_s.$$

Maximizing first within each suffix and then over suffixes proves Proposition 2 with $L_{j,s} = 2R_s L_\Phi$. The Lipschitz constants $\nu_{s,t}$ are available for the update rules used here. For one gradient step of size η on a region with $\mu I \preceq \nabla^2 \ell \preceq \Lambda I$, $\nu \le \max\{|1 - \eta\mu|, |1 - \eta\Lambda|\}$.

Affine suffix truncation and quadratic representation. Under the affine assumptions of Proposition 2(b), retain only the final L response terms:

$$\widetilde{w}_{\pi,L} = M^k w_0 + \sum_{r=0}^{L-1} M^r b_{\pi_{k-r}}, \qquad \widetilde{J}_L(\pi) = \|\widetilde{w}_{\pi,L} - z_R\|_2^2.$$

The omitted prefix contribution is

$$h_{\pi,L} := w_\pi - \widetilde{w}_{\pi,L} = \sum_{r=L}^{k-1} M^r b_{\pi_{k-r}}.$$

Using $\|M\| \le \rho < 1$ and $\|b_i\| \le B$,

$$\|h_{\pi,L}\| \le \delta_L := B \sum_{r=L}^{k-1} \rho^r = B\frac{\rho^L(1 - \rho^{k-L})}{1 - \rho}. \tag{19}$$

Moreover,

$$\|\widetilde{w}_{\pi,L} - z_R\| \le \|M^k w_0 - z_R\| + B \sum_{r=0}^{L-1} \rho^r \le \|M^k w_0 - z_R\| + \frac{B}{1 - \rho}.$$

Since $w_\pi = \widetilde{w}_{\pi,L} + h_{\pi,L}$,

$$\begin{aligned} |J(\pi) - \widetilde{J}_L(\pi)| &= \big|2\langle \widetilde{w}_{\pi,L} - z_R, h_{\pi,L}\rangle + \|h_{\pi,L}\|^2\big| \\ &\le 2\left(\|M^k w_0 - z_R\| + \frac{B}{1 - \rho}\right)\delta_L + \delta_L^2 =: \varepsilon_L. \end{aligned} \tag{20}$$

Hence $\|J - \widetilde{J}_L\|_\infty \le \varepsilon_L = O(\rho^L)$, and the uniform regret bound gives Proposition 2(b).

Importantly, truncation removes dependence on the distant prefix without removing interactions among the retained suffix requests. Writing

$$\widetilde{w}_{\pi,L} - z_R = v + \sum_{r<L} d_{\pi_{k-r},r},$$

and expanding the squared norm gives

$$\widetilde{J}_L(\pi) = \|v\|^2 + \sum_{r=0}^{L-1} \left(2v^\top d_{\pi_{k-r},r} + \|d_{\pi_{k-r},r}\|^2\right) + 2 \sum_{0 \le r < s < L} d_{\pi_{k-r},r}^\top d_{\pi_{k-s},s}. \tag{21}$$

Thus suffix localization suppresses distant-prefix dependence while preserving arbitrary pairwise interactions within the retained suffix.

Suffix-conditioned assignment. Fix an ordered length-L suffix s and write $\pi = (\sigma, s)$, where σ contains the remaining $k - L$ requests. Conditioning on s fixes both the suffix response and its interaction with the initial state. Let

$$r_s = M^k w_0 - z_R + \sum_{j<L} M^j b_{s_{L-j}}, \qquad e_p(i) = M^{k-p} b_i,$$

so that

$$w_\pi - z_R = r_s + \sum_{p \le k-L} e_p(\sigma_p).$$

Expanding the squared norm separates the prefix dependence into unary and pairwise terms:

$$J(\pi) = \|r_s\|^2 + \sum_{p \le k-L} \left(2 r_s^\top e_p(\sigma_p) + \|e_p(\sigma_p)\|^2\right) + 2 \sum_{p<q \le k-L} e_p(\sigma_p)^\top e_q(\sigma_q). \tag{22}$$

The first two groups form an assignment objective over the prefix, with suffix-conditioned costs

$$C^{(s)}_{i,p} = 2 r_s^\top e_p(i) + \|e_p(i)\|^2.$$

We therefore define

$$\widehat{J}_L(\pi) = \|r_s\|^2 + \sum_p C^{(s)}_{\sigma_p, p},$$

which keeps the full unary effect of every prefix request and discards only prefix–prefix interactions.

This conditioning is sharper than truncating the prefix. Every prefix position satisfies $k - p \ge L$, so under $\|M\| \le \rho < 1$ and $\|b_i\| \le B$,

$$\|e_p(i)\| \le B\rho^{k-p} \le B\rho^L.$$

Hence each discarded interaction contains two responses that have each passed through at least L further updates. Using Cauchy–Schwarz and the geometric series,

$$\left|J(\pi) - \widehat{J}_L(\pi)\right| = 2 \left|\sum_{p<q} e_p(\sigma_p)^\top e_q(\sigma_q)\right| \tag{23}$$

$$\le 2B^2 \sum_{L \le u < t \le k-1} \rho^{t+u} \tag{24}$$

$$\le B^2 \left(\sum_{t \ge L} \rho^t\right)^2 = \frac{B^2 \rho^{2L}}{(1-\rho)^2}. \tag{25}$$

Thus conditioning on the suffix improves the affine approximation rate from $O(\rho^L)$ for prefix truncation to $O(\rho^{2L})$ by retaining the unary prefix effects and discarding only interactions between already attenuated prefix responses.

Finally, there are $k!/(k-L)! \le k^L$ ordered suffixes, and for each one the remaining prefix requires a single $(k-L) \times (k-L)$ assignment solve. The resulting search therefore costs $O(k^{L+3})$ for fixed L.

Rate check for truncation and conditioning. The two exponents are claims about a rate, which the executed study tests at a single depth. We check them directly under exactly the shared-affine assumptions of Proposition 2, on 40 synthetic instances at $k = 30$ with a five-dimensional state, fitting $\log \varepsilon_L$ against $L \log \rho$ over $L \le k/3$, where the prefix is still long enough for the rate to be observable (Table 2). The fitted exponents follow the predicted separation across all tested contractions, and the certificate $B^2\rho^{2L}/(1-\rho)^2$ holds at every tested length rather than only asymptotically. The errors are suprema over 2,000 sampled orders, which lower-bound the supremum over S_k.

Table 2: **The two affine decay exponents come out as predicted.** Fitted slopes of $\log \varepsilon_L$ against $L \log \rho$ at $k = 30$, over 40 instances per contraction. Proposition 2 predicts 1 for truncation, 2 for suffix-conditioned assignment, and hence a ratio of 2.

ρ	Truncation	Conditioning	Ratio
0.3	0.999	1.999	2.000
0.5	1.006	2.002	1.991
0.7	1.007	2.011	1.998

The rate comparison above holds at a fixed suffix depth. Reaching the same $O(\rho^{2L})$ error by truncating directly to depth $2L$ instead costs $O(k^{2L})$, so conditioning has the smaller exponent only for $L > 3$, with equality at $L = 3$. The case for conditioning at small L is therefore that it attains a rate the same depth of truncation does not, not that it is the cheaper route to that rate.

Queue-length-uniform bounds. The absolute errors above are controlled by contraction, but the normalized structural errors also divide by the decision scale Δ_J. To separate these two effects, define the observable last-position signal

$$\mu_1(k) := \mathbb{E}_\pi J(\pi) - \min_i \mathbb{E}[J(\pi) \mid \pi_k = i]\,. \tag{26}$$

It measures how much the expected objective can improve by choosing only the final request. Since every conditional mean is at least the global minimum,

$$\Delta_J = \mathbb{E}_\pi J(\pi) - \min_\pi J(\pi) \geq \mu_1(k).$$

Thus $\mu_1(k)$ is an observable lower bound on the normalization.

Suffix width. If two orders π and π' share the same ordered length-L suffix, then $\widetilde{J}_L(\pi) = \widetilde{J}_L(\pi')$. Equation 20 therefore gives

$$|J(\pi) - J(\pi')| \leq |J(\pi) - \widetilde{J}_L(\pi)| + |J(\pi') - \widetilde{J}_L(\pi')| \leq 2\varepsilon_L.$$

Taking the maximum over orders with a common suffix yields

$$W_L \leq 2\varepsilon_L. \tag{27}$$

Corollary 5 (Queue-length-uniform structural bounds)**.** *For an ordered length-L suffix s, let $\mathcal{C}_L^{\mathrm{cond}}$ contain functions of the form $f(\sigma s) = c_s + \sum_{p \leq k-L} C^{(s)}_{\sigma_p, p}$, and define*

$$D_{\mathrm{cond}}(L) := \frac{2}{\Delta_J} \inf_{f \in \mathcal{C}_L^{\mathrm{cond}}} \|J - f\|_\infty.$$

Under Proposition 2(b), there exists a constant c_ρ, depending on ρ, B, and $R_0 = \sup_k \|M^k w_0 - z_R\|$, but not on k or L, such that

$$D_{S,L} \leq \frac{2c_\rho \rho^L}{\mu_1(k)}, \qquad D_{\mathrm{cond}}(L) \leq \frac{2B^2\rho^{2L}}{(1-\rho)^2\mu_1(k)}. \tag{28}$$

The second bound applies to the suffix-conditioned family of Proposition 2(c). At $L = 0$, this family reduces to assignment, giving

$$D_{\mathcal{A}} \leq \frac{2B^2}{(1-\rho)^2\mu_1(k)}.$$

If, along a sequence of instances, the response constants remain uniformly bounded and the decision scale stays non-degenerate, $\inf_k \mu_1(k) = m > 0$, then any fixed normalized tolerance τ is achieved with

$$L \geq \frac{\log(2c_\rho/(m\tau))}{\log(1/\rho)} = O(\log(1/\tau)),$$

independently of k.

The proof is immediate from the previous bounds. Equation 27 gives $W_L \leq 2\varepsilon_L$. Equation 19 gives $\delta_L \leq B\rho^L/(1-\rho)$, and $\|M\| \leq \rho < 1$ keeps $R_0 = \sup_k \|M^k w_0 - z_R\|$ finite, so Equation 20 collapses to $\varepsilon_L \leq c_\rho \rho^L$ with c_ρ free of k and L. Dividing by $\Delta_J \geq \mu_1(k)$ yields the first inequality. Likewise, dividing Equation 12 by the same lower bound gives the second.

The non-degeneracy condition is essential. Contraction controls the numerator, whereas $\mu_1(k)$ controls whether the normalized decision scale remains visible as the queue grows. If every request behaves identically in the final position, then $\mu_1(k) = 0$ and the bound becomes uninformative. This does not imply that ordering itself is irrelevant, only that last-position variation no longer lower-bounds Δ_J.

The relevant scaling variables are therefore the contraction and the decision scale, rather than queue length by itself. Appending requests while keeping a fixed per-request contraction leaves the bound uniform in k. If the update instead becomes increasingly close to the identity as k grows, the generic argument no longer supplies a queue-length-independent depth, and the factor $(1-\rho_k)^{-2}$ also worsens. This deterioration of the upper bound is not an instance-wise lower bound on the suffix depth required.

The same separation extends beyond affine dynamics. Under Proposition 2(a),

$$D_{S,L} \le \frac{2L_\Phi \max_s(R_s\kappa_s d_s)}{\mu_1(k)}.$$

Hence a policy family with $R_s d_s \le C$ and $\kappa_s \le q^L$ uniformly in k inherits the same queue-length-uniform conclusion. We do not certify such uniform constants for the executed full-network policy. The larger-queue study therefore measures these quantities empirically rather than invoking this corollary.

Strict separation between memory and additivity. Neither suffix localization nor position additivity implies the other, even under shared affine dynamics. We show the two directions by explicit constructions.

Short memory does not imply additivity. Let $w_0 = z_R = 0$ and

$$M = \begin{pmatrix} 0 & 1/2 \\ 0 & 0 \end{pmatrix}, \qquad b_i = (x_i, x_i)^\top, \qquad (x_1, x_2, x_3, x_4) = (-2, 1, -1, 2). \tag{29}$$

Since $M^2 = 0$, only the final two requests affect the terminal state:

$$w_\pi = M b_{\pi_3} + b_{\pi_4}.$$

Writing $a = x_{\pi_3}$ and $b = x_{\pi_4}$ gives

$$J(\pi) = 2b^2 + ab + \frac{1}{4}a^2. \tag{30}$$

Thus J is constant over all prefixes sharing the same ordered two-request suffix, and hence $W_2 = 0$. The remaining suffix dependence is nevertheless not additive. Consider

$$\pi^{(1)} = (2,4,3,1), \quad \pi^{(2)} = (1,3,4,2), \quad \pi^{(3)} = (2,3,4,1), \quad \pi^{(4)} = (1,4,3,2).$$

Any assignment objective $f(\pi) = c + \sum_p C_{\pi_p,p}$ satisfies

$$f(\pi^{(1)}) + f(\pi^{(2)}) - f(\pi^{(3)}) - f(\pi^{(4)}) = 0, \tag{31}$$

because every request–position term cancels in this signed contrast. For the true objective, Equation 30 gives the corresponding values $41/4, 5, 5$, and $5/4$, whose signed contrast is 9. Therefore, if $\varepsilon = \max_\pi |J(\pi) - f(\pi)|$, the triangle inequality gives $9 \le 4\varepsilon$, and hence

$$\inf_{f \in \mathcal{A}} \max_\pi |J(\pi) - f(\pi)| \ge \frac{9}{4}. \tag{32}$$

Direct enumeration gives

$$\mathbb{E}_\pi J(\pi) = \frac{115}{24}, \qquad \min_\pi J(\pi) = 1, \qquad \Delta_J = \frac{91}{24},$$

so

$$D_{\mathcal{A}} \ge \frac{108}{91} > 1.$$

Thus exact two-request suffix localization can coexist with a substantial failure of position additivity.

Additivity does not imply short memory. For the reverse direction, take the one-dimensional instance

$$M = \frac{4}{5}, \qquad w_0 = 0, \qquad z_R = 1, \qquad b_1 = 1, \qquad b_2 = b_3 = b_4 = 0.$$

If request 1 occupies position p, then

$$J(\pi) = \left[\left(\frac{4}{5} \right)^{4-p} - 1 \right]^2 .$$

The objective therefore depends only on the request–position assignment, so $D_{\mathcal{A}} = 0$. Its four position values, in units of $1/15625$, are $3721, 2025, 625$, and 0, which gives

$$\Delta_J = \frac{6371}{62500}.$$

However, fixing an ordered suffix of length two does not determine the objective: when request 1 lies outside the suffix, it may still occupy prefix position 1 or 2. Consequently,

$$W_2 = \frac{3721 - 2025}{15625} = \frac{1696}{15625}, \qquad \frac{W_2}{\Delta_J} = \frac{6784}{6371} > 1.$$

Hence exact position additivity can coexist with substantial residual dependence on the prefix.

A.4 From assignment to scalar sorting

Scope of the slice projection. Proposition 1 certifies position additivity through one specific route, an independent additive projection of each request-pair slice. A small residual is therefore sufficient for approximate position additivity but not necessary. A coefficient tensor can be linearizable by a construction this projection does not find, so a large residual leaves the case uncertified rather than settled.

How tight is the sorting certificate? Equation 10 is a sufficient bound, so a large value leaves a case uncertified rather than unstructured. Table 3 compares best-family error, the specific sorting surrogate, its certificate, and realized decision regret on the complete $k = 4$ surfaces. The certificate is conservative where the family fits. The worst softmax cases nevertheless lose more than the random-order gap, whereas the reported SalUn cases do not; median performance therefore does not make sorting uniformly safe.

Double centering does not change the decision, only what the bound charges for it. Charging the position-only effect to the residual inflates the certificate by a median of 81% across the 95 complete surfaces the audit refits, with a range of 5 to 99%, while leaving every selected order, every realized regret, and every centered score unchanged.

Table 3: **The sorting certificate is conservative where the family fits.** Entries are medians with observed $[\min, \max]$ across the complete $k = 4$ surfaces of each setting, all normalized by the random-order gap. $D_{\mathcal{A}}$ is the best assignment family's uniform error, the sorting error is that of the surrogate $sq^\top$ itself, the certificate is the right-hand side of Equation 10, and the realized regret is what the sorted order loses on the complete table.

Setting	Inst.	$D_{\mathcal{A}}$	Sorting error	Certificate	Realized regret
		0.010	0.059	0.080	0.000
Frozen ridge, vision	15	[0.001, 0.066]	[0.015, 0.266]	[0.021, 0.375]	[0.000, 0.000]
		0.018	0.062	0.082	0.000
Frozen ridge, AG News	15	[0.002, 0.068]	[0.017, 0.319]	[0.023, 0.446]	[0.000, 0.092]
		0.022	0.074	0.116	0.000
Frozen ridge, 20 Newsgroups	15	[0.002, 0.126]	[0.015, 0.321]	[0.022, 0.473]	[0.000, 0.000]
		0.035	0.081	0.130	0.000
Direct softmax head	15	[0.005, 0.634]	[0.017, 2.616]	[0.021, 4.005]	[0.000, 1.248]
		0.065	0.584	0.872	0.009
Factorized softmax head	15	[0.010, 0.686]	[0.117, 2.523]	[0.184, 4.059]	[0.000, 1.789]
		0.236	1.867	2.672	0.122
NegGrad+	10	[0.119, 0.391]	[1.189, 2.998]	[1.807, 4.966]	[0.000, 0.309]
		0.673	1.734	2.287	0.322
SalUn	10	[0.215, 1.030]	[1.225, 2.686]	[1.655, 4.297]	[0.000, 0.775]

Linear-position projection proofs. For $p = 1, \dots, k$, let $q_p = p - (k+1)/2$, so $\|q\|_2^2 = k(k^2 - 1)/12$, and let $H = I - \frac{1}{k}\mathbf{1}\mathbf{1}^\top$. Writing a for the row means of C, b for its column means and m for its grand mean,

$$C = HCH + a\mathbf{1}^\top + \mathbf{1}b^\top - m\mathbf{1}\mathbf{1}^\top,$$

and for any permutation $\Phi_{a\mathbf{1}^\top}(\pi) = \sum_i a_i$, $\Phi_{\mathbf{1}b^\top}(\pi) = \sum_p b_p$ are constants, because each request and each position is used exactly once. Comparing permutations under C is therefore the same as comparing them under $\bar{C} = HCH$. Write $\bar{C} = P + \bar{\Xi}_q$ with $P = sq^\top$ and $\bar{\Xi}_q\mathbf{1} = \bar{\Xi}_q q = 0$, which also leaves $\bar{\Xi}_q$ column-centered since $\bar{C}$ is. For any permutation π,

$$\Phi_P(\pi) = \sum_p s_{\pi_p} q_p.$$

Because q_p is strictly increasing, the rearrangement inequality shows that sorting s_i nonincreasingly minimizes Φ_P. This proves exact scalar collapse when $\bar{\Xi}_q = 0$.

The centering never loosens the certificate. If $C = a\mathbf{1}^\top + sq^\top + \Xi_q$ is the one-sided projection, then applying H on both sides gives $\bar{C} = (Hs)q^\top + H\Xi_q H$, since $Hq = q$ and $\mathbf{1}^\top H = 0$. As $\bar{\Xi}_q$ is the residual of the best such fit and H is an orthogonal projection, $\|\bar{\Xi}_q\|_F \le \|H\Xi_q H\|_F \le \|\Xi_q\|_F$. The two projections also induce the same ordering, because they differ in s only by a constant shift, which the rearrangement inequality ignores.

For the approximate result, let π_C minimize $\Phi_{\bar{C}}$ and π_s minimize Φ_P. Then

$$\Phi_{\bar{C}}(\pi_s) - \Phi_{\bar{C}}(\pi_C) \le \Phi_{\bar{\Xi}_q}(\pi_s) - \Phi_{\bar{\Xi}_q}(\pi_C) \le \operatorname{range}_\pi \Phi_{\bar{\Xi}_q}(\pi).$$

If X_π is the permutation matrix selecting the entries summed by π, then

$$\Phi_{\bar{\Xi}_q}(\pi) - \Phi_{\bar{\Xi}_q}(\sigma) = \langle \bar{\Xi}_q, X_\pi - X_\sigma \rangle \le \|\bar{\Xi}_q\|_F \|X_\pi - X_\sigma\|_F \le \sqrt{2k}\,\|\bar{\Xi}_q\|_F.$$

The last inequality uses that two permutation matrices differ in at most $2k$ unit entries. This bounds the regret of π_s against the assignment objective Φ_C. Adding the two uniform errors between J and Φ_C, one at the selected order and one at the true optimum, gives Equation 10 with $\varepsilon_{\mathcal{A}}$ in place of the surrogate mismatch. Chaining part (a) supplies $\varepsilon_{\mathcal{A}} \le \varepsilon_{\mathrm{surr}} + \sqrt{\binom{k}{2}}\,\|\Xi\|_F$, since the projected assignment differs from the shared quadratic surrogate by the interaction residual and the surrogate differs from J by $\varepsilon_{\mathrm{surr}}$. The alternative entrywise bound is $\operatorname{range}_\pi \Phi_{\bar{\Xi}_q}(\pi) \le 2k\|\bar{\Xi}_q\|_\infty$.

B EXPERIMENTAL PROTOCOLS

This section defines how each experimental setting is executed and which executions count as valid. A *setting* fixes a policy, model, and data configuration, while an *instance* additionally fixes the requests, checkpoint, reference, and request-bound randomness. Permutations are deletion orders within an instance. Complete permutation surfaces support uniform statements over S_k, whereas identification and held-out splits support only statements on their declared samples. Frozen-head objectives are deterministic in double precision, and fixed-randomness repeats reproduce the full-network terminal objectives exactly. Results are reported in Appendices C and D.

Table 4 records which full-network analyses reuse an instance. Seed labels are local identifiers: the same number at different queue lengths does not denote the same request set.

Table 4: **Full-network instance correspondence.** "Main" means the complete $k = 4$ cohort used for structural summaries.

Analysis	Instances	Relation to main $k = 4$ cohort
NegGrad+ budget development	seeds 0–2, $k = 4$	Same requests; alternate budgets
NegGrad+ main	seeds 0–9, $k = 4$	Complete 24-order surfaces
SalUn pilot/replication	seeds 0–2 / 3–9, $k = 4$	Matched request partitions by seed
Composition	seeds 0–2, $k = 5$	New five-request queues
Held-out identification	seeds 0–2, $k = 8$	New eight-request sampled designs
Second tape / reference-free	seeds 0–2, $k = 4$	Same requests/reference; separate 10,000-audit replays
Pooled policy control	seeds 0–9, $k = 4$	Reuse NegGrad+ main instances

Sampled versus uniform claims. When S_k is too large to enumerate, the largest residual observed on a sampled evaluation set is only a sample-level statistic. It does not certify the maximum over all permutations. We therefore restrict sampled errors to their declared evaluation sets. Uniform quantities such as $D_{\mathcal{A}}$ and $D_{S,L}$ inherit the uniform regret guarantee only when their underlying error is controlled uniformly. At large k, such uniform control comes from Section 4, not from treating a sampled maximum as exhaustive.

Information used by each construction. The structural constructions use different information regimes. A *dynamics-based* construction follows directly from the update equations. A *reference-assisted* construction uses terminal objectives together with a retraining reference to fit or evaluate a structural family. A *probe-based* construction instead uses the base model, request data, retained-data derivatives, and declared singleton probes. Constructions fixed on development executions before a new queue are prospective on that queue. Complete-table fits are retrospective.

We separate reference generation, identification executions, offline fitting and solving, and candidate execution. One terminal-order query executes the full k-request sequence and evaluates its terminal objective. Identification executions are disjoint from held-out or newly executed evaluation orders. Table 5 reports counts, not FLOPs or wall-clock time; those depend on batch composition, forward/backward work, and evaluation and were not benchmarked here.

Table 5: **Information and execution costs by construction.** "Seq." counts terminal-order executions. Parentheses give optimizer updates when they follow directly from the protocol. Reference-assisted workflows include reference generation and are not claimed to be cheaper than retraining.

Construction	$T_{\text{reference}}$	$T_{\text{identification}}$	$T_{\text{fit/solve}}$	$T_{\text{candidate}}$
Ridge sorting	Exact retain solve	k seq.	Least squares, sort	1 seq.
Ridge assignment	Exact retain solve	$(k-1)^2+1$ seq.	Minimax, Hungarian	1 seq.
NegGrad+ $k=8$	Two retain-only runs	112 seq. (10,752)	Minimax, solve	1 seq. (96)
Head reconstruction	Exact retain solve	0 seq.; k probes	Construct, solve	1 seq.

B.1 Frozen heads

Ridge provides an exact retained reference for measuring endpoint prediction error and ordering regret. CIFAR-10 uses normalized penultimate features from a public ImageNet-pretrained ResNet-18 and a two-output ridge head with bias and $\lambda = 0.01$. AG News and 20 Newsgroups use mean-pooled frozen DistilBERT embeddings, a shared data-independent 256-dimensional projection, and per-example normalization. The protocol fixes request size, train/validation splits, and the retained reference before selection.

B.2 Full-network NegGrad+

Terminal comparison map. For each main-cohort instance, the first 2,000 examples of the fixed CIFAR-100 test split form the audit set; the remainder is used only for reported accuracies. The classifier predicts the 20 CIFAR-100 superclasses. If $z_w(x) \in \mathbb{R}^{20}$ denotes its logits, the comparison map subtracts the mean across the 20 coordinates separately for each example. The terminal objective is

$$j(w) = \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} \left\| z_w(x) - \overline{z}_w(x)\mathbf{1} - \left(z_R(x) - \overline{z}_R(x)\mathbf{1}\right) \right\|_2^2 .$$

Here z_R is the pointwise mean of two models trained from the declared initialization on the final retained training set. The audit examples and reference are fixed across every order of an instance. Only training data are used for request updates; the test audit set is read at terminal evaluation. The second-tape and reference-free controls reuse the seed-indexed requests and reference files but come from separate replays evaluated on all 10,000 test examples. They are therefore distinct policy–objective instances from the main 2,000-example cohort, even when their request labels match.

Execution. Each request uses 12 NegGrad+ steps with learning rate 3×10^{-3} and minibatch size 256. The loss weights retained and forget cross-entropy by 0.95 and -0.05, respectively. Each

request starts a fresh SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}. The updated model weights and batch-normalization state pass to the next request. Retain batches use the run's CIFAR augmentation and forget batches do not.

Operating point and validity. The 12-step operating point was fixed by the preceding pilot calibration using retained utility and order-mean progress toward the retraining reference; no ordering statistic or deleted-group screen selected it. The four-budget $\{3, 12, 24, 48\}$ sweep was run afterward with model, requests, reference, and request-bound randomness held fixed. Under the three screens reported here, 12 steps is the only tested budget that passes on all three development instances. The deleted-group screen is therefore a retrospective audit, not a backdated selection rule. Per-instance values for the retain-weight sweep at fixed step size are in the repository.

Structure across the budget sweep. The same four budgets also carry structural coordinates, which the selection did not look at (Table 6). Nominal update count does not order suffix attenuation monotonically, and assignment error has no corresponding trend. The effective suffix depth changes across the sweep, but all alternative budgets fail at least one deletion-quality screen. We therefore treat this as operating-region sensitivity and ground the causal mechanism claim on the damped-Newton intervention of Appendix C.2, where contraction is fixed before execution.

Table 6: **Structural coordinates across the update-strength sweep.** Entries are medians with observed $[\min, \max]$ over the same three development instances, on complete $k = 4$ surfaces. "Valid seeds" counts the instances passing all three reported screens. History retention is the measured matched-history attenuation, and $D_{S,1}$ and $D_{S,2}$ are the suffix errors at one and two positions.

Steps/request	Valid seeds	History retention	$D_{S,1}$	$D_{S,2}$	$D_{\mathcal{A}}$
		0.318	1.362	0.515	0.443
3	0/3	$[0.312, 0.321]$	$[0.843, 1.399]$	$[0.341, 0.531]$	$[0.260, 0.512]$
		0.065	0.598	0.221	0.302
12	3/3	$[0.059, 0.066]$	$[0.573, 0.883]$	$[0.204, 0.286]$	$[0.265, 0.441]$
		0.118	0.394	0.149	0.221
24	2/3	$[0.108, 0.127]$	$[0.363, 0.417]$	$[0.117, 0.202]$	$[0.166, 0.229]$
		0.278	0.993	0.673	0.256
48	0/3	$[0.225, 0.312]$	$[0.840, 1.167]$	$[0.497, 0.892]$	$[0.198, 0.378]$

At step t the retain loader is the complement of every request deleted through that step, $R_t = \mathcal{D} \setminus \bigcup_{s \le t} F_{\pi_s}$, so deleted samples are never reintroduced as retain data. All ten reported instances satisfy the retained-utility criterion, worst-order retain accuracy at least $\min(A_{\text{base}}, A_{\text{retrain}}) - 0.05$, and the reference-progress criterion, order-mean terminal output closer to the retraining reference than the base output. The deleted-group accuracy screen is reported as an additional retrospective check. All ten pass, with a smallest decrease of 0.120, so the measured rebound reflects recovery under later updates.

B.3 Full-network SalUn

Execution. SalUn is executed from the authors' implementation. Each request uses 12 steps with learning rate 10^{-4} and minibatch size 256, a fresh SGD optimizer per request with momentum 0.9 and weight decay 5×10^{-4}, retain-only augmentation, and batch-normalization in training mode. The updated weights and batch-normalization state pass to the next request, as for NegGrad+. At every request, the mask is recomputed from the current model by accumulating absolute gradients of the negated forget loss and globally selecting the top 50% of parameter coordinates. The authors' random-label update is then applied only on those coordinates. Its NumPy and PyTorch draws are seeded from the request-bound generator; forget examples are unaugmented, and a retained sample fills the fixed 12×256-example pass.

Operating point and validity. SalUn uses the 12-step budget fixed previously by the full-network protocol rather than a SalUn-specific sweep. At this operating point all ten instances satisfy the same retained-utility, reference-progress and deleted-group checks, with worst-order retain accu-

racy 0.847–0.898, reference-distance ratios of 0.631–0.753, and deleted-group accuracy falling by 0.114–0.175 against the screen's 0.05. No structural diagnostic enters any of these checks.

C MECHANISMS AND STRUCTURAL DIAGNOSTICS

This appendix carries the evidence for where the observed reductions come from and the controls that separate them from family flexibility. The ledgers below record the diagnostics summarized by Figure 2.

Table 7 compares the structural regimes on surfaces enumerated in full. The settings do not share a single sufficient family: position additivity dominates the frozen-head surfaces, while suffix localization distinguishes the complete full-network surfaces from the ridge extension. Objectives and normalizations are setting-specific, so entries compare instances within a setting rather than across settings.

For complete permutation surfaces, $D_{\mathcal{A}}$ is the exact best assignment-family error over all $k!$ orders and D_S the exact suffix family error. The two sampled designs carry a different quantity and are reported with the identification results in Appendix D.

Table 7: **Structural regimes on the completely enumerated surfaces.** Entries are medians across instances with observed $[\min, \max]$. *Gap* is $100\Delta_J/\bar{J}$, every error is exact over all $k!$ orders, and the last two columns count the instances whose assignment and suffix errors fall below one. All suffix fits condition on the final two requests.

Setting	Req.	Inst.	Gap (%)	$D_{\mathcal{A}}$	D_S	A pass	S pass
			0.878	0.018	0.562		
Ridge head	4	45	[0.063, 11.659]	[0.001, 0.126]	[0.427, 0.658]	45/45	45/45
			0.076	0.035	0.589		
Direct head	4	15	[0.003, 1.055]	[0.005, 0.634]	[0.492, 1.700]	15/15	13/15
			0.811	0.065	0.547		
Factorized head	4	15	[0.073, 1.446]	[0.010, 0.686]	[0.383, 1.303]	15/15	14/15
			0.838	0.017	1.290		
Ridge head	8	3	[0.824, 0.996]	[0.015, 0.021]	[1.287, 1.318]	3/3	0/3
			9.285	0.236	0.148		
NegGrad+	4	10	[3.407, 15.201]	[0.119, 0.391]	[0.055, 0.477]	10/10	10/10
			9.647	0.186	0.238		
NegGrad+	5	3	[3.660, 13.805]	[0.150, 0.771]	[0.227, 0.582]	3/3	3/3
			8.488	0.673	0.260		
SalUn	4	10	[5.831, 13.501]	[0.215, 1.030]	[0.185, 0.510]	9/10	10/10

C.1 POSITION ADDITIVITY FROM UPDATE DYNAMICS

Cumulative-response construction. Let u_i be the singleton displacement and set $M = (I - \eta H_R)^T$ for a budget of T steps. A request deleted at position p remains absent in every later stage, giving the shared-curvature response

$$e_{i,p} = \sum_{t=0}^{k-p} M^t u_i, \qquad p = 1, \ldots, k. \tag{33}$$

For $1 \le p \le k-2$, direct subtraction yields

$$e_{i,p+2} - 2e_{i,p+1} + e_{i,p} = M^{k-p-1}(M - I)u_i. \tag{34}$$

The telescoping identity for M implies

$$\|M - I\| \le T\eta\|H_R\| \max\{1, \|I - \eta H_R\|\}^{T-1}.$$

If the singleton gradient norm is at most G_i along its T steps, then $\|u_i\| \le T\eta G_i$ and

$$\|e_{i,p+2} - 2e_{i,p+1} + e_{i,p}\| \le T^2\eta^2\|M\|^{k-p-1}\|H_R\|G_i \max\{1, \|I - \eta H_R\|\}^{T-1}. \tag{35}$$

Thus the cumulative response is affine in position up to second order in the step size for fixed T and k.

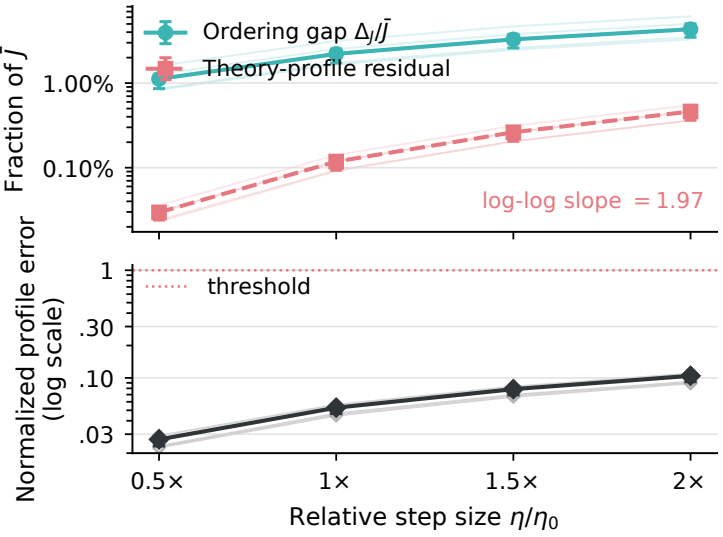


Figure 5: **Positional-profile error grows at the second-order rate predicted by the update expansion.** The upper panel shows absolute order gap and profile residual against learning rate, and the lower panel the same residual normalized by the gap.

From the retained recurrence to the terminal objective. The executed ridge policy minimizes a mean retained loss, so its normalization changes with the stage. Let the full training set have size N, let every request have size m, and let $n_t = N - tm$ after stage t. If r_i is the sum of the unregularized per-example gradients in request i, expansion of the actual T-step recurrence at the full-data optimum gives, uniformly over the finite set of orders,

$$w_\pi(\eta) = w_0 + \eta c + T\eta \sum_{p=1}^{k} h_p r_{\pi_p} + O(\eta^2), \qquad h_p = \sum_{t=p}^{k} \frac{1}{n_t}, \tag{36}$$

where c is independent of π. Expanding the squared-distance objective around w_0 therefore gives a scalar request–position objective with profile h plus an $O(\eta^2)$ remainder. The exact profile is not fixed-q because n_t changes. For fixed k, $h_p = (k-p+1)/N + O(m/N^2)$. Consequently, projection onto fixed-q has both a small first-order normalization term and a second-order dynamics term. If the first-order order variation is nonzero, $\Delta_J = \Theta(\eta)$, while D_{sort} need not scale quadratically. This derivation is specific to the matched ridge recurrence and does not convert the finite-sample sweep into an asymptotic certificate.

Matched step-size intervention. We fixed $T = 4$ gradient steps per request, the five held-out CIFAR-10 request partitions, the base ridge head, and every exact retained-data reference, and varied only the learning rate over $0.5\eta_0$, η_0, $1.5\eta_0$, and $2\eta_0$, where $\eta_0 = 0.526$. All 20 matched configurations contained 65 independently replayed prefixes and 24 finite terminal objective values. The largest product of learning rate and retained Hessian eigenvalue was $1.801 < 2$. The recurrence-specific residual has fixed-effect log–log slope 1.971. Its near-quadratic growth is consistent with the positional-curvature calculation in Equation 35 and the second-order remainder in Equation 36.

Figure 5 shows every configuration of that sweep, in absolute terms and normalized by each configuration's gap. Here h is the recurrence-specific profile in Equation 36. Its residual is smaller than the fixed linear-position profile in all 20 matched partition–step-size configurations.

Shared-response quadratic closure. We reused the singleton displacements and shared response from the pre-outcome analysis, and used the exact retained reference only for this retrospective comparison. The linear surrogate scores the predicted displacement by its first-order inner product with $w_0 - w_R$. The quadratic surrogate instead evaluates the exact squared norm of that same predicted endpoint, thereby adding precisely the interactions in Equation 8. Nothing about the

response model, requests, or reference changed between the two columns. This comparison tests the terminal-objective consequence of the derived shared response, but does not assert that the actual ridge execution has one fixed response matrix. Its retained Hessian changes with the deleted request set. Table 8 reports the two scorings.

Table 8: **Quadratic terminal scoring captures interactions missed by linear scoring.** Entries are medians over five partitions and three update budgets. Errors are normalized as $2\varepsilon/\Delta_J$.

Domain	Linear terminal	Quadratic terminal	Residual ratio
Vision	0.145	0.006	0.041
AG News	0.141	0.006	0.043
20 Newsgroups	0.212	0.009	0.045

C.2 Suffix localization from contraction

Exact contraction under damped Newton. Both frozen-ridge update rules are affine in the weights, so the Lipschitz constants that Proposition 2(a) multiplies are not estimated. Writing $H_t = X_r^\top X_r/n_r + \lambda I$ for the retained Hessian and $g = H_t w - b_t$ for the gradient, T gradient steps give $M_t = (I - \eta H_t)^T$, while the damped Newton step $w \mapsto w - \delta H_t^{-1} g$ gives

$$M_t = I - \delta H_t^{-1} H_t = (1 - \delta) I.$$

The Hessian cancels, so at $\delta = 0.5$ the ordered-suffix products are $\kappa_1 = 0.5$ and $\kappa_2 = 0.25$ for every partition, request and suffix. Each step's Jacobian depends only on which requests have been deleted when it runs, and for a fixed ordered suffix that set does not depend on the prefix, which is why the suffix map is well defined. Measuring the realized attenuation $\|U_s(u) - U_s(v)\|/\|u - v\|$ of every pair of prefixes entering the same suffix reproduces these constants exactly under Newton and never exceeds them under gradient descent, where κ_1 is 0.9947 at one step and 0.9190 at sixteen.

Matched gradient–Newton intervention. Development used three instances. The prediction was then frozen before any held-out Newton table existed. Two things were frozen, and they carry different weight. Reproducing the analytic constants above is an implementation check. The Jacobian is $(1 - \delta)I$ by construction, so a run disagreeing with it indicates a fault in the harness. The prespecified structural prediction is separate and falsifiable, and concerns the relative movement of the suffix errors against their equal-capacity prefix and middle controls as the contraction changes. The bound of Proposition 2(a) also involves R_s and d_s, and it predicts nothing about the direction the controls take. The five held-out partitions are matched to the archived gradient instances on checkpoint, feature extractor, request sample ids, λ, learning rate and step scale. Only the preconditioner differs. The held-out intervention supports the prespecified separation: stronger analytic contraction favors the suffix family relative to its matched positional controls. The archived gradient surfaces do not show that separation, so the result concerns where positional memory resides rather than a uniform reduction in family error (Table 9).

Table 9: **Matched gradient descent and damped Newton differ in which positional families suffice.** Entries are medians with observed $[\min, \max]$ across the five held-out partitions. Scale is $100\Delta_J/\bar{J}$.

Update rule	Scale (%)	$D_{\mathcal{A}}$	$D_{S,1}$	$D_{S,2}$	$D_{P,2}$
Gradient descent	0.565	0.003	1.095	0.543	0.559
	$[0.416, 0.806]$	$[0.001, 0.005]$	$[1.000, 1.169]$	$[0.496, 0.579]$	$[0.512, 0.599]$
Damped Newton	31.684	0.111	0.597	0.143	1.488
	$[29.902, 47.809]$	$[0.052, 0.283]$	$[0.555, 0.768]$	$[0.133, 0.203]$	$[1.268, 1.605]$

Damping sweep. The damped Newton Jacobian is $(1-\delta)I$, so the one-step contraction $\kappa_1 = 1-\delta$ is set by the damping and known before execution. The same five partitions were re-executed at three further damping values, changing nothing else. Those three were fixed before their outcomes were

observed, and $\delta = 0.5$ had been evaluated previously. Table 10 reports every partition at every damping.

Table 10: **Analytically stronger Newton contraction progressively favors suffix-localized structure.** Entries are medians with observed $[\min, \max]$ across the same five instances, with both scale columns in percent and $j_0 = j(w_0)$ the objective at the undeleted state.

κ_1	δ	$\Delta_J/\bar{J}$	Δ_J/j_0	$D_{\mathcal{A}}$	$D_{S,1}$	$D_{S,2}$	$D_{P,2}$	$D_{M,2}$
		8.263	5.038	0.022	1.063	0.487	0.720	1.755
0.90	0.10	[6.563, 11.721]	[3.999, 7.118]	[0.011, 0.059]	[0.951, 1.105]	[0.431, 0.495]	[0.614, 0.740]	[1.559, 1.840]
		19.492	5.769	0.060	0.910	0.347	0.991	1.808
0.75	0.25	[16.593, 28.456]	[4.890, 8.202]	[0.028, 0.159]	[0.819, 1.015]	[0.312, 0.368]	[0.810, 1.016]	[1.602, 1.957]
		31.684	2.837	0.111	0.597	0.143	1.488	2.051
0.50	0.50	[29.902, 47.809]	[2.561, 3.885]	[0.052, 0.283]	[0.555, 0.768]	[0.133, 0.203]	[1.268, 1.605]	[1.742, 2.182]
		37.595	0.710	0.093	0.247	0.035	1.869	2.059
0.25	0.75	[34.761, 56.612]	[0.555, 0.905]	[0.047, 0.221]	[0.222, 0.408]	[0.032, 0.067]	[1.795, 2.124]	[1.946, 2.325]

The two scale columns move differently: stronger damping sharpens the structure of the ordering problem while eventually shrinking what ordering is worth relative to the fixed starting distance. The intervention therefore separates structural simplicity from decision scale rather than supporting a claim about deployment value.

The realized contractions match their analytic values on every instance–damping pair. Stronger damping separates the suffix families from the equal-capacity positional controls, whereas assignment error is non-monotone. Absolute suffix width is non-monotone as well, so the separation is not an effect of the shared normalization.

Matched-history attenuation. We compare histories that delete the same two requests in opposite orders and then apply the same two-request suffix with request-bound randomness (Table 11).

Table 11: **Matched histories contract in output space and in the terminal objective under a common suffix.** Statistics aggregate 120 matched paths from ten request sets. Output separation is RMS logit distance, and alignment is the absolute objective difference divided by its Cauchy bound $2\|m - z_R\|_{\mathrm{RMS}}\|d\|_{\mathrm{RMS}}$, whereas j_{RMS} is a mean squared distance.

Common suffix	Output RMS	$\lvert j_{\mathrm{RMS}}(z) - j_{\mathrm{RMS}}(z')\rvert$	Cauchy bound	Alignment
Before	2.025	0.621	12.664	0.052
After one request	0.211	0.079	1.292	0.069
After two requests	0.130	0.045	0.760	0.058

Let $d_z = 20|\mathcal{A}|$ be the number of centered-logit coordinates and define $j_{\mathrm{RMS}}(z) = d_z^{-1}\|z - z_R\|_2^2$. The main full-network objective is $j(z) = 20 j_{\mathrm{RMS}}(z)$: it averages squared Euclidean distance across audit examples but sums over the 20 superclass coordinates. This constant factor does not affect attenuation ratios or alignment, while the absolute objective differences and bounds in Table 11 use per-coordinate MSE units. For $d = z - z'$ and $m = (z + z')/2$, the exact identity $j_{\mathrm{RMS}}(z) - j_{\mathrm{RMS}}(z') = (2/d_z)\langle m - z_R, d\rangle$ holds on every pair. From the initial prefix to the completed common suffix, median output retention is 6.62% and median absolute-objective-difference retention is 6.69%. Their agreement, together with the stable alignment fraction, attributes the smaller objective width primarily to output attenuation. Because d is observed, this is a finite-execution consistency analysis. Run on the three matched pilots per policy that store prefix states, the same comparison leaves a median objective-difference retention of 6.9% under NegGrad+, observed range $[6.5, 7.2]$, and 10.2% under SalUn, observed range $[10.1, 10.5]$. The next experiment evaluates held-out transitions.

Conditioning versus truncation. Proposition 2 states two rates at a fixed suffix depth, $O(\rho^L)$ for truncation and $O(\rho^{2L})$ for conditioning, and this is where the two are measured against each other on executed surfaces. The comparison needs a design the conditional family cannot saturate. At $k = 4$ with $L = 2$ each suffix leaves two prefix orders that an assignment fits exactly, so an error of

zero there would report capacity. We therefore use the complete $k = 5$ NegGrad+ surfaces at $L = 1$, where each of the five suffixes leaves 24 prefix orders against ten free parameters, 50 in rank across the 120 orders.

Conditioning is more accurate than either nested comparator on every instance. This checks the construction at one depth on a design where the conditional family has ample room to miss. Establishing the exponent would require varying L or ρ. Because the three nested complete-table families have unequal rank, the main text separately compares first-, third-, and final-position conditioning at matched rank 50. Its four folds are fixed from cyclic relabeling orbits before outcomes are read; every 90-order training design has full rank and balanced conditioned-request counts. For fold f, define the training-only center and scale

$$\overline{J}_f^{\mathrm{train}} = \frac{1}{|T_f^{\mathrm{train}}|} \sum_{\pi \in T_f^{\mathrm{train}}} J(\pi), \qquad \Delta_f^{\mathrm{train}} = \overline{J}_f^{\mathrm{train}} - \min_{\pi \in T_f^{\mathrm{train}}} J(\pi),$$

and $\widetilde{J}_f^{\mathrm{train}}(\pi) = (J(\pi) - \overline{J}_f^{\mathrm{train}})/\Delta_f^{\mathrm{train}}$. Each family $\mathcal{F}_r$ is fitted by

$$\widehat{g}_{r,f} \in \arg\min_{g \in \mathcal{F}_r} \max_{\pi \in T_f^{\mathrm{train}}} |\widetilde{J}_f^{\mathrm{train}}(\pi) - g(\pi)|, \qquad \widehat{J}_{r,f}(\pi) = \overline{J}_f^{\mathrm{train}} + \Delta_f^{\mathrm{train}} \widehat{g}_{r,f}(\pi).$$

If the minimax solution is nonunique, a pre-fixed tie rule minimizes the prediction norm $\sum_{\pi \in S_5} g(\pi)^2$; this uses the complete permutation design but no held-out outcome. Only evaluation uses the complete-surface headroom Δ_J:

$$D_{r,f}^{\mathrm{test}} = \frac{2}{\Delta_J} \max_{\pi \in T_f} |J(\pi) - \widehat{J}_{r,f}(\pi)|.$$

Thus the fitted original-scale predictions depend only on the training outcomes. The final-position model wins both controls in every instance–fold comparison. This supports positional specificity, not the affine decay exponent.

C.3 Separation of structural families

Nested position families. Table 12 compares three nested position families. The linear family fixes the shared positional profile, the monotone family learns that profile jointly with one score per request, and assignment allows an unrestricted request–position cost matrix. The monotone fit is retrospective and non-convex. The reported value is the best of the two search procedures fixed in advance, which agree to 5×10^{-5} on every surface.

Table 12: **Nested position families separate fixed-profile sorting, learned-profile sorting, and unrestricted assignment.** D_{lin} fixes a shared linear positional profile, D_{mono} learns a shared monotone profile, and $D_{\mathcal{A}}$ is unrestricted. R is the normalized regret of the selected order. The last column counts instances meeting, on, and failing the criterion for D_{mono}, with the boundary band $[0.95, 1.05]$.

Setting	Instances	D_{lin}	D_{mono}	$D_{\mathcal{A}}$	R_{mono}	$R_{\mathcal{A}}$	m/b/f
Frozen ridge, vision	15	0.056	0.011	0.010	0.000	0.000	15/0/0
Frozen ridge, AG News	15	0.055	0.026	0.018	0.000	0.000	15/0/0
Frozen ridge, 20 Newsgroups	15	0.061	0.055	0.022	0.000	0.000	15/0/0
Direct softmax head	15	0.071	0.068	0.035	0.000	0.000	14/0/1
Factorized softmax head	15	0.545	0.305	0.065	0.009	0.000	13/0/2
NegGrad+	10	1.582	0.281	0.236	0.122	0.064	10/0/0
SalUn	10	1.346	0.857	0.673	0.348	0.256	6/1/3

The learned profile nearly closes the gap to assignment for NegGrad+, whereas SalUn exhibits substantially greater instance-to-instance variation. The apparent failure of scalar sorting under a fixed linear profile therefore does not extend to the more general shared-profile family.

Matched parameterization intervention. We compare direct logistic regression with its function-matched factorization $W = VU$ on held-out request unions. All configurations passed the frozen validity criteria. Factorization increased the paired order range in 14/15 instances and changed the optimal order in 8/15. Assignment met $D_{\mathcal{A}} < 1$ in every instance and was substantially tighter than scalar sorting. The initial predictor and executed objective were matched, so the contrast isolates the update parameterization.

Within-suffix interaction. The suffix criterion can be met for two different reasons. Either the final two positions contribute additively, in which case the suffix structure is the tail of an assignment cost matrix, or the two surviving requests interact. Table 13 separates them by restricting the same family to $c + u_{\pi_{k-1}} + v_{\pi_k}$. The additive family is contained in the general one, so the difference is exactly the within-suffix interaction, and it is contained in assignment, so $D_{\mathcal{A}}$ lower-bounds it.

Table 13: **General suffix localization can retain interactions that an additive suffix cannot represent.** Entries are medians across instances with observed $[\min, \max]$ on complete full-network permutation surfaces. All suffix fits condition on the final two requests.

Setting	Instances	D_S	D_S^{add}	$D_{\mathcal{A}}$
NegGrad+	10	0.148 $[0.055, 0.477]$	0.293 $[0.146, 0.489]$	0.236 $[0.119, 0.391]$
NegGrad+ ($k = 5$)	3	0.238 $[0.227, 0.582]$	0.238 $[0.230, 0.924]$	0.186 $[0.150, 0.771]$
SalUn	10	0.260 $[0.185, 0.510]$	0.746 $[0.335, 1.139]$	0.673 $[0.215, 1.030]$

SalUn shows a material gap between general and additive suffix fits, unlike NegGrad+. Its localized dependence therefore retains interactions that cannot be represented as separate costs for the final two positions.

Equal-capacity positional controls. To test whether suffix fits are favored merely by flexibility, we compare them with arbitrary functions of the requests occupying the first L positions or a fixed contiguous middle block. Each family has $k!/(k-L)!$ groups, and all widths use the same Δ_J normalization (Table 14).

Table 14: **Equal-capacity positional controls distinguish suffix localization from generic low-dimensional dependence.** Prefix, middle, and suffix families condition on the same number of ordered positions and use the same Δ_J normalization.

Instances	L	Prefix	Middle	Suffix
$k = 4$ (10)	1	2.137	2.192	0.368
	2	1.926	2.096	0.148
$k = 5$ (3)	1	2.254	2.244	0.331
	2	2.232	2.243	0.238

The matched controls favor suffix localization overall, although less uniformly than the damped-Newton intervention. This comparison isolates recency from generic conditioning capacity but does not construct an online suffix score.

Queue-length extension at $k = 5$. The extension keeps the NegGrad+ policy fixed and changes only the number of superclass requests from four to five. Each seed contains all 120 orders and 325 nonempty prefix executions, and Table 15 gives the resulting widths.

Table 15: **At $k = 5$, the general suffix error separates prefix memory from additive suffix structure.** All entries are normalized by the total ordering gap Δ_J.

Seed	Suffix L=1	General suffix L=2	Additive suffix L=2	Assignment
0	0.327	0.227	0.230	0.186
1	1.005	0.582	0.924	0.771
2	0.331	0.238	0.238	0.150

The extension supports suffix localization while leaving room for within-suffix interactions. The strict separation is supplied by Proposition 3; this experiment does not estimate a contraction parameter.

C.4 Nonlinear reach of the suffix bound

Direct-head analytic bound. Proposition 2(a) is the one quantitative result that covers updates which are not affine, so how much it delivers on an executed nonlinear head sets what we are entitled to say about the full-network policies. The direct head appends a constant feature for bias and regularizes the entire two-class weight matrix, including that coordinate. Each deletion transition uses one, four, or sixteen ordinary full-batch SGD steps with learning rate 0.05, mean cross-entropy, $\lambda = 10^{-4}$, and fresh optimizer state. For a retained design X, the softmax Hessian therefore lies between λI and $(\lambda + \|X\|_2^2/(2n))I$. We compute the resulting contraction factor separately for the two retained sets in every ordered suffix and use the exact operator norm of the validation design as L_Φ.

The comparison uses every one of the 15 matched direct-head instances. None is selected by its gap or by bound tightness. Median analytic two-stage contraction factors are 0.99999, 0.99996, and 0.99984 across the three budgets, and the corresponding finite-pair endpoint ratios are 0.996, 0.978, and 0.919. The latter are trajectory diagnostics rather than regional Lipschitz constants. The global objective bound holds for every suffix in every instance, and Table 16 reports how conservative it is on absolute and decision-normalized scales.

The analytic bound remains valid but conservative on the executed nonlinear heads. We therefore use it as a sufficient-condition check, ground the mechanism claim in the matched Newton intervention, and treat full-network attenuation as finite-execution evidence.

Table 16: **The analytic common-suffix bound holds on executed direct-head trajectories with roughly an order of magnitude to spare.** Entries are medians over five partitions. Curvature constants are global to each retained design, while reachable-state diameters and target radii are evaluated on the executed instances. $\mathcal{B}_2$ is the bound of Equation 11.

Steps	W_2	$\mathcal{B}_2$	$\mathcal{B}_2/W_2$	D_S	$\mathcal{B}_2/\Delta_J$
1	2.39×10^{-8}	1.68×10^{-7}	9.24	0.623	5.85
4	8.84×10^{-8}	6.66×10^{-7}	9.78	0.614	6.05
16	2.65×10^{-7}	2.51×10^{-6}	11.93	0.575	6.86

D Out-of-sample structural identification

This appendix asks whether structural families identified from limited information continue to predict orders that were not used to construct them. The budgets run from no terminal-order queries at all, through bases of size $O(k)$ and $O(k^2)$, to a balanced 112-query full-network design. Table 18 carries the structural diagnostics for the two settings whose order space is too large to enumerate.

D.1 Probe-based structural reconstruction

The lowest budget uses no terminal-order queries at all. The constructions in Figure 2b read the k singleton responses and the retained derivatives, and their values are reported on queues held out

of the earlier retained-loss development. The unary quadratic construction carries no cross-request interaction and is the one the interaction term separates from (Table 17). The comparative ordering of the four remains the same on the held-out queues, although their numerical values also change. Fitting the assignment family directly to the complete terminal-objective table instead of to singleton responses gives 0.019 and 0.017, so the response-level construction is the more accurate of the two here despite seeing no order. Errors are instance medians, only an order-independent constant is removed before aligning them, and no score scale is refitted. The comparison is reference-assisted and retrospective.

Table 17: **The four singleton-response constructions on the calibration queues.** Medians across instances of $2\varepsilon/\Delta_J$, with the mean normalized regret of the order each construction selects. Figure 2b reports the same constructions on the fresh-decision queues.

Construction	Reconstruction error	Selection regret
Linear	0.151	0.0051
Unary quad.	0.098	0.0005
Projected quad.	0.021	0.0001
Full quad.	0.006	0.0000

Table 18: **The same diagnostics on the two sampled designs.** Entries are medians across instances with observed $[\min, \max]$. Every error is fitted on the identification orders and evaluated on the declared held-out split, so it describes that split rather than S_k. The $k = 15$ design carries no suffix axis because W_2 was not estimated there.

Setting	Design	Req.	Inst.	Gap (%)	$D_{\mathcal{A}}^{\text{test}}$	D_{S}^{test}
Ridge head	197/500	15	3	2.651 [1.835, 2.785]	0.110 [0.101, 0.160]	—
NegGrad+	112/336	8	3	11.886 [11.024, 16.399]	0.446 [0.392, 0.520]	0.570 [0.551, 0.713]

D.2 Large-queue position-structure identification

Large-k structural identification. The large-queue study uses three fixed CIFAR-10 request partitions and the one-step retained-gradient policy with an exact retained reference. Pivoted QR selects the assignment basis before any objective value is observed. Because the basis is saturated, all structural evaluation uses disjoint orders, the remaining complete surface at $k = 8$ and a sampled set at $k = 15$, against a capacity-matched shuffle on the same evaluated orders and objective values (Table 19).

Assignment generalizes beyond its saturated basis, whereas the matched shuffle does not. Table 20 separately audits the order returned by solving each fitted $k = 8$ model against the complete surface. The recovery comes from extrapolating the fitted family rather than observing the returned order in the identification basis.

Table 19: **Identification budgets and held-out fit for the two queue lengths.** Medians across the three partitions. The held-out error is $2\varepsilon/\Delta$ on the evaluation orders, and the last three columns describe the assignment fit, which the sorting tiers do not report. Shuffle R^2 is the capacity-matched permutation of the same objective values. The k-order and assignment rows are the ones Figure 4a carries.

Req.	Basis	Orders	Held-out error	Evaluated	R^2	NRMSE	Shuffle R^2
8	Sorting, k	8	0.068	—	—	—	—
8	Sorting, $k{+}1$	9	0.066	—	—	—	—
8	Assignment	50	0.055	40,270	0.9997	0.017	−3.69
15	Sorting, k	15	0.107	—	—	—	—
15	Sorting, $k{+}1$	16	0.094	—	—	—	—
15	Assignment	197	0.110	500	0.9971	0.054	−10.61

Sorting bases for the same surfaces. The sorting comparison in Figure 4a chooses its identification orders by the same pivoted-QR rule applied to the sorting design alone. The basis is drawn from S_8 at $k = 8$, and at $k = 15$, where no complete surface exists, from the 197 executed identification orders, which asks whether fifteen of those queries would have served. At $k = 8$, the fitted score is minimized over all of S_8 and audited on the complete surface. At $k = 15$, it is minimized only over the disjoint 500-order evaluation sample, so the reported rank is not a global statement over S_{15}. The extra basis order does not change the qualitative comparison, while assignment remains the tighter description on the complete $k = 8$ surface.

Table 20: **True rank of the orders returned by the fitted $k = 8$ families.** Each fitted score is minimized over the complete permutation space; regret and uniqueness are then evaluated on the stored surface.

Instance	Family	Queries	True rank	R_{cand}	Unique
0	Sorting	8	1/40,320	0	Yes
0	Assignment	50	1/40,320	0	Yes
1	Sorting	8	1/40,320	0	Yes
1	Assignment	50	1/40,320	0	Yes
2	Sorting	8	1/40,320	0	Yes
2	Assignment	50	1/40,320	0	Yes

After fitting, we also solved each $k = 15$ assignment globally with the Hungarian algorithm and executed that one predicted order. None of the three orders had been evaluated during identification or sampled testing (Table 21).

Table 21: **The fitted $k = 15$ assignment minimizer improves on both identification and sampled evaluation orders when executed.** Values are squared distances to the exact retained ridge optimum. Improvement is measured relative to the mean sampled held-out objective.

Partition	Selected	Best identification	Best sampled	Improvement
0	2.2670×10^{-4}	2.2880×10^{-4}	2.2826×10^{-4}	3.32%
1	2.2730×10^{-4}	2.2873×10^{-4}	2.2796×10^{-4}	3.07%
2	2.2877×10^{-4}	2.3055×10^{-4}	2.3023×10^{-4}	2.46%

D.3 Full-network held-out prediction

Balanced sampled design at $k = 8$. We used a balanced 112/336 identification/held-out design in which every ordered request pair occurs equally often at each tested position pair. The nonoverlapping order sets are disjoint orbits of a fixed affine permutation group. This construction gives exact request-pair balance at each tested position pair.

For each position-pair family, the value of a request-pair group was the midpoint of its two identification executions. The assignment coefficients were instead obtained by a minimax fit using only the same 112 identification executions. No coefficient, offset, family, or orientation was adjusted using the held-out executions. All seeds used the same order split and request partition. Three seeds changed the model/execution instance. For a fitted family $\mathcal{F}$ with surrogate $\widehat{J}_{\mathcal{F}}$, the held-out error is

$$D_{\mathcal{F}}^{\text{test}} = \frac{2\max_{\pi\in\mathcal{T}} |J(\pi) - \widehat{J}_{\mathcal{F}}(\pi)|}{\Delta_{\text{sample}}},$$

where $\mathcal{T}$ is the held-out set and Δ_{sample} is the mean-to-minimum gap across all 448 sampled orders. This denominator is an evaluation scale, not an input to fitting. Each of the three 448-order seed tables passed the validity screen on all sampled orders.

All three positional families condition on two ordered positions and therefore have equal capacity. Their held-out comparison is reported in Table 22.

Table 22: **At $k = 8$, assignment and suffix structure generalize to held-out full-network orders while equal-capacity controls do not.** Each instance uses 112 identification and 336 held-out orders with exact pair balance. Reported errors are evaluated only on the held-out design, and the last column is the observed suffix error, the width normalized by the sampled gap.

Instance	Prefix	Middle	Suffix	Assignment	Observed D_S
0	2.775	3.578	0.551	0.392	0.300
1	2.592	2.743	0.570	0.520	0.376
2	4.505	4.505	0.713	0.446	0.418

D.4 Validation on unseen executions

Candidate transfer. We fixed each fitted family and its tie-breaking rule before executing the returned candidates: the original 112 identification executions, the minimax assignment and suffix fits, lexicographic tie breaking, and an ascending request-ID completion for the unspecified prefix. Executing the two candidates per instance under the original request-bound randomness, all six passed the validity screen (Table 23).

Candidate quality is mixed across both families and instances, so these runs do not establish a stable search advantage at matched budget. Their role is to verify that a model fitted on one set of executions can produce a valid unseen candidate; the structural comparison remains the held-out prediction result in Figure 4.

Table 23: **Executed candidates against the fixed 448-order design.** Rank 1 is the best order in that pool. Regret is normalized by the sampled gap, so a negative value means the candidate beats every order in the pool. The last column is the exact probability that 113 draws without replacement from this finite design match or beat the candidate; it is not a probability under uniform search over S_8.

Instance	Family	Rank	Regret	Random-search match
0	Assignment	20	$+0.121$	0.997
0	Suffix	19	$+0.114$	0.995
1	Assignment	2	$+0.013$	0.252
1	Suffix	13	$+0.116$	0.971
2	Assignment	1	-0.038	0.000
2	Suffix	46	$+0.129$	1.000

E Additional controls and diagnostics

This appendix asks whether the structure reported in Appendices C and D could instead be explained by family capacity, by the choice of terminal objective, or by the normalization the errors use.

E.1 Capacity controls

Capacity-matched nulls. For each head instance, we shuffled the 24 terminal objectives across permutations 1,000 times and refit the same minimax assignment family. The null leaves the objective multiset, gap, and assignment design unchanged. We evaluate both separation from this capacity-matched null and the realized objective regret of the fitted assignment optimizer (Table 24). The latter is not bounded tightly by $D_{\mathcal{A}} < 1$ alone.

Table 24: **Assignment structure exceeds a capacity-matched shuffled-label baseline and usually selects a near-optimal order.** "Optimal" counts instances whose fitted assignment minimizer attains the terminal-objective minimum. Regret is $(J(\widehat{\pi}) - J(\pi^\star))/\Delta_J$.

Setting	Instances	Shuffle p	Optimal	Max. regret
Frozen ridge	45	≤ 0.001	44/45	0.0007
Direct head	15	≤ 0.001	13/15	0.151
Factorized head	15	≤ 0.001	10/15	0.452

We permuted each table's objective labels 1,000 times and re-fit the identical suffix $L{=}1$, suffix $L{=}2$, and assignment families, which retains the objective distribution and gap while destroying the request–position association. On the exhaustive surfaces all three families beat every one of the 1,000 shuffles in all ten instances and all three at $k = 5$. None of the $1{,}000$ capacity-matched shuffles matched the observed errors.

The separation from the shuffled fits shows that the observed localization contains request–position information beyond family capacity.

E.2 Behavioural anchoring of the terminal objective

The terminal objective is a dense structural-analysis objective, chosen so that the induced permutation problem is measurable on every order. This subsection asks what accompanies the ordering variation it exposes, using the executions already reported and adding none. The complete $k = 4$ full-network surfaces record retained and deleted-group accuracy for each of the 24 orders, so both quantities are available beside each objective value.

Every quantity below is measured within one policy and one instance. The two policies optimise different things, so an order effect under one of them describes that policy alone, and the table places their accuracies on a common scale only to the extent of asking whether an effect is present. The contrasts are differences from the random-order mean of the same surface, and the rank correlation is Spearman's between the 24 objective values and the 24 accuracies of one instance (Table 25).

Retained accuracy is more closely aligned with the objective than deleted-group accuracy under both policies. The latter varies more across orders but has little rank alignment with the objective. Terminal reference distance is therefore a utility-anchored comparison objective, not a forgetting metric; a deployment would pair it with a separate forgetting criterion.

Table 25: **Ordering variation in the terminal objective is accompanied by a consistent retained-accuracy effect within each policy.** Entries are medians with observed $[\min, \max]$ over ten complete $k = 4$ surfaces per policy. The first two columns are percentage-point differences from the random-order mean of the same surface, the third is the observed spread across its 24 orders, and the last is the within-instance Spearman correlation between objective and accuracy. Rows describe one policy at a time.

Policy	Quantity	J-best	J-worst	Range	Rank corr.
NegGrad+	Retained acc.	$+0.603$	-0.569	1.913	-0.670
	Deleted-group acc.	$+0.583$	$+0.660$	4.533	-0.166
SalUn	Retained acc.	$+0.772$	-0.711	2.189	-0.833
	Deleted-group acc.	$+0.502$	-0.801	5.397	-0.087

Absolute levels and the screens they are compared with are reported in Table 26. The deleted-group screen requires the largest terminal A_F over all orders to be at most $A_F^{\text{base}} - 0.05$ and is checked retrospectively. The retained screen requires the smallest terminal retained accuracy to exceed $\min(A_R^{\text{base}}, A_R^{\text{retrain}}) - 0.05$. Reference progress requires the order-mean terminal distance to be below the base-model distance and was used, with the retained screen, in operating-point selection. These checks are not a privacy guarantee.

Table 26: **Absolute deleted-group accuracy and declared screens.** Entries are medians over ten complete $k = 4$ surfaces per policy. The bracket in A_F^{unlearn} is the range over every terminal order and instance; d_U/d_0 is the order-mean reference distance divided by the base distance. The F screen and R floor columns report the median instance thresholds.

Policy	A_F^{base}	A_F^{retrain}	A_F^{unlearn}	F screen	R floor	d_U/d_0
NegGrad+	0.893	0.000	0.695 [0.569, 0.835]	0.843	0.848	0.617
SalUn	0.900	0.000	0.737 [0.613, 0.839]	0.850	0.848	0.708

The unlearned deleted-group accuracies remain well above their retraining counterparts. The screens establish a decrease, retained utility, and reference progress at the tested operating points; they do not establish retraining-equivalent forgetting.

E.3 Robustness to execution randomness

Repeating a fixed order under the same request-bound tape reproduces the terminal objective exactly, so the question here is not numerical reproducibility but whether the ordering signal survives a different draw. We paired two separate replays of all 24 orders for three $k = 4$ NegGrad+ request sets, holding the model, requests, reference, budget, and 10,000-example audit fixed while changing the request-bound tape (Table 27).

The two tapes select the same best order in only one of the three instances, whereas their assignment and suffix verdicts agree. Thus the exact ranking is less stable than the supported structural family. Only NegGrad+ was paired in this way, so this diagnostic does not establish the same behavior for SalUn or coefficient-level stability across tapes.

Table 27: **A different request-bound tape moves individual objective values but not the structural verdict.** Entries are medians with observed $[\min, \max]$ over three paired complete $k = 4$ NegGrad+ surfaces. The first block is measured between the two tapes, normalized by the random-order gap. The second reports each family's error under each tape separately.

Quantity	Median $[\min, \max]$
RMS objective disagreement	0.193 [0.180, 0.270]
Largest objective disagreement	0.409 [0.357, 0.519]
Rank correlation between tapes	0.867 [0.862, 0.945]
$D_{\mathcal{A}}$, first tape	0.302 [0.265, 0.441]
$D_{\mathcal{A}}$, second tape	0.267 [0.189, 0.347]
$D_{S,2}$, first tape	0.221 [0.204, 0.286]
$D_{S,2}$, second tape	0.204 [0.126, 0.276]

E.4 Relation to existing ordering rules

Kumar et al. (2026) order a forget set by base-model predictive uncertainty. That rule acts on the object this paper permutes, a fixed request set whose order is the decision, so it can be placed on our surfaces directly. Zhao et al. (2024) instead vary how a forget set is decomposed into sequential requests. That changes the request set rather than the order of a fixed one, so it is not a permutation rule on the same S_k, and we do not treat it as one here.

The uncertainty rule was executed reference-free on ten full-network instances, reading one base-model forward pass per request, loading no retraining reference and issuing no terminal-order query. Those ten instances also have complete 24-order surfaces, so the order it selects can be scored exactly without executing anything further (Table 28).

The uncertainty rule neither selects competitive orders consistently nor tracks the scalar score induced by the terminal-distance objective. This is a matched comparison on the same request sets, but the rule was designed for a different objective. Its reference-free information advantage therefore motivates the fixed-objective audit in the next subsection rather than a claim that uncertainty ordering fails for its intended use.

Table 28: **An existing uncertainty rule, scored exactly on the complete surfaces.** Ten complete $k = 4$ NegGrad+ surfaces, the same instances on which the rule was executed reference-free. Rank 1 is the best of the 24 orders, regret is normalized by the random-order gap so that 1 is the random-order mean, and the correlation is Spearman's between the rule's per-request scores and the scalar sorting score of Proposition 1(b).

Quantity	Median $[\min, \max]$
Rank among the 24 orders	15 $[7, 24]$
Normalized regret	1.455 $[0.177, 2.999]$
Better than the random-order mean	3/10
Correlation with the induced score	0.300 $[-0.400, 1.000]$

E.5 Structure under reference-free terminal objectives

The terminal objective compares an executed model with a retraining reference, which a deployment does not have. We therefore used three separate 12-step replays with the same seed-indexed requests and reference files as the first three main-cohort instances, but evaluated on the full 10,000-example test split. On each replay we recomputed the structure under two fixed objectives that need no retraining. Let $c_y(x; w)$ be the terminal softmax probability of the true label. For request i, let $\widehat{F}_i^{\mathrm{del}}$ be the empirical CDF of this score on its deleted training examples and let $\widehat{F}_i^{\mathrm{nm}}$ use fixed CIFAR-100 test examples from the same superclass. With $q_b = b/10$, $b = 1, \ldots, 9$, we define

$$J_{\mathrm{NM}}(w) = \sum_i \sum_{b=1}^{9} \big(\widehat{F}_i^{\mathrm{del}}(q_b) - \widehat{F}_i^{\mathrm{nm}}(q_b)\big)^2.$$

The second objective is the policy loss on fixed sets,

$$J_{\mathrm{pol}}(w) = 0.95\,\mathrm{CE}(w; R_{\mathrm{eval}}) - 0.05\,\mathrm{CE}(w; F_{\mathrm{union}}),$$

where R_{eval} is one fixed 5,000-example permanent-retain sample and F_{union} is the union of all four deleted groups. The same examples, thresholds, and weights are used for every permutation. Each objective is audited against the same capacity-matched shuffle (Table 29).

Both substitutions preserve the assignment and suffix verdict while the matched prefix and middle controls fail. Thus the retraining reference is not the sole carrier of the observed structure. This retrospective evaluation of complete stored replays does not provide a prospective reference-free identification procedure for a new queue.

Table 29: **Separate replays carry the same structural verdict under two objectives that need no retraining reference.** Entries are medians with observed $[\min, \max]$ over three complete $k = 4$ NegGrad+ surfaces. $D_{P,2}$ and $D_{M,2}$ are equal-capacity positional controls. "Separated" counts Holm-adjusted lower-tail comparisons with $p < 0.05$ over the 1,000 capacity-matched shuffles.

Objective	Needs reference	$D_{\mathcal{A}}$	$D_{S,2}$	$D_{P,2}$	$D_{M,2}$	Separated
J_{ref}	Yes	0.302 $[0.265, 0.441]$	0.221 $[0.204, 0.286]$	2.246 $[1.423, 2.437]$	2.580 $[1.550, 3.029]$	3/3
J_{NM}	No	0.234 $[0.116, 0.376]$	0.320 $[0.189, 0.377]$	1.634 $[1.544, 1.794]$	1.872 $[1.494, 2.012]$	3/3
J_{pol}	No	0.190 $[0.119, 0.773]$	0.242 $[0.101, 0.835]$	1.963 $[1.693, 2.119]$	2.157 $[1.790, 2.428]$	3/3

E.6 Sequential ordering versus pooled execution

This is a policy-scope control, not an ordering result. The paper fixes a sequential policy and asks which order to run it in. A deployment that may instead aggregate the requests is choosing a different policy. On the same ten complete $k = 4$ NegGrad+ instances, we executed a single pooled update over the union of the four request sets under a matched total budget of 48 steps, with the same base model, reference and hyperparameters. All ten pooled executions satisfy the validity screens, and each reaches a terminal objective below the best of its own 24 sequential orders, a median reduction of 37.8% (30.6–44.2%) against the best sequential order.

When aggregation is available, changing the execution policy can therefore dominate reordering within a fixed sequential policy. This does not bear on the structural claims, which are about the permutation problem a fixed request-wise policy induces. That fixed policy is a modeling scope, not a claim that pooling is unavailable in every deployment.

E.7 Queue-length diagnostics

Attenuation across queue lengths. Corollary 5 says the quantities to watch as a queue grows are the contraction and the decision scale, not k. Holding the NegGrad+ policy and its budget fixed, the measured two-request attenuation is nearly unchanged across $k = 4$, 5 and 8, with medians 0.069, 0.071 and 0.065 and between-queue variation smaller than the within-queue range. This finite-execution attenuation diagnostic therefore does not drift with k over the range we execute.

Decision-scale diagnostics. Corollary 5 puts every queue-length dependence into $\mu_1(k)$ of Equation 26, which the complete surfaces measure directly. On the full-network settings the last position alone carries most of the available signal, $\mu_1/\Delta_J = 0.89$ at $k = 4$ and 0.93 at $k = 5$, and the scale itself does not shrink with the queue. μ_1 is 8.31% of the mean objective at $k = 4$ and 8.98% at $k = 5$. The decision scale is therefore non-degenerate over the queue lengths we execute, which is weaker than the uniform condition $\inf_k \mu_1(k) > 0$ that Corollary 5 assumes.

F Limitations and Scope

Our analysis conditions on a fixed request-wise policy and terminal objective. Changing the policy, including pooling requests, is a separate decision and can dominate reordering in the tested setting (Appendix E.6). Reference distance is used as a dense structural objective rather than a complete forgetting or privacy criterion (Appendices E.2 and E.5). Exact optimal orders are more sensitive to execution randomness than the structural families studied here (Appendix E.3). Complete permutation surfaces are limited to modest queue sizes, while larger-k results rely on held-out evaluation (Appendix D). Adaptive arrivals, joint policy selection, and exact complexity under contraction bounded away from one remain open; the last boundary is detailed in Appendix A.2.