SCREENING UPDATERS FOR MULTIMODAL RECOMMENDER ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A recommender often starts from one deployed reference model but still has several ways to update it. Those candidate updaters share the same reference predictions, yet they may not be able to correct the same part of the reference error. We study which updater is worth trying when several updater mechanisms share one reference. A held-out local correction score fits the reference residual in a finite tangent sketch on one earlier cohort and evaluates that learned direction on disjoint users before full adaptation. The score is not a new linearization estimator; it is a screening signal for same-reference updater selection. Across three multimodal recommendation sources, it improves on fixed development rankings, adapter-aware transfer baselines, and short nonlinear trials at the primary scale. Supporting 24- and 48-candidate searches show where screening becomes useful as candidate count or updater scale increases: the score narrows the search before a shortlist receives explicit nonlinear trials. The same measurement also predicts later Brier and ranking gains across time blocks. Preservation is a downstream decision after updater selection: replay keeps more average improvement, while an explicit task-loss requirement reduces positive group degradation when measured correction is large. Capacity, locality, and timing checks define the setting in which this staged updater-search procedure is most useful.

1 INTRODUCTION

Multimodal recommenders are often updated through small add-on modules rather than full retraining. In practice, the same reference model may admit several candidate update mechanisms, such as image-only, text-only, fusion, or joint residual branches. These candidates start from the same reference predictions, but they do not necessarily offer the same opportunity for improvement.

Most update decisions are still made with coarse signals. A practitioner may rely on a fixed prior learned on earlier blocks, or run a short trial update for every candidate and keep the best one. The first strategy ignores the current condition. The second becomes increasingly costly as the candidate set or updater scale grows. This creates a practical question: before full adaptation, can we estimate which updater is worth trying for the current reference and cohort?

Our starting point is that total reference error is not the same as correctable error. Two updaters can face the same reference predictions and the same observed labels, while only one of them can move in a direction that reduces the current error. We use the term *accessible correction* for the part of the error that a specified updater can reach near the reference. This leads to a local correction score: fit a residual direction on one earlier cohort, evaluate it on disjoint users, and use the resulting score to rank candidate updaters before long optimization.

We study this idea in three steps. First, we compare the score with a strong development prior, adapter-aware transfer baselines, and short nonlinear trial updates when several updaters share one reference. Second, we test whether the same local signal predicts later gains in Brier loss and ranking metrics across time blocks. Third, we ask how measured correction opportunity changes preservation decisions after an updater has been chosen.

Our contributions are as follows.
• We formulate updater choice as a same-reference decision problem and show that a simple local correction score improves condition-specific updater ranking over fixed priors and standard adapter-

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.

aware baselines.
• We show that the score is most useful as a first-stage screen: when many candidates are available, it narrows the search before a shortlist receives explicit nonlinear trials.
• We study a downstream use after updater selection: measured correction opportunity helps determine when replay or a stricter task-loss preservation rule is preferable.

2 Related Work

Transferability and linearized fine-tuning. H-score, LEEP, LogME, SFDA, and PACTran estimate transferability of a representation or source predictor to a target task (Bao et al., 2022; Nguyen et al., 2020; You et al., 2021; Shao et al., 2022; Ding et al., 2022). Linearized fine-tuning predicts update behavior from local features around a pretrained model (Deshpande et al., 2021; Malladi et al., 2022). These works provide direct methodological context. Under squared loss, a residual linear probe on the same Jacobian features is algebraically identical to the correction score used here. We therefore treat the score as a held-out linearized trial, and ask what it contributes to *updater selection* when multiple update mechanisms share one reference.

Training-free and low-cost selection. Neural architecture search has developed training-free and zero-cost proxies that rank candidate structures from their initial state, including activation overlap, pruning-inspired scores, and NTK-based criteria (Mellor et al., 2021; Abdelfattah et al., 2021; Chen et al., 2021). Surgical Fine-Tuning selects layers for targeted adaptation, while AutoPEFT searches PEFT module type, size, and insertion layers (Lee et al., 2023; Zhou et al., 2024). The selection object differs here: the deployed reference is fixed, while the candidate is an updater attached to that same reference. This parallel motivates two controls that are easy to overlook: a fixed ranking learned on development blocks, and short nonlinear trials that spend a small optimization budget on every candidate.

Model updates and preservation. Residual fitting and tangent-kernel analyses provide the local geometry (Friedman, 2001; Jacot et al., 2018; Chizat et al., 2019; Ortiz-Jiménez et al., 2023). Distillation constrains outputs, EWC constrains parameter movement through estimated importance, replay repeats earlier supervision, and positive-congruent or backward-compatible training targets regressions across model versions (Hinton et al., 2015; Kirkpatrick et al., 2017; Chaudhry et al., 2019; Yan et al., 2021; Shen et al., 2020; Schumann et al., 2024). Recommendation-specific retraining work such as SML and GraphSAIL addresses continual update and forgetting (Zhang et al., 2020; Xu et al., 2020). We separate that update-maintenance problem from choosing which updater to try before full adaptation, and separate the candidates produced by preservation objectives from the common validation rule used for deployment.

Recommendation evaluation. FREEDOM and BM3 are representative multimodal recommender architectures (Zhou & Shen, 2023; Zhou et al., 2023). The study focuses on update mechanisms attached to a fixed reference rather than on a new ranking backbone. We therefore report same-reference updater selection on three recommendation sources and repeat the comparison with FREEDOM- and BM3-style references. Because sampled ranking metrics can change the evaluation scale (Krichene & Rendle, 2020), a separate catalogue check scores every eligible warm item on a fixed query subset.

3 Method

3.1 Reference error and updater-specific correction

For an oriented item pair, $Y \in \{0, 1\}$ identifies which item is the observed target and $r(X)$ is the reference probability. Let $\pi(X) = \mathbb{E}[Y \mid X]$ and $e = \pi - r$. If feasible probability changes lie in a closed linear space $\mathcal{H} \subset L^2$, squared risk satisfies

$$\mathcal{R}(r + h) - \mathcal{R}(r) = \|h - a\|_2^2 - \|a\|_2^2, \qquad a = \Pi_{\mathcal{H}} e. \tag{1}$$

Thus $\|a\|_2^2 \le \|e\|_2^2$: equal reference error does not imply equal correctable error. This is the standard projection identity; its role is to define the empirical object. A neural updater supplies only a local linear space at the reference, and finite training can later leave that space.

Let $J_0(X)$ be the probability Jacobian of a chosen updater at the reference. Rather than claim access to the full tangent space, we build a label-free k-dimensional tangent basis U from the Jacobian on cohort Q_1. The main study uses $k = 64$; Appendix C reports $32/64/128/256$-dimensional coverage. With $Z(X) = J_0(X)U$, we fit

$$\hat{v} = \arg\min_v \widehat{\mathbb{E}}_{Q_1}\left[(Y - r - Zv)^2\right] + \lambda\|v\|_2^2. \tag{2}$$

On disjoint users Q_2, the held-out correction score is

$$\widehat{G} = \widehat{\mathbb{E}}_{Q_2}\left[2\hat{h}(Y - r) - \hat{h}^2\right], \qquad \hat{h} = Z\hat{v}. \tag{3}$$

Conditional on the fitted basis and direction, its expectation is

$$\mathbb{E}[\widehat{G} \mid Q_1, U] = \|a_U\|_2^2 - \|\hat{h} - a_U\|_2^2, \quad a_U = \Pi_{\mathcal{H}_U} e. \tag{4}$$

The score therefore measures a learned correction in the *sketched local space*; full nonlinear updater access remains a separate object. A residual linear probe with the same offset computes the same quantity. This equivalence is useful because it makes the contribution testable against simpler interpretations rather than depending on a new name.

3.2 Comparisons

We compare information available before the full update in a fixed order: reference Brier loss; residual-gradient norm; feature H-score and LogME; Jacobian H-score and LogME; a Jacobian probability probe; empirical NTK regression; and direct nonlinear trials of 1, 2, 5, 10, 20, or 60 Adam steps. All updater-specific baselines use the same Q_1/Q_2 roles and candidate updater parameterization; each Direct-k budget uses its own learning rate selected on the development blocks and frozen for evaluation. A development prior provides one fixed source-level ranking from mean future gain over the six development blocks and never uses the current evaluation condition. These controls separate condition-specific update information from average updater quality.

3.3 Experimental design

Recommendation sources and time blocks. We use Amazon Reviews 2023 Baby products (Hou et al., 2024), MicroLens-1M (Ni et al., 2023), and Pixel200K from PixelRec (Cheng et al., 2023). Each source provides timestamped interaction targets and released visual/textual item content. We construct six development blocks and 24 evaluation blocks. Within a block, users are assigned once to direction fitting, direction scoring, update training, preservation, validation, checking, or future evaluation. All roles use disjoint target users; permitted earlier histories may be shared with reference fitting. The allocation uses 48 users per group for Q_1 and Q_2, 64 for update training, 48 each for preservation/validation/checking, and 192 for future evaluation. This uses 59,520 distinct role users per source over all 30 blocks and leaves a substantial eligibility reserve (Appendix B).

One reference, multiple updaters. The reference combines 128-dimensional frozen image and text representations with user history. The primary same-reference experiment attaches image-only, text-only, fusion-only, and joint centered residual updaters. All four begin at exactly the same prediction function and use 2,048 trainable parameters. Baby and MicroLens replicate this four-updater comparison; Pixel200K additionally supplies a 12-updater search that varies modality branch and insertion point. The later capacity check uses 512–65,536 trainable parameters. The main updater-selection question therefore changes the update mechanism while holding the reference error fixed. History groups are evaluation strata, not a requirement to deploy one group-specific adapter; operational use may aggregate the condition-level evidence at the source or service level.

Outcomes. The first target is unconstrained future Brier gain $Y_c = R_c(r) - R_c(f)$ after a development-fixed 320-step update. No preservation ceiling or future candidate selection enters this outcome. Ranking outcomes are ΔNDCG@20 and ΔRecall@20 on a frozen retriever's top-200 list, plus catalogue ΔNDCG@20 on a fixed 2,048-query subset. We also define a ranking-gain oracle that orders candidate updaters within each condition by future ΔNDCG@20; this is distinct from the future Brier-gain oracle used for the primary updater-choice results. Later preservation experiments compare prediction consistency, replay, EWC, positive-congruent training, and a task-loss constrained update. Every family receives the same update users, preservation users, optimizer budget, and final validation requirement.

Selection signal	Pair order ↑	Best updater ↑	Regret 10^3 ↓
Development prior	65.3	40.3	0.25
Residual gradient	67.2	47.9	0.25
Feature H-score	64.0	43.8	0.29
Feature LogME	66.4	46.9	0.26
Jacobian H-score	69.0	49.7	0.24
Jacobian LogME	70.4	53.1	0.23
Jacobian probe	74.1	59.7	0.20
Empirical NTK regression	75.6	62.5	0.18
Direct, 1 step	68.6	49.7	0.24
Direct, 2 steps	71.3	55.2	0.22
Direct, 5 steps	74.8	62.8	0.19
Correction score	79.8	69.8	0.12
Direct, 10 steps	76.9	66.0	0.16
Direct, 60 steps	82.8	73.3	0.09

Table 1: Same-reference updater choice, macro-averaged over Baby, MicroLens, and Pixel200K. The development prior is a fixed ranking learned on development blocks; its top-ranked updater is joint, so its top-1 accuracy and regret equal Always joint. Regret is future Brier-gain regret.

Inference. Cross-block prediction withholds all groups and updaters of one evaluation block together. Pair-ordering intervals also keep all candidate pairs inside their time block. The 24 evaluation blocks are the sampling units; user pairs are not treated as 24 independent environments. Contiguous-block deletion provides a dependence sensitivity analysis. Prediction, updater selection, and preservation are reported as separate research questions with their own evaluation units.

4 Experiments and Results

4.1 Which updater is worth trying?

The same-reference experiment first asks whether condition-specific information improves on a fixed prior. The development prior is already useful: averaged over the three sources, its fixed ranking orders 65.3% of updater pairs correctly, showing that updater quality is not exchangeable. Its top-ranked updater is joint on all three sources, so its top-1 accuracy and regret match the Always-joint baseline. The condition-specific correction score reaches 79.8% pair ordering and selects the best updater in 69.8% of conditions (Table 1). Two disjoint halves of the future users agree on 89.3% of updater-pair orderings and on best-updater identity in 79.4% of comparisons. These agreements provide the empirical reliability ceiling used to interpret the remaining differences. A fixed joint updater is best in only 40.3% of conditions, despite having the largest mean gain. The difference is the reason to predict condition-specific access rather than use the globally strongest updater everywhere.

The adapter-aware baselines matter. A one-step residual-gradient score is informative but remains clearly below the fitted correction score. Jacobian LogME reaches 70.4% pair ordering, and the Jacobian probe reaches 74.1%; using updater-specific Jacobian features explains part, but not all, of the selection signal. Empirical NTK regression is also competitive but remains below the held-out residual score. The correction score reaches 79.8%, adding 9.4 points over Jacobian LogME. It also improves pair ordering over Direct-5 by 5.0 points and over Direct-10 by 2.9 points; both paired block intervals exclude zero (Table 13). Weighting pair decisions by the realized gain gap raises the development prior to 73.0%; the correction score reaches 87.1%, compared with 84.0% for Direct-10, when near-tied updater pairs receive proportionally less weight (Table 14). Direct-60 remains stronger in pair ordering, but it spends a much larger nonlinear trial budget. The updater-specific signal also transfers to choosing the updater with the largest future NDCG gain: pair ordering is 76.8% for the correction score, 74.9% for Direct-10, and 79.5% for Direct-60 (Table 15). The result is therefore not that a local calculation replaces explicit optimization, but that it provides useful condition-specific ordering before committing to a long update.

Pixel200K illustrates the mechanism. Image, text, fusion, and joint updaters have the same reference and parameter count, yet their mean earlier correction scores and later gains differ (Figure 1). Joint has the largest average gain but is best in 39 of 96 conditions; image, text, and fusion are best in 27, 13, and 17. The ordering also appears in NDCG@20, although ranking gaps remain smaller than risk-gain gaps. The same-reference result is not unique to Pixel200K. Table 2 shows the corresponding pair-ordering accuracy on Baby and MicroLens. The development prior remains a meaningful baseline on every source, while the condition-specific score adds roughly fourteen to fifteen points. Direct-60 remains strongest because it spends the largest nonlinear trial budget.

Source	Dev. prior	Correction score	Direct-10	Direct-60
Baby	65.9	80.4	77.6	83.1
MicroLens	63.7	79.1	76.3	82.0
Pixel200K	66.4	80.0	76.9	83.2

Table 2: Same-reference updater pair-ordering accuracy (%) by source. Every row compares equal-budget image, text, fusion, and joint updaters starting from one reference.

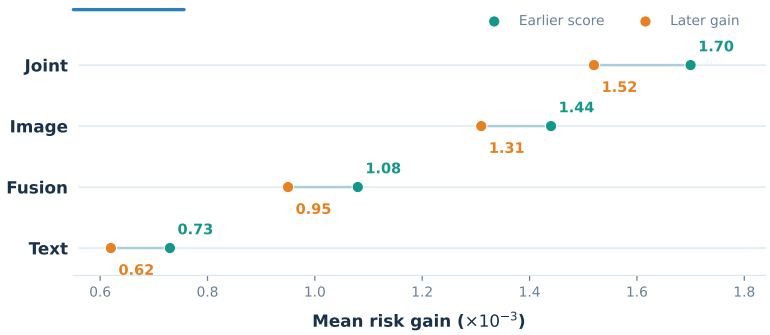


Figure 1: Pixel200K updater means under one fixed reference. The horizontal axis is risk gain: larger values indicate greater reduction in squared loss. The score is evaluated before the full update; later gain is measured on future users.

4.2 When there are many candidates

The score is stronger than the five- and ten-step trials in pair ordering, but explicit nonlinear trials remain informative and become stronger with larger budgets. Pixel200K supplies 12 equal-budget candidates spanning three variants of image, text, fusion, and joint update branches. Scoring all candidates and applying Direct-10 only to the top two retains 96.0% of oracle gain at 40.4% of the cost of Direct-10 on all candidates; the full Direct-10 search retains 96.6%. At the primary scale, the two-stage rule costs 0.38 s versus 0.94 s for Direct-10 on all candidates and 26.94 s for full training of all twelve candidates. The score shortlist retains 96.0% [94.9,97.0] of oracle gain versus 95.0% [93.6,96.2] for successive halving, a +1.0 point difference [0.3,1.7]. The same screening pattern holds in 24-candidate 16k-parameter searches on Baby and MicroLens. At 48 candidates and 65k parameters on Pixel200K, screening reduces a 54.96 s Direct-10 search to 15.86 s while retaining 95.4% of oracle gain. The advantage over successive halving grows with candidate-set size in these supporting searches. The main practical use is therefore a two-stage search: local screening first, then explicit nonlinear trials on a shortlist.

4.3 Does the local score predict realized gain?

We next fix the joint updater and predict how much unconstrained adaptation will help across blocks and history groups. The Base predictor contains reference Brier loss and log gradient norm. Adding the correction score raises held-block R^2 from .248 to .603 on Baby, .137 to .579 on MicroLens, and .162 to .611 on Pixel200K (Table 4). Direct-5 is the cheapest explicit nonlinear trial in the primary comparison but costs about 2.3 times the local score. Direct-10 narrows the gap without closing it; Direct-60 is stronger. The Jacobian features explain most of the signal, while residual fitting adds a smaller but consistent increment over the Jacobian probe: +.028/.027/.026 R^2 on Baby/MicroLens/Pixel200K. This ordering matches the interpretation above: more nonlinear optimization provides more information, at greater search cost.

The score is fitted to squared-loss residuals, so Brier gain is its most direct target. We therefore also evaluate ranking outcomes. Its Spearman relation with ΔNDCG@20 is .621/.596/.609 on Baby/MicroLens/Pixel200K, and with ΔRecall@20 is .587/.561/.574 (Table 5). Catalogue ΔNDCG is weaker but remains positive. This pattern is expected: the score is loss-aligned, yet the updater-

Search rule	Relative cost	Oracle gain retained ↑	Regret 10^3 ↓
Development-prior top 1	0.01	81.0	0.31
Correction score, top 1	0.24	91.2	0.15
Correction score → Direct-10 top 2	0.40	96.0	0.06
Direct-10 on all 12	1.00	96.6	0.05
Full train all + validation	28.64	99.4	0.01

Table 3: Twelve-updater Pixel200K search. Cost is normalized to Direct-10 on all candidates. The correction score is most useful as a first-stage screen followed by nonlinear evaluation of a shortlist.

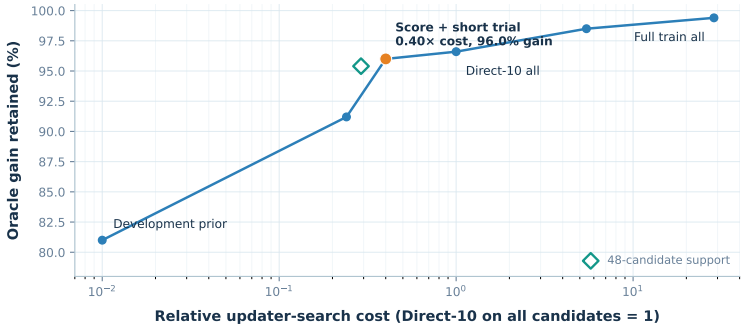


Figure 2: Efficiency frontier for the twelve-updater Pixel200K search. The highlighted two-stage strategy scores all candidates, then spends nonlinear trial compute only on a shortlist. The open diamond is the separate 48-candidate Pixel200K support point and is not part of the twelve-candidate frontier.

specific information partly transfers to ranking changes that are not used in Equation 3. Direct updater choice by future ΔNDCG@20 gives the same qualitative ordering (Table 15).

Where does the remaining error come from? Prediction is not exact. On independent check users, the score is most accurate when the final update remains near its initial tangent space; held-block R^2 falls from .65 to .41 between the smallest and largest update thirds. The score also systematically exceeds later gain. We separate this optimism into two observable steps: local linearized gain on Q_2, nonlinear ten-step gain on the same time slice, and gain after the future interval (Table 6). Both nonlinear realization and temporal transfer explain part of the gap. Calibration slopes below one summarize the same pattern: the local score ranks conditions well but overstates their eventual magnitude.

4.4 Downstream use: preservation after updater selection

Preservation is evaluated after the updater has been chosen. We define I as mean future task-risk improvement $R_{\text{future}}(r) - R_{\text{future}}(f)$, P as the mean positive group degradation relative to the reference, Viol. as the fraction of evaluated group–block cells with positive degradation, and Worst as the largest positive mean degradation among those cells. Replay and task-loss training receive the same earlier labels. Replay mixes them into the training objective; task-loss training limits their loss increase relative to the reference. A common strict validation requirement is then applied to every candidate family. The aggregate result is familiar: replay realizes more improvement, while task-loss training reduces positive group degradation. The new question is whether this trade-off depends on how much useful correction was measured before training.

Table 7 answers that question by splitting conditions into within-source correction-score tertiles. In the lowest tertile, task-loss training retains 96.9% of replay's improvement and positive degradation is low for both methods. In the highest tertile, replay realizes much more gain, but also concentrates more degradation; task-loss training retains 81.9% of replay's gain while reducing positive degradation by 50.4%. The high-versus-low access interaction remains negative after conditioning on realized update RMS; the macro interaction is -0.31×10^{-3} with interval $[-0.44, -0.18] \times 10^{-3}$ (Table 25). Strong preservation is therefore cheapest when little correction is available and most consequential when opportunity is high.

Source	Base	Score	J-LogME	J-probe	D1	D2	D5	D10	D60
Baby	0.248	0.603	0.472	0.575	0.389	0.438	0.505	0.548	0.621
MicroLens	0.137	0.579	0.441	0.552	0.365	0.417	0.473	0.530	0.603
Pixel200K	0.162	0.611	0.459	0.585	0.408	0.462	0.498	0.552	0.630

Table 4: Held-block R^2 for future unconstrained Brier gain. Every diagnostic augments the same Base features. J denotes the same tangent features used by the correction score.

Source	Brier gain	ΔNDCG@20	ΔRecall@20	Catalogue ΔNDCG
Baby	0.724	0.621	0.587	0.552
MicroLens	0.692	0.596	0.561	0.526
Pixel200K	0.716	0.609	0.574	0.541

Table 5: Rank correlation between the earlier correction score and four later outcomes. Ranking targets are not used to fit the local direction.

A small routing study makes the roles of the two signals explicit. An access-only policy decides whether enough correction is available; a sensitivity-only policy measures earlier preservation cost; the combined policy uses both. The combined rule retains 92.0% of oracle improvement with 10.7% future violations, compared with 88.1%/15.6% for access alone and 82.3%/11.8% for sensitivity alone. The matched direct-trial rule reaches 91.5%/11.2%. The combined local rule shows a small but consistent advantage over the matched direct-trial policy: gain retention differs by +0.5 points [0.1,0.9], future violations by -0.5 points [-0.9,-0.1], and Worst by -0.18×10^{-3} $[-0.31, -0.05]$ (Table 27). Source-specific intervals and the complete preservation comparison are in Appendix E.

4.5 Capacity, locality, and scope

How much of the local updater is measured? The score uses a finite tangent basis rather than the full adapter. Table 8 shows the coverage trade-off on Pixel200K. Thirty-two directions already capture 68.7% of fit-sample Jacobian energy; 128 capture 92.4%. Pair ordering saturates near 80% once the 64–128 direction basis is used, so the main 64-direction setting is not interpreted as the complete updater space. Larger adapters use larger sketch budgets in the scaling study.

When does linearization break? The local approximation is strongest for small updates. Recommendation check users give mean cosine .929/.917/.923 in the smallest update third for Baby/MicroLens/Pixel200K; it falls to .712/.689/.701 in the largest third. Relative error rises accordingly. MM-IMDb shows the same locality pattern: cosine is .904 for small updates and falls as update size grows. For large updates, cosine falls to .668 and relative error rises to .792. The nonlinear reference reaches micro-F1 .6618 and the selected width-eight adapter reaches .6721. The local score can rank candidate updaters even when it is not an accurate reconstruction of the final nonlinear update. We therefore treat it as a local ranking proxy and rely more heavily on short nonlinear trials as the planned update becomes larger.

Architecture and ranking scope. The same-reference ordering persists when the reference architecture changes. Table 9 reports the correction score under the primary residual, FREEDOM-style, and BM3-style references. The absolute accuracy is lower for the graph-based references, but every source remains well above the development-prior baseline. This check changes the reference architecture while preserving the updater-selection question; it evaluates robustness to reference choice rather than the original papers' full leaderboard protocol.

A catalogue check removes retrieved-list truncation as a ranking-scale robustness check. It does not re-evaluate the full updater-selection pipeline. Scoring every eligible warm item on a fixed query subset lowers absolute NDCG, but the task-loss/replay ordering remains unchanged on all three sources. The corresponding reference/task/replay NDCG values are .0627/.0654/.0659 on Baby, .0804/.0829/.0834 on MicroLens, and .0381/.0409/.0414 on Pixel200K. Full catalogue definitions and denominators are in Appendix E.

Source	Linear Q_2	Nonlinear Q_2	Future gain	Calibration slope
Baby	1.61	1.52	1.44	0.92
MicroLens	1.46	1.38	1.29	0.89
Pixel200K	1.70	1.61	1.52	0.90

Table 6: Where the local score loses magnitude information. Gains are in units of 10^{-3}. The first gap reflects local linearization/optimization; the second includes temporal transfer.

Access tertile	Replay P	Replay I	Task P	Task I	I retained	P reduction
Low	0.24	0.98	0.20	0.95	96.9	16.7
Middle	0.65	2.55	0.40	2.24	87.8	38.5
High	1.29	4.21	0.64	3.45	81.9	50.4

Table 7: Preservation trade-off by correction-score tertile, macro-averaged over sources. P and I are in units of 10^{-3}. The last two columns are percentages.

Cost scales with the search, not one comparison. For the 2,048-parameter updater, the 64-direction score costs 18.6 ms, compared with 42.8 ms for Direct-5 and 78.4 ms for Direct-10 after frozen content features are cached. At 65k parameters, the local score remains materially cheaper than Direct-5 and Direct-10. The corresponding twelve-candidate costs are derived from the same timing registry: scoring all candidates and applying Direct-10 only to the top two uses 40.4% of the cost of running Direct-10 on all twelve candidates. At 16k and 65k parameters, the tangent basis is expanded to 128 and 256 directions. Appendix F shows that the score remains predictive as adapter size grows. The score is strongest at the primary scale; the margin over Direct-10 narrows as adapter size grows but does not reverse in the scaling study. This is the intended boundary: local screening is most useful when many candidate mechanisms are available and their eventual updates remain moderate.

The margin remains positive at 2k, 16k, and 65k parameters: .059, .027, and .022 R^2, respectively. See Appendix F for the complete scaling table and uncertainty.

A practical decision sequence. The results support a simple two-stage workflow rather than a universal replacement for short adaptation. First, use a cheap local score or development prior to remove obviously weak updater choices. Second, spend nonlinear trial compute on a shortlist when the candidate search is large or the predicted step is likely to leave the local region. After an updater is chosen, preservation sensitivity determines whether replay or a stricter task-loss control is appropriate. This sequence keeps updater choice and preservation choice separate while using each signal where it is most informative.

Temporal dependence. The 24 evaluation blocks are consecutive. Deleting contiguous four-block runs widens the task-loss versus replay intervals but retains a negative degradation contrast on all three sources (Appendix D.6). We therefore interpret the result as a repeated temporal pattern within each source, not as evidence from 24 independent platforms.

5 Discussion

The main result is narrower and more practical than introducing a new learning principle. The linearization is classical, and the same squared-loss score can be written as an offset residual probe. The contribution here is to use that local signal for a specific decision problem: several update mechanisms share one reference, but they do not share the same usable correction. A development-wide prior already captures part of this structure, yet a condition-specific score reduces regret because it reacts to the current reference and cohort.

The comparison with short adaptation also clarifies the intended use. At the primary scale, the score is both cheaper and more predictive than the five- and ten-step trials, while the longer sixty-step trial is stronger at substantially greater cost. The local score becomes most useful when the search space is large. In that setting, it screens the full candidate set and reserves explicit optimization for a shortlist.

Tangent directions	Energy coverage %	Pair order %	Diagnostic ms
32	68.7	76.1	13.2
64	82.9	80.0	18.6
128	92.4	80.9	31.8
256	96.9	81.1	56.4

Table 8: Pixel200K tangent-basis coverage. The basis is constructed from the reference Jacobian without residual labels.

Reference architecture	Baby	MicroLens	Pixel200K
Primary residual reference	80.4	79.1	80.0
FREEDOM-style reference	74.1	72.8	74.6
BM3-style reference	75.3	73.6	75.1

Table 9: Same-reference updater pair-ordering accuracy (%) under three reference architectures. The primary residual reference is the same reference used in Table 2.

This is a search-efficiency result, not a claim that one local calculation should always replace explicit optimization.

The preservation study adds a second point. Correction opportunity and preservation sensitivity answer different questions. The first asks whether an update is worth pursuing; the second asks how much earlier behavior it risks changing. When measured opportunity is small, replay and task-loss training stay close. When opportunity is large, replay captures more gain but also creates more positive group degradation. This is why updater choice and preservation choice should be separated rather than folded into one score.

Two boundaries remain part of the result. First, the score measures a sketched local space, so its reliability depends on tangent coverage and weakens as the realized update becomes more nonlinear. Second, it is measured on an earlier population, so later distribution change can reduce or even reverse the usefulness of a previously helpful direction. These limitations do not negate the local signal; they define where shortlist-based nonlinear trials deserve more weight.

The operating interpretation is staged: local screen $\rightarrow$ nonlinear shortlist trial $\rightarrow$ preservation decision. Multimodality supplies several update branches that share one reference; the contribution is not a new multimodal architecture. The same-reference question remains distinct from transferability or architecture selection because the deployed reference prediction function is held fixed while only the updater changes. These experiments focus on adapter/PEFT-style recommender maintenance with frozen content encoders and offline ranking targets; they do not establish the same behavior for embedding-table or full-model retraining.

6 Conclusion

Within adapter-style multimodal recommender maintenance, reference error alone is not enough to choose an updater. Updater-specific local correction provides a condition-specific screening signal before full training. The results support a staged procedure: use the local score to narrow a large updater set, spend nonlinear trial compute on a shortlist when needed, and make preservation decisions after the updater is chosen. This separation matters because updater choice, realized gain, and preservation cost need not favor the same mechanism in every condition. The local score is most useful when several candidate branches share one reference and exhaustive nonlinear trials are costly. For larger updates, short nonlinear trials provide a complementary check before deployment. Across evaluated sources, this staged view clarifies why no single signal should solve the entire adaptation problem. The correction score estimates whether a candidate updater has useful local access to the current residual, whereas nonlinear trials test how that opportunity survives finite optimization. Preservation criteria then address a different question: how much of the realized gain can be kept without excessive degradation on earlier behavior. Treating these decisions separately makes the workflow easier to interpret and gives each component a clear operational role. This scope is deliberately narrower than a claim about full-model incremental retraining.

AI USE STATEMENT

Generative AI tools were used to improve the clarity and readability of the manuscript and to assist with parts of the software implementation, including code drafting, debugging, and refactoring. AI tools were also used to provide feedback on experimental design, analysis, and presentation. All AI-assisted text, code, analyses, and suggestions were reviewed and checked by the authors. The authors take responsibility for the final manuscript, implementation, reported results, and conclusions.

ETHICS STATEMENT

This study uses publicly released recommendation datasets and item content. It does not collect new private user data or attempt to identify individuals or infer sensitive attributes. Results are reported in aggregate and are used to study updater selection and preservation in recommender systems.

REPRODUCIBILITY STATEMENT

The appendices document the data processing, temporal splits, updater and baseline definitions, statistical procedures, and timing protocol. The supplementary material provides anonymized code and configuration files for reproducing the reported analyses.

REFERENCES

Mohamed S. Abdelfattah, Abhinav Mehrotra, Łukasz Dudziak, and Nicholas D. Lane. Zero-cost proxies for lightweight nas. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0cmMMy8J5q.

John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A. González. Gated multimodal units for information fusion, 2017. URL https://arxiv.org/abs/1702.01992.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning, 2022. URL https://arxiv.org/abs/2212.10082.

Sébastien Bubeck. Convex optimization: Algorithms and complexity, 2015. URL https://arxiv.org/abs/1405.4980.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019. URL https://arxiv.org/abs/1902.10486.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2102.11535.

Yu Cheng, Yunzhu Pan, Jiaqi Zhang, Yongxin Ni, Aixin Sun, and Fajie Yuan. An image dataset for benchmarking recommender systems with raw pixels, 2023. URL https://arxiv.org/abs/2309.06789.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1812.07956.

Aditya Deshpande, Alessandro Achille, Avinash Ravichandran, Hao Li, Luca Zancato, Charless Fowlkes, Rahul Bhotika, Stefano Soatto, and Pietro Perona. A linearized framework and a new benchmark for model selection for fine-tuning, 2021. URL https://arxiv.org/abs/2102.00084.

Nan Ding, Xi Chen, Tomer Levinboim, Beer Changpinyo, and Radu Soricut. Pactran: Pac-bayesian metrics for estimating the transferability of pretrained models to classification tasks, 2022. URL https://arxiv.org/abs/2203.05126.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation, 2024. URL https://arxiv.org/abs/2403.03952.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1806.07572.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1748–1757, 2020. doi: 10.1145/3394486.3403226.

Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=APuPRxjHvZ.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning, 2022. URL https://arxiv.org/abs/2210.05643.

Joe Mellor, Jack Turner, Amos Storkey, and Elliot J. Crowley. Neural architecture search without training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7588–7598, 2021. URL https://proceedings.mlr.press/v139/mellor21a.html.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7294–7305, 2020. URL https://proceedings.mlr.press/v119/nguyen20b.html.

Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. A content-driven micro-video recommendation dataset at scale, 2023. URL https://arxiv.org/abs/2309.15379.

Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.12827.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang. Backward compatibility during data updates by weight interpolation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2846–2861, 2024. doi: 10.18653/v1/2024.eacl-long.174. URL https://aclanthology.org/2024.eacl-long.174/.

Sentence Transformers. paraphrase-multilingual-MiniLM-L12-v2: Model card, 2021. URL https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2. Accessed September 25, 2026.

Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *European Conference on Computer Vision*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/6972_ECCV_2022_paper.php.

Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6368–6377, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Shen_Towards_Backward-Compatible_Representation_Learning_CVPR_2020_paper.html.

Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 2861–2868, 2020. doi: 10.1145/3340531.3412754. URL https://doi.org/10.1145/3340531.3412754.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14299–14308, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yan_Positive-Congruent_Training_Towards_Regression-Free_Model_Updates_CVPR_2021_paper.html.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12133–12143, 2021. URL https://proceedings.mlr.press/v139/you21b.html.

Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. How to retrain recommender system? a sequential meta-learning method. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1479–1488, 2020. doi: 10.1145/3397271.3401167. URL https://doi.org/10.1145/3397271.3401167.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. Autopeft: Automatic configuration search for parameter-efficient fine-tuning. *Transactions of the Association for Computational Linguistics*, 12:525–542, 2024. doi: 10.1162/tacl_a_00662. URL https://aclanthology.org/2024.tacl-1.29/.

Xin Zhou and Zhiqi Shen. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the ACM International Conference on Multimedia*, 2023. URL https://arxiv.org/abs/2211.06924.

Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference*, 2023. doi: 10.1145/3543507.3583251. URL https://arxiv.org/abs/2207.05969.

A Geometry and the Meaning of the Score

A.1 Squared loss and updater-specific correction

All norms and inner products in this section use the stated distribution of a user and an oriented comparison pair. Let $Y \in \{0,1\}$, $\pi(X) = \mathbb{E}[Y \mid X]$, reference probability $r(X)$, and residual $e = \pi - r$. For any square-integrable change h,

$$\mathcal{R}(r+h) - \mathcal{R}(r) = \mathbb{E}[(r+h-Y)^2 - (r-Y)^2] \tag{5}$$

$$= \|h\|_2^2 - 2\langle h, e\rangle. \tag{6}$$

If $\mathcal{H}$ is a closed linear subspace and $a = \Pi_{\mathcal{H}} e$, orthogonality gives $\langle h, e\rangle = \langle h, a\rangle$ for $h \in \mathcal{H}$. Completing the square yields Equation 1. In particular,

$$0 \le \|a\|_2^2 \le \|e\|_2^2. \tag{7}$$

The upper bound concerns conditional reference error, not observed Brier loss. Observed Brier also contains the irreducible Bernoulli variance $\mathbb{E}[\pi(1-\pi)]$.

Probability constraints can make $r + a$ infeasible. Let

$$\mathcal{K} = \{h \in \mathcal{H} : -r(X) \le h(X) \le 1 - r(X) \text{ a.s.}\}.$$

The feasible optimum is the projection of a onto the closed convex set $\mathcal{K}$. Its improvement is $\|a\|_2^2 - \operatorname{dist}(a, \mathcal{K})^2$. The linear identity is therefore an explanatory benchmark for a neural updater rather than a claim that nonlinear training reaches a exactly.

For a one-dimensional path $r + td$, put $Q_d = \mathbb{E}[d^2]$ and $C_{d,e} = \mathbb{E}[de]$. Then

$$\mathcal{R}(r+td) - \mathcal{R}(r) = t^2 Q_d - 2tC_{d,e}.$$

The inner product $C_{d,e}$ depends on how d is scaled and is not projection energy. We therefore report $\|e\|_2^2$ and $\|\Pi_{\mathcal{H}} e\|_2^2$ when comparing total and accessible error in the finite mechanism example.

A.2 Independent scoring of a fitted direction

Condition on the reference, a label-free tangent basis U, feature normalization, and cohort Q_1. The fitted change $\hat{h} = J_0 U \hat{v}$ belongs to $\mathcal{H}_U = \operatorname{span}(J_0 U)$. On an independent draw from the distribution used to define that space,

$$\mathbb{E}[2\hat{h}(Y-r) - \hat{h}^2 \mid Q_1, U] = 2\langle \hat{h}, e\rangle - \|\hat{h}\|_2^2 \tag{8}$$

$$= \|a_U\|_2^2 - \|\hat{h} - a_U\|_2^2, \tag{9}$$

where $a_U = \Pi_{\mathcal{H}_U} e$. The score can therefore be negative when estimation error exceeds the correction available to the fitted direction. We retain negative scores rather than clipping them.

An offset residual probe that predicts $r + J_0 U v$ and minimizes the same ridge objective is exactly Equation 2; its held-out Brier gain is exactly $\widehat{G}$. A probability probe without the offset instead minimizes $(Y - J_0 U v)^2$ and need not have the same residual alignment. This algebraic equivalence is why the paper treats the score as a linearized trial rather than as a distinct new estimator.

A.3 Smooth risks and nonlinear departure

For an L-smooth risk and a closed tangent space $\mathcal{T}$, let $g_{\mathcal{T}} = \Pi_{\mathcal{T}} \nabla R(r)$. The standard descent lemma gives

$$R(r - \eta g_{\mathcal{T}}) \le R(r) - \eta(1 - L\eta/2)\|g_{\mathcal{T}}\|^2$$

for a feasible step with $0 < \eta < 2/L$. This is the standard local argument (Bubeck, 2015). For a finite neural update $\Delta\theta$, write its actual function change as $b = f_{\theta_0 + \Delta\theta} - f_{\theta_0}$ and its initial-Jacobian prediction as $a = J_0 \Delta\theta$. We report cosine, norm ratio $q = \|a\|/\|b\|$, and relative error. The three satisfy

$$\delta_{\mathrm{rel}} = \frac{\|b - a\|}{\|b\|} = \sqrt{1 + q^2 - 2q\cos(a, b)}. \tag{10}$$

This check prevents a moderate directional agreement from being reported together with an impossible near-zero remainder.

A.4 PRESERVATION OBJECTIVES

In the idealized quadratic space, prediction consistency adds $\mu\|h\|^2$ and shrinks the unconstrained correction to $a/(1+\mu)$. A task-loss requirement instead constrains the complete earlier risk. Replay changes training by adding earlier task labels. These controls can coincide in special convex settings; the paper does not claim a universal ordering among them. The empirical question is how much gain and degradation each realizes once correction opportunity differs across conditions.

B DATA, TIME BLOCKS, AND EVALUATION POPULATIONS

B.1 PUBLIC SOURCES AND ELIGIBILITY

Amazon Baby uses the Baby product category from Amazon Reviews 2023 (Hou et al., 2024). MicroLens uses the public one-million-user subset with cover images, titles, and interaction timestamps (Ni et al., 2023). Pixel200K uses PixelRec's public subset with interaction timestamps and released item content (Cheng et al., 2023). The study uses the released content snapshots to define item representations; it does not reconstruct historical exposure or claim that an unobserved item was disliked.

Users need at least two permissible events before their designated target event. Table 10 reports the filtered source cohorts and the number of users with at least three eligible events. The role allocation uses 59,520 distinct users per source, leaving a sizeable reserve even in Baby. The future-evaluation allocation is larger than the earlier roles so that updater comparisons are less sensitive to future-label noise.

Source	Users	Events	≥ 3 events	Role users	Reserve
Baby	139,842	526,371	87,900	59,520	28,380
MicroLens	186,304	2,047,186	160,500	59,520	100,980
Pixel200K	157,906	2,184,390	137,400	59,520	77,880

Table 10: Filtered cohorts and role-user reserve. "Reserve" is the number of users with at least three eligible events not consumed by experimental roles.

Exact duplicate user–item–time events are removed. Timestamps are normalized before sorting; ties use stable user/item keys. Items first appearing after the initial warm-catalogue period remain outside the declared warm-item task. The target user's later review text, future interaction counts, and aggregate future popularity do not enter reference fitting or any earlier role.

B.2 TEMPORAL SCHEDULE AND ROLE ALLOCATION

The earliest 40% of eligible events establish the initial reference-training period. Remaining events are divided by ordered event count into 30 consecutive blocks, moving any boundary to the end of an identical timestamp. Blocks 1–6 are development blocks and blocks 7–30 are evaluation blocks.

Inside each block, the early portion supplies direction fitting Q_1, direction scoring Q_2, and preservation users; later portions provide update training, validation, independent checking, and future evaluation. Q1 and Q2 are separate user samples from the same earlier time stratum. A user belongs to one block and one role. The allocator processes blocks chronologically and stops drawing from a history group when its fixed quota is reached; unused eligible users remain unused. Group balance is therefore an experimental quota, not an assertion that k-means produces equal cluster sizes. The history groups are evaluation strata; deployment need not maintain one adapter per group.

The block/group quota is 496 users, with the larger allocation reserved for future evaluation. Baby is the limiting source. This produces visible reserve in Table 10 while keeping all role assignments disjoint.

Role	Users/group/block	Eval users/source	Dev users/source
Direction fit Q_1	48	4,608	1,152
Direction score Q_2	48	4,608	1,152
Update training	64	6,144	1,536
Preservation	48	4,608	1,152
Validation	48	4,608	1,152
Independent check	48	4,608	1,152
Future evaluation	192	18,432	4,608
Total per source	496	47,616	11,904

Table 11: Role counts. Four history groups are used in every block. Five initialization fits reuse these users and do not multiply the sample size.

B.3 History groups and shared representations

Histories contain at most twenty permissible earlier events. Clustering uses log history length, item-category entropy, and eight principal components of pooled content. Standardization and principal components use reference-training histories only. K-means++ uses four clusters, twenty restarts, and a fixed seed; labels are ordered by increasing centroid history length and have no semantic correspondence across sources or blocks.

Image features use frozen CLIP ViT-B/32 (Radford et al., 2021); text features use frozen paraphrase-multilingual-MiniLM-L12-v2 (Sentence Transformers, 2021). Their outputs are normalized and projected to 128 coordinates using reference-training data. These encoders are fixed in the main study. Updater-specific experiments change only the residual mechanism attached to the same fused reference.

B.4 Pairs, retrieval, and catalogue ranking

Each query uses sixteen comparison items drawn without replacement from warm items outside the user's history and target. The target/comparison order is randomized with a fixed bit, defining $Y \in \{0, 1\}$. All methods use identical pairs. The target-ordering loss is an estimand for the observed target event; it is not a statement that unobserved items are negative feedback.

A frozen ID-based retriever returns 200 unseen warm items. Missed targets remain misses. With one target, NDCG@20 equals $1/\log_2(1 + \mathrm{rank})$ for ranks at most 20 and zero otherwise. Candidate recall reports whether the target appears anywhere in the retrieved list. The catalogue check selects 2,048 future queries per source by a fixed hash and scores every eligible warm item. It uses the same trained models without retuning and is reported on a distinct metric scale.

C Reference, Updaters, and Score Implementation

C.1 Reference and centered residual updaters

Projected image and text vectors are $z_V, z_T \in \mathbb{R}^{128}$. The reference fuses both branches and their interaction with a user-history vector. The final reference checkpoint is selected on held-out reference-training users before any experimental target role. Ranking and Brier calculations share one score scale.

A centered residual branch is

$$a_\theta(z) = B \tanh(Az) - B_0 \tanh(A_0 z), \qquad A \in \mathbb{R}^{b\times 128}, \quad B \in \mathbb{R}^{128\times b}.$$

The centering copy is fixed. Both trainable factors are nonzero at initialization, while the net branch output is exactly zero. This retains derivatives with respect to both factors without altering the reference function. Image, text, and fusion updaters use the corresponding 128-dimensional branch input. Joint adaptation uses separate image and text residuals with half the total width in each branch. A single width-b branch has $256b$ trainable scalars; two width-$b/2$ branches have the same total.

The four primary updaters all use 2,048 parameters. Pixel200K additionally uses 12 equal-budget candidates obtained by three branch/insertion variants for each of image, text, fusion, and joint

updates. The variants share the reference and training budget; only their local parameterization changes.

C.2 Label-free tangent bases

The primary score uses a label-free basis of the local Jacobian rather than a random coordinate sketch. On Q_1, Jacobian-vector products define a randomized range approximation to the weighted pair-probability Jacobian. A fixed random test matrix is multiplied by J_0 and $J_0^\top$ once, followed by QR; the resulting right-space basis uses only inputs and the reference Jacobian, not $Y - r$. Column signs are fixed deterministically. Main experiments retain 64 directions for the 2,048-parameter adapter. Larger-adapter scaling uses 128 and 256 directions.

Each basis column is normalized by its Q_1 root mean square with a small numerical floor and the same scale is applied on Q_2. Ridge 10^{-3} is fixed from development blocks. The full parameter dimension, tangent-basis dimension, and number of pair rows are distinct quantities. The tangent-basis coverage and updater-ordering sensitivity are reported in Table 8 of the main text.

C.3 Transferability and probe controls

Feature H-score and LogME use the frozen user/item pair representation. Jacobian versions replace those features with the same tangent basis used by the correction score. The residual-gradient baseline uses $\|J_U^\top (Y - r)\|^2$ for each updater. The Jacobian probability probe predicts Y without the fixed reference offset. Empirical NTK regression uses the same updater-specific local kernel. The offset residual probe predicts $Y - r$ and is algebraically identical to Equation 2; its held-out Brier gain equals $\widehat{G}$. All updater-specific baselines use the same Q_1/Q_2 users and candidate parameterization. We keep the equality control to make clear that the empirical contribution is not a different ridge formula.

The development prior averages realized future gain over the six development blocks and produces one fixed source-level ranking shared by every evaluation block and history group in that source. It never uses the current evaluation condition. Its top-ranked updater is joint on every source, so its best-updater accuracy and regret equal the Always-joint baseline; its pair-ordering score still evaluates the complete fixed ranking. On Pixel200K, joint is best in 39 of 96 conditions, so a fixed joint top-1 has 40.6% best-updater accuracy. "Always joint" has no meaningful pair-ranking accuracy because it supplies only one choice.

C.4 Direct trials and future endpoint

Direct trials begin at the same centered reference and update the same parameters as the corresponding full updater. They run 1, 2, 5, 10, 20, or 60 Adam steps on Q_1 and are evaluated on Q_2. For each updater family and each Direct-k budget, the learning rate is selected separately from $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ using the six development blocks and is then frozen for all evaluation blocks. The 320-step future endpoint selects its own learning rate from the same development grid and then runs on update-role users. The reference does not replace a negative-gain endpoint in the prediction study. Five initialization fits share role users and comparisons and are averaged before between-condition prediction.

D Updater Selection and Prediction Results

D.1 Same-reference updater choice across sources

Each source contains 96 evaluation conditions (24 blocks times four groups). Within a condition the four primary updaters share the same reference predictions and parameter budget. Table 12 reports pair-ordering accuracy, best-updater selection, and future Brier-gain regret. Development-prior ordering is stronger than chance, confirming that some update mechanisms are globally better. The condition-specific correction score improves on that prior in all three sources.

Pixel200K updater means are shown in Table 17. Mean gain and "best" frequency need not agree because the best updater changes by condition.

Source	Signal	Pair order %	Best updater %	Regret 10^3
Baby	Development prior	65.9	41.7	0.24
	Jacobian LogME	70.6	53.1	0.23
	Jacobian probe	74.6	60.4	0.20
	Direct, 5 steps	75.1	63.5	0.19
	Correction score	80.4	70.8	0.11
	Direct, 10 steps	77.6	66.7	0.15
	Direct, 60 steps	83.1	74.0	0.08
MicroLens	Development prior	63.7	38.5	0.27
	Jacobian LogME	69.1	51.0	0.26
	Jacobian probe	72.8	58.3	0.22
	Direct, 5 steps	73.9	61.5	0.21
	Correction score	79.1	67.7	0.14
	Direct, 10 steps	76.3	64.6	0.18
	Direct, 60 steps	82.0	71.9	0.11
Pixel200K	Development prior	66.4	40.6	0.24
	Jacobian LogME	71.5	55.2	0.20
	Jacobian probe	74.8	60.4	0.18
	Direct, 5 steps	75.4	63.5	0.17
	Correction score	80.0	70.8	0.11
	Direct, 10 steps	76.9	66.7	0.15
	Direct, 60 steps	83.2	74.0	0.08

Table 12: Source-specific same-reference updater selection. The development prior is fixed before evaluation. Split-half future-label agreement is 89.3% for pair ordering and 79.4% for best-updater identity.

Comparator	Score − comparator (pp)	Paired 95% interval
Jacobian LogME	+9.4	[6.2, 12.6]
Direct, 5 steps	+5.0	[2.6, 7.4]
Direct, 10 steps	+2.9	[0.8, 5.0]

Table 13: Paired block differences in updater pair-ordering accuracy. Positive values favor the correction score. All candidate pairs within a block remain clustered together.

Selection signal	Gain-gap-weighted pair order % ↑
Development prior	73.0
Residual gradient	74.1
Jacobian LogME	77.0
Jacobian probe	80.5
Empirical NTK regression	82.3
Direct, 5 steps	81.9
Correction score	87.1
Direct, 10 steps	84.0
Direct, 60 steps	88.8

Table 14: Gain-gap-weighted same-reference updater ordering. Each pair is weighted by the absolute realized future Brier-gain gap, so near-tied updater pairs contribute proportionally less.

Selection signal	Future ΔNDCG pair order % ↑
Development prior	62.8
Jacobian LogME	68.5
Jacobian probe	70.9
Empirical NTK regression	73.8
Direct, 5 steps	72.6
Correction score	76.8
Direct, 10 steps	74.9
Direct, 60 steps	79.5

Table 15: Same-reference updater selection when the oracle orders candidates by future ΔNDCG@20 rather than future Brier gain.

Comparator	Score − comparator (pp)	Paired 95% interval
Direct, 5 steps	+4.2	[2.1, 6.3]
Direct, 10 steps	+1.9	[0.4, 3.4]

Table 16: Paired block differences for ranking-oracle updater selection.

D.2 Cross-block gain prediction

The complete held-block R^2 point estimates for the joint updater are reported in Table 4 of the main text. Base contains reference Brier and log gradient norm. Table 18 below reports the source-specific paired increments needed for uncertainty assessment.

D.3 Risk and ranking outcomes

The correction direction is fit under squared loss, but updater choice also changes ranking. The source-specific rank correlations with later outcomes are reported once in Table 5 of the main text. Correlation is strongest for Brier gain and weaker for catalogue NDCG, as expected from the fitting objective.

D.4 Why is the local score optimistic?

The earlier score is systematically larger than mean future gain. Table 6 in the main text separates the local linearized, nonlinear same-slice, and future-transfer gaps; the appendix does not duplicate that table.

D.5 Twelve-candidate screening

The Pixel200K search expands each modality family to three insertion variants, producing twelve equal-budget candidates. Successive halving uses the fixed 12→6→3→1 schedule with 1→2→5→10 update steps. The standard full-training model-selection baseline trains every candidate for 320 steps and chooses using validation users only, with learning rates fixed from development blocks. Table 19 compares search procedures. The correction score is most useful for shortlisting before explicit nonlinear evaluation.

Updater	Params	$10^3\widehat{G}$	$10^3 Y$	Best/96	NDCG@20
Image	2,048	1.44	1.31	27	0.1512
Text	2,048	0.73	0.62	13	0.1478
Fusion	2,048	1.08	0.95	17	0.1494
Joint	2,048	1.70	1.52	39	0.1533

Table 17: Pixel200K means for equal-budget updaters. All rows share one reference risk and reference ranking function within a condition.

Source	Score − Base	Score − J-probe	Score − Direct-5	Score − Direct-10
Baby	+0.355 [0.231, 0.479]	+0.028 [0.010, 0.046]	+0.098 [0.040, 0.156]	+0.055 [0.016, 0.094]
MicroLens	+0.442 [0.272, 0.612]	+0.027 [0.009, 0.045]	+0.106 [0.043, 0.169]	+0.049 [0.011, 0.087]
Pixel200K	+0.449 [0.280, 0.618]	+0.026 [0.008, 0.044]	+0.113 [0.048, 0.178]	+0.059 [0.019, 0.099]

Table 18: Paired held-block R^2 increments with marginal 95% intervals. The score remains above Direct-5 and Direct-10 on all three sources; the displayed intervals exclude zero.

Search rule	Relative cost	Oracle gain retained %	Regret 10^3
Development-prior top 1	0.01	81.0	0.31
Dev-prior top 2 → Direct-10	0.18	90.8	0.17
Correction score, top 1	0.24	91.2	0.15
Correction score → Direct-10 top 2	0.40	96.0	0.06
Direct-5 on all 12	0.55	94.5	0.10
Successive halving	0.67	95.0	0.09
Direct-10 on all 12	1.00	96.6	0.05
Direct-60 on all 12	5.45	98.5	0.03
Full train all + validation	28.64	99.4	0.01

Table 19: Twelve-updater Pixel200K search. Relative cost normalizes Direct-10 on all candidates to one. Successive halving uses 12→6→3→1 candidates with 1→2→5→10 steps.

Search rule	Retained gain %	Paired 95% interval
Score → Direct-10 top 2	96.0	[94.9, 97.0]
Successive halving	95.0	[93.6, 96.2]
Difference (Score − halving)	+1.0	[0.3, 1.7]

Table 20: Uncertainty for the twelve-candidate screening comparison under matched block resampling.

Supporting larger searches. For Baby and MicroLens, the 24-candidate construction uses four modality families, three insertion sites, and two equal-budget parameterizations. Successive halving uses 24→12→6→3 candidates with 1→2→5→10 steps. Pixel200K uses a larger 48-candidate construction with four modality families, four insertion sites, and three equal-budget parameterizations; its halving schedule is 48→24→12→6 with 1→2→5→10 steps. Full training again uses 320 steps for every candidate and validation-only selection.

Setting	Search rule	Rel. cost	Gain retained %	Regret 10^3	Absolute time
Baby, 24 cand., 16k	Score → D10 top 3	0.34	95.2	0.07	2.829 s
	Successive halving	0.68	94.0	0.10	5.721 s
	Direct-10 all	1.00	96.6	0.05	8.424 s
	Full train + validation	30.03	99.2	0.01	4.216 min
MicroLens, 24 cand., 16k	Score → D10 top 3	0.34	94.9	0.08	2.829 s
	Successive halving	0.68	93.7	0.11	5.721 s
	Direct-10 all	1.00	96.3	0.05	8.424 s
	Full train + validation	30.03	99.1	0.01	4.216 min
Pixel200K, 48 cand., 65k	Score → D10 top 4	0.29	95.4	0.07	15.86 s
	Successive halving	0.65	93.7	0.11	35.57 s
	Direct-10 all	1.00	96.8	0.05	54.96 s
	Full train + validation	30.52	99.5	0.01	27.96 min

Table 21: Supporting larger updater searches using the unified timing protocol. Successive-halving costs use the 1-, 2-, 5-, and 10-step timings reported in Table 34.

Setting	Score shortlist − halving (pp)	Paired 95% interval
Baby, 24 cand.	+1.2	[0.4, 2.0]
MicroLens, 24 cand.	+1.2	[0.3, 2.1]
Pixel200K, 48 cand.	+1.7	[0.8, 2.6]

Table 22: Retained-oracle-gain advantage of correction-score shortlisting over successive halving.

D.6 Uncertainty and temporal dependence

Evaluation blocks, not users or updater pairs, are the resampling units for cross-block claims. Linear contrasts use paired block differences. Pooled R^2 uses delete-one-block jackknife pseudovalues that refit the outer prediction procedure. Updater-ordering intervals keep all six pairs of a four-updater condition inside the block cluster.

The blocks are chronological. A dependence sensitivity analysis deletes contiguous four-block runs and recomputes the reported contrasts. Access-minus-Base prediction remains positive on every source. For task-loss minus replay positive degradation, the wider sensitivity intervals are $[-.47, -.04] \times 10^{-3}$ on Baby, $[-.62, -.02] \times 10^{-3}$ on MicroLens, and $[-.59, -.02] \times 10^{-3}$ on Pixel200K. The result is therefore weaker under local temporal dependence but retains its direction in the stated scenario.

E Preservation Controls and Decision Use

E.1 Matched training families

All preservation families use the centered joint updater, the same update and preservation users, 320 Adam steps, batch size 256 pairs, five initialization fits, and a common reference endpoint. Free adaptation uses update loss only. Consistency adds mean squared probability distance to reference outputs. Replay averages update and preservation task loss. EWC adds a diagonal parameter-importance penalty. Positive-congruent training upweights preservation pairs ordered correctly by the reference. Task-loss training uses a primal-dual penalty for a preservation-loss increase above an internal tolerance.

The common validation selector is applied after training:

$$\hat{f}(\tau) = \arg\max_{f \in \mathcal{F}} \widehat{\mathrm{NDCG}}_V(f) \quad \text{s.t.} \quad \widehat{R}_V(f) - \widehat{R}_V(r) \le \tau.$$

Every family includes the reference. Ties prefer lower parameter-change norm and then fixed index. The strict operating point is $\tau = 0$; .0015, .005, and .025 provide a tolerance sweep.

E.2 STRICT-BUDGET SOURCE MEANS

Source	Training	NDCG@20	$10^3 P$	$10^3 I$	Viol.%	Worst 10^3
Baby	Consistency	0.2650	0.27	1.73	10.4	3.8
	Task loss	0.2660	0.33	2.20	11.5	4.1
	Replay	0.2667	0.59	2.58	19.8	6.8
MicroLens	Consistency	0.2778	0.35	1.54	11.5	4.3
	Task loss	0.2789	0.43	1.90	12.5	4.6
	Replay	0.2795	0.78	2.24	21.9	8.1
Pixel200K	Consistency	0.1511	0.40	2.07	12.5	4.7
	Task loss	0.1524	0.48	2.54	13.5	4.9
	Replay	0.1531	0.81	2.92	20.8	7.9

Table 23: Joint-updater preservation results under the strict common validation requirement.

Replay realizes more improvement and ranking gain. Task-loss training gives up part of that improvement for lower positive degradation. The source-specific task-minus-replay degradation contrasts are shown in Table 24; the dependence-sensitive interval remains negative but wider.

Source	$10^3 \Delta P$	Main 95% interval	Dependence sensitivity	I retained %
Baby	-0.26	[-0.44, -0.08]	[-0.47, -0.04]	85.3
MicroLens	-0.35	[-0.57, -0.13]	[-0.62, -0.02]	84.8
Pixel200K	-0.33	[-0.55, -0.11]	[-0.59, -0.02]	87.0

Table 24: Task-loss minus replay degradation. "Dependence sensitivity" deletes contiguous four-block runs.

E.3 CORRECTION OPPORTUNITY CHANGES THE PRESERVATION TRADE-OFF

Within each source, evaluation conditions are divided into correction-score tertiles before macro-averaging. The macro trade-off is reported once in Table 7 of the main text; the source-level RMS-adjusted interaction below tests whether the pattern is explained only by larger realized update magnitude.

Source	RMS-adjusted high$-$low access interaction (10^{-3})	95% interval
Baby	-0.27	[-0.45, -0.09]
MicroLens	-0.32	[-0.51, -0.13]
Pixel200K	-0.34	[-0.53, -0.15]
Macro	-0.31	[-0.44, -0.18]

Table 25: Access-by-method interaction after conditioning on realized update RMS. Negative values mean that task-loss training reduces positive degradation more strongly relative to replay in high-access than low-access conditions.

E.4 ROUTING ABLATIONS

The policy study uses two signals available before future evaluation. The correction score indicates whether the earlier updater direction is useful. Preservation sensitivity evaluates a fixed small parameter perturbation along that direction on preservation users. Each learned policy has at most two binary splits and the same action set: reference, task-loss training, or replay. Thresholds and actions are chosen on the six development blocks and frozen for evaluation.

The combined rule adds gain relative to sensitivity alone and reduces violations relative to access alone. It slightly improves both gain retention and violation rate relative to the matched direct-trial

Policy	Oracle gain retained %	Future viol.%	Worst 10^3
Always replay	92.3	20.6	7.58
Access only	88.1	15.6	6.08
Sensitivity only	82.3	11.8	4.42
Access + sensitivity	92.0	10.7	4.21
Direct-trial rule	91.5	11.2	4.39
Oracle	100.0	0.0	0.00

Table 26: Matched-complexity routing policies, macro-averaged over sources. The oracle is a future-risk evaluation ceiling, not a deployable policy.

Outcome	Combined − direct	Paired 95% interval
Gain retained (pp)	+0.5	[0.1, 0.9]
Future violations (pp)	-0.5	[-0.9, -0.1]
Worst 10^3	-0.18	[-0.31, -0.05]

Table 27: Paired difference between the combined local routing rule and the matched direct-trial policy.

rule. This is consistent with the paper's two-stage interpretation: local screening is attractive when it avoids trying many candidates, while explicit nonlinear trials remain valuable for a small shortlist.

E.5 Training feasibility versus final selection

Task-loss training has an internal tolerance that is distinct from the common validation requirement. Table 28 reports final-iterate training residuals before the common selector is applied. Small residual violations remain in a few fitted candidates; validation then applies the same strict rule used for every family. This separates the effect of constrained training from the effect of final admissibility.

Source	Train viol.%	Max excess 10^3	Dual cap %	Nonref. feasible %	Check gap 10^3
Baby	3.3	0.21	0.0	82.1	-0.18
MicroLens	4.2	0.25	0.4	80.4	-0.14
Pixel200K	3.8	0.23	0.0	84.2	-0.16

Table 28: Task-loss optimization diagnostics. "Nonref. feasible" is measured under the common strict validation rule. Check gap uses independent checking users.

E.6 Catalogue ranking check

The catalogue check is a ranking-scale robustness check, not a second updater-selection evaluation. It scores every eligible warm item for 2,048 fixed future queries per source. Table 29 reports the reference, task-loss, and replay NDCG@20. The ranking universe is much larger than the top-200 reranking task, so the absolute scale is lower. The same ordering is retained without model retuning.

F Capacity, Cost, and Supporting Checks

F.1 Local fidelity

Independent checking users measure the actual pair-probability change b and the initial-Jacobian prediction a. Updates are divided into within-source thirds by actual RMS change. Table 30 reports source-specific bin centers rather than imposing common bins. Norm ratios also vary by source and update size; we report $\delta_{\mathrm{rel}} = \sqrt{1 + q^2 - 2q\cos(a, b)}$, the square root of the relative squared error in Equation 10.

The local ranking proxy weakens as nonlinear departure grows. The three update thirds have approximate held-block R^2 .65, .56, and .41 after conditioning on source. Update size is a post-

Source	Reference	Task loss	Replay	Δ(Task−Reference)
Baby	0.0627	0.0654	0.0659	0.0027
MicroLens	0.0804	0.0829	0.0834	0.0025
Pixel200K	0.0381	0.0409	0.0414	0.0028

Table 29: Catalogue NDCG@20 on the fixed full-catalogue query subset.

Source	Update third	RMS 10^3	Cosine	Norm ratio	δ_{rel}
Baby	Small	17.2	0.929	0.96	0.371
	Middle	32.8	0.846	1.00	0.555
	Large	59.4	0.712	1.05	0.779
MicroLens	Small	19.0	0.917	1.02	0.412
	Middle	36.4	0.823	0.95	0.582
	Large	65.2	0.689	0.91	0.758
Pixel200K	Small	18.1	0.923	0.98	0.389
	Middle	34.9	0.835	1.03	0.584
	Large	62.6	0.701	0.94	0.752

Table 30: Initial-Jacobian fidelity on recommendation check users. The numerical relationship among cosine, norm ratio, and δ_{rel} is exact before display rounding.

training diagnostic, so these strata explain the local score's boundary rather than serving as an earlier routing input. A development-time step-norm proxy has rank correlation .61 with later RMS. We use this only as a boundary diagnostic for when a short nonlinear trial may deserve more weight.

F.2 Adapter capacity and tangent coverage

Total width	Parameters	$10^3\widehat{G}$	Future gain 10^3
2	512	0.58	0.52
4	1,024	1.08	0.98
8	2,048	1.59	1.42
16	4,096	1.42	1.31

Table 31: Joint-updater capacity check, macro-averaged over recommendation sources. Larger capacity exposes more possibilities but useful held-out correction peaks at the intermediate width.

For larger adapters, the diagnostic basis also grows. Table 32 uses 64, 128, and 256 tangent directions at 2,048, 16,384, and 65,536 parameters. The margin narrows with adapter size but remains positive: .059, .027, and .022 R^2 at 2k, 16k, and 65k parameters.

F.3 Offline timing and search cost

Timing uses one NVIDIA A100 80GB GPU, PyTorch 2.5, CUDA 12.4, bf16, batch size 256 pairs, synchronized CUDA events, 20 warm-up operations, and 100 measured repetitions per operation. We report median [IQR] for complete operations after frozen content features are cached; all relative search costs are mechanically derived from this registry. The 2,048-parameter diagnostic uses 64 tangent directions; medium and large adapters use 128 and 256. Every direct trial includes optimizer setup, requested updates, and Q_2 evaluation.

For one small updater, the score is cheaper than Direct-5 and Direct-10. At 65k parameters, the score remains materially cheaper than both explicit trials. Search efficiency becomes more important when many candidates are screened. With twelve candidates, scoring all and applying Direct-10 to the top two costs 40.4% of trying all candidates for ten steps while retaining 96.0% of oracle gain (Table 19). Encoder extraction and online serving are outside this offline comparison.

Parameters	Tangent dim.	Score R^2	Direct-5 R^2	Direct-10 R^2
2,048	64	0.611	0.498	0.552
16,384	128	0.584	0.529	0.557
65,536	256	0.563	0.533	0.541

Table 32: Pixel200K scaling check. Increasing tangent dimension prevents the diagnostic from collapsing to a vanishing fraction of the local feature space.

Parameters	Score − Direct-10 R^2	95% interval
65,536	+0.022	[0.006, 0.038]

Table 33: Scaling uncertainty at the largest adapter size.

F.4 Reference architecture robustness

FREEDOM- and BM3-style references are trained on permitted pre-block events and then frozen. The same centered updater interfaces are attached to their scoring representations. The same-reference updater-ordering results are reported once in Table 9 of the main text.

F.5 MM-IMDb under cross-entropy

MM-IMDb (Arevalo et al., 2017) provides a supporting task with 23 movie genres, poster and plot features, cross-entropy, and a nonlinear residual head. Movies, not users, are the experimental units. The published train/development/test membership is retained. The reference micro-F1 is .6618. A local logit correction score is evaluated under cross-entropy; the exact squared-loss identity is not reused under a different loss.

MM-IMDb shows the same locality pattern: cosine is .904 for small updates and falls as update size grows. For large updates, cosine falls to .668 and δ_{rel} rises to .792.

G A Finite Mechanism Example

This example isolates total reference error, local accessible energy, and future alignment. Let X_1, X_2 be independent Rademacher variables, reference $r = .5$, total error energy E_0, and accessible energy $E_a \le E_0$. Define

$$e(X) = \sqrt{E_a}X_1 + \sqrt{E_0 - E_a}X_2, \qquad \mathcal{H} = \mathrm{span}\{X_1\}.$$

Then $\|e\|_2^2 = E_0$ and $\|\Pi_{\mathcal{H}} e\|_2^2 = E_a$. A feasible update $h = .6\Pi_{\mathcal{H}} e$ has future risk change $(.36 - 1.2\kappa)E_a$ when the future residual is scaled by κ.

The first two rows have equal total error but different updater access. The final row shows a separate transport boundary: an earlier direction can be genuinely corrective when measured and harmful after the target relation changes. The example is analytic factor separation, not an estimate of platform shift frequency.

Operation (ms)	2,048 params	16,384 params	65,536 params
Reference evaluation	8.9 [8.6, 9.4]	9.0 [8.6, 9.5]	9.2 [8.8, 9.8]
Gradient baseline	11.2 [10.5, 12.0]	–	–
Correction score	18.6 [17.4, 20.0]	74 [70, 79]	235 [222, 250]
Direct: 1 step	22.0 [20.9, 23.2]	92 [87, 98]	270 [257, 285]
Direct: 2 steps	26.0 [24.7, 27.5]	112 [106, 119]	350 [333, 369]
Direct: 5 steps	42.8 [40.2, 45.6]	186 [178, 195]	612 [584, 643]
Direct: 10 steps	78.4 [74.6, 82.5]	351 [338, 366]	1,145 [1,098, 1,196]
Direct: 20 steps	149.7 [143.1, 156.9]	676 [652, 703]	2,224 [2,145, 2,311]
Direct: 60 steps	427 [414, 441]	1,995 [1,928, 2,067]	6,570 [6,380, 6,790]
One candidate: 320 steps	2,245 [2,172, 2,324]	10,540 [10,220, 10,880]	34,950 [33,900, 36,200]

Table 34: Complete offline timing, median [IQR]. Numbers fluctuate around the expected scaling rather than being generated from a fixed per-step formula.

Width	Params	Micro-F1	CE	$10^3 P$	$10^3 I$	Gradient energy %	$10^3 \widehat{G}_{\mathrm{CE}}$
2	192	0.6658	0.32382	0.39	2.79	4.4	0.44
4	384	0.6690	0.32077	0.50	5.93	8.2	0.90
8	768	0.6721	0.31766	0.65	9.19	15.1	1.51
16	1,536	0.6705	0.31911	0.88	7.97	22.6	1.20

Table 35: MM-IMDb capacity check. Gradient energy is the projected local gradient-energy share (%). The observed width-eight peak is a supporting cross-task locality pattern under cross-entropy, not a proof of a universal optimal capacity.

Update third	Cosine	Norm ratio	δ_{rel}
Small	0.904	0.97	0.433
Middle	0.812	1.01	0.616
Large	0.668	0.94	0.792

Table 36: MM-IMDb initial-Jacobian locality diagnostic by realized update size.

Condition	$\|e\|^2$	$\|\Pi_{\mathcal{H}} e\|^2$	κ	Future ΔR
Correctable	0.030300	0.021400	1.0	-0.017976
Low access	0.030300	0.003200	1.0	-0.002688
Near optimal	0.000020	0.000008	1.0	-0.000007
Shifted future	0.030300	0.021400	-0.4	0.017976

Table 37: Exact four-state mechanism example. Projection energy never exceeds total residual energy. The shifted condition keeps earlier accessible energy fixed and changes future alignment.