CROSS-EPOCH INTERACTIONS IN TRAINING DATA ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training data attribution predicts how modifying training data affects a model and helps practitioners choose corrections without retraining each candidate. Influence functions approximate retraining through inverse curvature at a unique optimum, although their predictions can differ substantially from deep-network retraining. Unrolled differentiation avoids this assumption by following the recorded optimization steps. Practical approximations reduce cost by shortening trajectories or averaging within training segments. However, these methods do not explicitly quantify how visits in different epochs combine. Propagated visit contributions can reinforce or cancel in the terminal response. We quantify these interactions through the finite-horizon stability dimension (FHSD), the average squared scaled replacement derivative of the final parameters. This quantity decomposes exactly over visit pairs and, under random reshuffling, is tightly bounded by the epoch count times the sum of single-visit energies. Building on this decomposition, we propose PATHGRAM, which propagates each candidate's visit contributions through a shared leading Hessian eigenspace to predict signed loss changes. On randomly reshuffled BERT-tiny and BERT-base, cross-epoch pairs account for about 96% and 94% of the mean squared linear prediction, respectively. In the primary BERT and Swin-T experiments, PATHGRAM achieves normalized mean absolute errors (NMAEs) of 0.071–0.106 and recovers both full-batch BERT oracle top-six sets. We further refine the approximate neighboring path with four gradient measurements. On independent confirmation runs, this refinement reduces NMAE from 0.1034 to 0.0027 for Swin-T feature adaptation and from 0.0741 to 0.0023 for BERT-tiny LoRA.

1 INTRODUCTION

Training data attribution (TDA) links model behavior to training data, supporting error correction and prediction of dataset counterfactuals (Koh & Liang, 2017; Hara et al., 2019; Ilyas et al., 2022). We predict how replacing one training example changes evaluation loss under fixed initialization and data order.

Influence functions approximate retraining through inverse curvature at an optimum (Koh & Liang, 2017), although deep-network predictions can differ substantially from retraining (Basu et al., 2021; Bae et al., 2022; Schioppa et al., 2023). Unrolled differentiation follows the optimization steps and accounts for the recorded training trajectory (Maclaurin et al., 2015; Hara et al., 2019; Mlodozeniec et al., 2025). Practical estimators summarize this history differently: TracIn accumulates checkpoint gradient similarities (Pruthi et al., 2020), while SOURCE approximates unrolling through segment-level gradient and curvature statistics (Bae et al., 2024).

Temporal attribution resolves trajectory-specific contributions through data value embeddings (Wang et al., 2025b). Stagewise attribution further identifies changes in influence, including sign reversals, during learning (Lee et al., 2026). Multipass stability analysis also accounts for correlations induced by data reuse (Vary et al., 2026). A replacement applied throughout training couples these visits in a shared terminal response. We therefore ask how strongly they reinforce or cancel at the final parameters.

Each visit to a replaced example changes a training update. Later updates carry that change to the final model, where contributions from different visits can reinforce or cancel. Consequently, the

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.

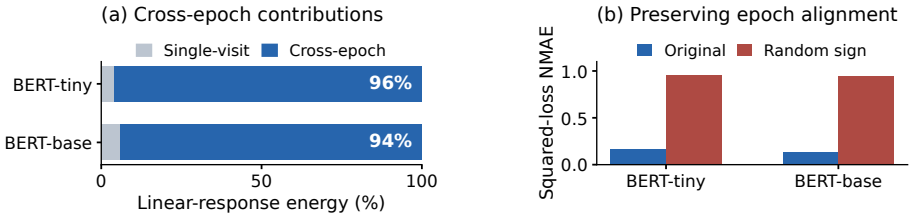


Figure 1: Cross-epoch terms dominate the mean squared linear prediction under random reshuffling. (a) Estimated linear-energy decomposition, averaged over candidates and schedules. (b) Randomizing epoch signs increases the NMAE of squared quadratic predictions (Section 6.2).

combined effect depends on both the size of each contribution and their alignment. We quantify this effect with the *finite-horizon stability dimension* (FHSD), which averages squared replacement responses over candidates. Its visit-pair decomposition separates individual contributions from their interactions across epochs (Section 4).

This decomposition motivates PATHGRAM, which retains each candidate's gradient changes throughout training. Candidates share one curvature approximation, allowing a single recorded training run to support many replacement predictions (Section 5). Four additional gradient measurements refine the approximate replaced path and improve predictions for complete replacements.

Our main contributions are as follows:

- We derive an exact visit-pair decomposition and bound FHSD by the epoch count times the sum of single-visit energies under random reshuffling. A scalar construction attains this bound even when every update contracts perturbations (Theorem 2 and Proposition 3).
- We find that cross-epoch pairs account for about 96% and 94% of the mean squared linear prediction on reshuffled BERT-tiny and BERT-base, respectively (Figure 1). Randomizing epoch signs preserves single-visit energies while raising BERT-tiny's normalized mean absolute error (NMAE) for predicting squared retraining loss changes from 0.161 to 0.960 (Section 6.2).
- PATHGRAM achieves NMAEs of 0.071–0.106 in the primary BERT and Swin-T experiments and recovers both full-batch BERT oracle top-six sets (Table 1). Four-checkpoint refinement reaches 0.0027 for Swin-T feature adaptation and 0.0023 for BERT-tiny LoRA in independent confirmation runs (Table 3).

2 RELATED WORK

Stability along optimization. Stability connects changes in the training set to generalization by measuring how strongly those changes affect the learned predictor (Bousquet & Elisseeff, 2002; Shalev-Shwartz et al., 2010; Feldman & Vondrak, 2018). SGD analyses cover smooth and nonsmooth losses (Hardt et al., 2016; Bassily et al., 2020) and refine sensitivity through data-dependent or on-average quantities (Kuzborskij & Lampert, 2018; Lei & Ying, 2020). Geometric refinements use second-order information (Maurer, 2017) or learned-output geometry (Lei et al., 2026; Kharel et al., 2026). Multipass preconditioned SGD further reveals reuse correlations in effective-dimension bounds (Vary et al., 2026). FHSD resolves the squared replacement response at a specified finite iterate into pairs of propagated visits, retaining their signed interactions before any reduction to a scalar sensitivity measure.

Data order and trajectory influence. Data-order analyses study schedule-dependent generalization (Nikolakakis et al., 2023) and reshuffling dynamics (Lok et al., 2025). For attribution, TracIn accumulates checkpoint gradient similarities (Pruthi et al., 2020). Unrolled differentiation transports perturbations through the update sequence (Maclaurin et al., 2015; Hara et al., 2019; Schioppa et al., 2023). Temporal attribution resolves training-stage dependence through propagated gradients or checkpoint-local influence (Wang et al., 2025b; Lee et al., 2026). Stored per-visit embeddings and FHSD share every propagated summand, and they differ in whether the pairwise alignment between summands is reported as its own quantity. SOURCE reduces unrolling cost using segment statistics, approximating gradients and Hessians as stationary within each segment instead of retaining the exact

positions of individual visits (Bae et al., 2024). Distributional TDA relates trajectory derivatives to trained-model distributions (Mlodozeniec et al., 2025). We study interactions between visits of the same replacement and compare predictions with segment averaging, TracIn and last-epoch influence on Swin-T. Appendices F.10 and A.1 give the full comparison and the path information represented by these formulations.

Influence approximations. Influence functions approximate retraining using inverse curvature at an optimum (Koh & Liang, 2017). Deep-network studies examine discrepancies from complete retraining (Basu et al., 2021; Bae et al., 2022). Scalable approximations use projected inverse curvature (Schioppa et al., 2022) or projected-gradient features (Park et al., 2023). Recent methods correct deletion-induced curvature changes or accelerate inverse-curvature solves (Rubinstein & Hopkins, 2025; Wang et al., 2025a). PathGram instead approximates finite-path transport in a shared subspace. For finite dataset changes, DeltaGrad combines cached trajectories with gradient refreshes and quasi-Newton curvature (Wu et al., 2020). It provides a retraining comparison for our finite-replacement refinement.

Attribution targets and evaluation. Retraining-based influence examines memorization (Feldman & Zhang, 2020), while datamodels and sparse recovery estimate effects from training-set membership (Ilyas et al., 2022; Dagli et al., 2026). Attribution studies distinguish sensitivity to training randomness from the effects of individual and group deletions (K & Søgaard, 2021; Brophy et al., 2023). With only the final model available, further training defines a separate sensitivity target (Wei et al., 2025). We evaluate complete example replacements under matched initialization and data order, including damped IF on Swin-T. Signed loss prediction and correction selection measure magnitude agreement and candidate ordering.

Matrix-free curvature computation. Hessian–vector products access curvature along a supplied direction without forming the Hessian (Pearlmutter, 1994; Schraudolph, 2002). Matrix-function methods use such products to evaluate spectral quantities through projections and quadrature (Higham, 2008; Golub & Meurant, 2010). We use these tools to measure positive-curvature response energy separately from signed loss prediction. Appendix B.11 relates this calculation to existing curvature approximations.

3 Problem Setup

We predict how one training-data replacement changes evaluation loss. Let $S = (z_1, \ldots, z_n)$ be the training set, $w \in \mathbb{R}^p$ the trainable parameters and $\ell(w, z)$ the sample loss. Each candidate pairs an index i with a replacement z_i' from a specified rule, abbreviated by superscript (i). Original and replaced runs share initialization and batch order. For a fixed evaluation set Q, define $L_Q(w) = |Q|^{-1} \sum_{z \in Q} \ell(w, z)$. Our target is the difference in L_Q between the final replaced and original models, with negative values indicating improvement.

3.1 Why repeated visits interact

Consider a candidate visited once in each of two epochs. Its first visit changes an early update, and subsequent updates carry that perturbation forward. Its second visit adds another perturbation. Let a and b denote their contributions after propagation to the final parameters, as illustrated in Figure 2. In the linear response, the combined contribution is $a + b$, so

$$\|a + b\|^2 = \|a\|^2 + \|b\|^2 + 2a^\top b.$$

The first two terms measure individual visits, and the last measures their interaction. Aligned contributions reinforce, while opposing contributions cancel. Thus two candidates with the same individual visit energies can have very different total responses. This motivates tracking contributions through the remaining updates before combining them.

3.2 Following a replacement through training

We obtain these additive contributions by differentiating a continuous interpolation of the replacement. At interpolation weight α, use the example loss $(1 - \alpha)\ell(w, z_i) + \alpha\ell(w, z_i')$. The resulting path



Figure 2: **Two-epoch intuition.** Contributions are measured after propagation to the final parameters. Aligned unit contributions give combined energy four, and opposing ones give zero. Individual energies sum to two in both cases.

$w_t^{(i,\alpha)}$ recovers original training at $\alpha = 0$ and complete replacement at $\alpha = 1$ (Mlodozeniec et al., 2025). Hence the loss target is $\Delta_i L_Q = L_Q(w_T^{(i,1)}) - L_Q(w_T)$.

The base path takes T updates $w_{t+1} = w_t - \eta_t F_t(w_t)$ from fixed w_0. For batch $B_t \subseteq [n]$ of size b_t, its gradient is $F_t(w) = b_t^{-1} \sum_{j \in B_t} \nabla \ell(w, z_j) + \lambda_{\mathrm{wd}} w$, including weight decay $\lambda_{\mathrm{wd}} \geq 0$. Assume twice continuously differentiable losses along the interpolated paths. The batch Hessian $G_t = \nabla F_t(w_t)$ gives the update Jacobian $A_t = I - \eta_t G_t$. The current replacement changes the batch gradient by $\xi_t^{(i)}/n$, where $\xi_t^{(i)} = (n/b_t)\mathbf{1}\{i \in B_t\}[\nabla \ell(w_t, z_i') - \nabla \ell(w_t, z_i)]$. We call $\xi_t^{(i)}$ the scaled forcing.

The scaled tangent response $g_t^{(i)} = n\, \partial_\alpha w_t^{(i,\alpha)}|_{\alpha=0}$ then satisfies

$$g_{t+1}^{(i)} = A_t g_t^{(i)} - \eta_t \xi_t^{(i)}, \qquad g_0^{(i)} = 0. \tag{1}$$

Here $A_t g_t^{(i)}$ propagates the effect already accumulated, and $-\eta_t \xi_t^{(i)}$ adds the current visit's contribution. When $i \notin B_t$, the forcing is zero and propagation continues. The factor n removes the averaged objective's single-example scaling, so $g_t^{(i)}/n$ is the tangent approximation to the parameter displacement. Appendix A.2 gives the derivation.

4 Cross-Epoch Interactions in Response Energy

The two-visit example asks how much alignment changes the combined response. We now quantify this effect across candidates and bound its amplification over repeated epochs.

4.1 From Individual Responses to Visit Pairs

Fix the base data, initialization and realized schedule, denoting this information by $\mathcal{F}$. Expectations average over the specified candidate-index and replacement law. A positive-semidefinite matrix M_T weights directions of the terminal response.

Definition 1 (Finite-horizon stability dimension)**.** For a terminal metric $M_T \succeq 0$ fixed by $\mathcal{F}$, FHSD is the mean response energy $D_T(M_T) = \mathbb{E}[(g_T^{(i)})^\top M_T g_T^{(i)} \mid \mathcal{F}]$.

Choosing $M_T = I$ measures changes in the parameters. With $m_T = \nabla L_Q(w_T)$ and $M_T = m_T m_T^\top$, $D_T(M_T)/n^2$ measures mean squared linear loss predictions. This second choice connects response energy to the interaction measurements in Figure 1. A quadratic example relates this energy to a weighted direction count, motivating "dimension" (Appendix B.5).

To connect FHSD to individual visits, write $\Phi_{T,s+1} = A_{T-1} \cdots A_{s+1}$ for propagation after step s, with an empty product equal to I. Then $v_s^{(i)} = -\eta_s \Phi_{T,s+1} \xi_s^{(i)}$ is that visit's contribution at the final parameters. These $v_s^{(i)}$ generalize the two vectors a and b in Figure 2.

Theorem 2 (Visit-pair decomposition)**.** *Suppose the derivatives in Section 3 exist and* $\mathbb{E}[\left\|\xi_t^{(i)}\right\|^2 \mid \mathcal{F}] < \infty$ *for every* $t < T$*. Then*

$$g_T^{(i)} = \sum_{s<T} v_s^{(i)}, \qquad D_T(M_T) = \sum_{s,u<T} \mathbb{E}[(v_s^{(i)})^\top M_T v_u^{(i)} \mid \mathcal{F}]. \tag{2}$$

Terms with $s = u$ give the individual visit energies. Their sum, denoted $D_T^{\mathrm{diag}}(M_T)$, is the diagonal-time approximation. Terms with $s \neq u$ measure reinforcement or cancellation between visits. The expectation retains both mean contributions and variation across candidates. Appendix B.2 gives the equivalent matrix form, the time-ordered replacement Gramian.

4.2 How much can repeated epochs amplify energy?

Under random reshuffling, each epoch partitions the data into disjoint batches. An example therefore appears once per epoch, so all interactions between distinct visits occur across epochs.

Proposition 3 (Cross-epoch energy bound)**.** *If each index occurs in at most K_{vis} batches, then $0 \leq D_T(M_T) \leq K_{\mathrm{vis}} D_T^{\mathrm{diag}}(M_T)$ for every $M_T \succeq 0$. Under random reshuffling with E epochs, $K_{\mathrm{vis}} = E$. For every $E \geq 1$ and $\rho \in (0, 1)$, a scalar system whose updates contract by ρ attains $D_T = E\, D_T^{\mathrm{diag}}$.*

Identical $M_T^{1/2}$-weighted contributions across the E visits attain the upper bound, matching both direction and magnitude. Opposing contributions can cancel, as in Figure 2. Thus repeated use can amplify response energy even when individual updates contract perturbations (proof in Appendix B).

This result motivates preserving every visit's contribution. PathGram makes their propagation affordable across many candidates by sharing curvature.

5 The PathGram Estimator

PathGram turns the visit-pair analysis into individual loss predictions in three steps. It estimates each candidate's parameter response using shared curvature, converts that response to a signed loss change, and refines finite-replacement predictions with four additional gradients.

5.1 Sharing the propagation across candidates

The costly operation in Eq. (1) is applying G_t to a different response for every candidate at every update. We replace these repeated Hessian–vector products (HVPs) with one shared low-rank curvature representation. For the full training objective $F(w) = n^{-1} \sum_j \ell(w, z_j) + \lambda_{\mathrm{wd}} \|w\|^2 / 2$, use $\nabla^2 F(w_T)$ as the reference. Using HVPs (Pearlmutter, 1994), compute orthonormal eigenvectors $U \in \mathbb{R}^{p \times q}$ for its q largest algebraic eigenvalues in diagonal Λ. The curvature rank q is separate from the adapter rank. Set $P = UU^\top$ and $\bar{G} = U \Lambda U^\top$.

The approximate scaled response follows $\bar{g}_{t+1}^{(i)} = (I - \eta_t \bar{G}) \bar{g}_t^{(i)} - \eta_t \xi_t^{(i)}$ from $\bar{g}_0^{(i)} = 0$. It uses the same visit-specific forcing as the exact recurrence. To compute the update, split $\bar{g}_t^{(i)}$ into coordinates $a_t^{(i)} = U^\top \bar{g}_t^{(i)} \in \mathbb{R}^q$ inside the shared subspace and a vector $h_t^{(i)} = (I - P) \bar{g}_t^{(i)} \in \mathbb{R}^p$ outside it:

$$\begin{aligned} a_{t+1}^{(i)} &= (I_q - \eta_t \Lambda) a_t^{(i)} - \eta_t U^\top \xi_t^{(i)}, \\ h_{t+1}^{(i)} &= h_t^{(i)} - \eta_t (I - P) \xi_t^{(i)}. \end{aligned} \tag{3}$$

Both components start at zero and give $\bar{g}_t^{(i)} = U a_t^{(i)} + h_t^{(i)}$. The first line applies curvature in q coordinates. The second accumulates the remaining forcing with zero curvature in the approximation, retaining contributions outside the shared subspace. Increasing q changes propagation while preserving the complete forcing sequence. The resulting error is the sum of curvature mismatches acting on accumulated responses and propagated through later updates (Proposition 6, Appendix B.6).

5.2 Converting the response to a loss prediction

For each candidate, the estimated physical displacement is $d_T^{(i)} = \bar{g}_T^{(i)} / n$. A quadratic expansion at the original terminal model converts it to the target loss change:

$$\widehat{\Delta_i L_Q} = \frac{1}{n} \nabla L_Q(w_T)^\top \bar{g}_T^{(i)} + \frac{1}{2n^2} (\bar{g}_T^{(i)})^\top H_T \bar{g}_T^{(i)}, \qquad H_T = \nabla^2 L_Q(w_T). \tag{4}$$

The first term projects the response onto the evaluation gradient. The second accounts for evaluation-loss curvature along the displacement, using one HVP per candidate and retaining both curvature signs. Negative values predict a loss reduction, and smaller predictions rank candidates more favorably.

The workflow trains and caches the base path, prepares the eigenspace, then replays forcing and reads out the loss. After shared preparation, the prediction uses at most KT example-gradient differences and K evaluation HVPs for K candidates. Neighboring retraining uses KT batch-gradient updates, each evaluating all n examples in the full-batch setting. Appendix C separates gradient, projection and propagation costs.

5.3 Refining the finite-replacement path

The response above approximates the derivative at interpolation weight $\alpha = 0$. A complete replacement uses $\alpha = 1$, so the actual displacement satisfies $n(w_T^{(i,1)} - w_T) = g_T^{(i)} + r_T^{(i)}$, where $r_T^{(i)}$ is the nonlinear path remainder. This path error is separate from the terminal loss-expansion remainder (Appendix B.7). To improve the finite-replacement response, we measure how well the approximate neighboring path follows updates on the replaced training objective.

For one candidate under full-batch training, write $d_t = \bar{g}_t^{(i)}/n$ and $x_t = w_t + d_t$. Let $F^{(i)}$ denote its replaced objective. The update residual is

$$R_t = \nabla F^{(i)}(x_t) + \frac{x_{t+1} - x_t}{\eta_t}.$$

It is zero when x_t and x_{t+1} satisfy a gradient-descent update on $F^{(i)}$. Thus it measures the path discrepancy using one full gradient at x_t. Evaluating the replaced gradient on this shifted path captures both shared-curvature error and the effect of a finite displacement. Our four-checkpoint refinement (M4) samples R_t at $\{20, 40, 59, 79\}$ for the 80-update experiments. Linear interpolation with $R_0 = 0$ gives $\widetilde{R}_t$ at all updates. We propagate the residual through the shared curvature to correct the path:

$$c_{t+1} = (I - \eta_t \bar{G})c_t - \eta_t \widetilde{R}_t, \qquad c_0 = 0, \qquad d_T^{\mathrm{M4}} = d_T + c_T.$$

Sampling across training captures discrepancies created before the final updates. All four gradients are evaluated on the approximate path before the correction sweep. Each candidate supplies four new measurements while reusing the base checkpoints and shared eigenspace. We fixed this budget on development runs before confirmation. Refinement comparisons evaluate $L_Q(w_T + d) - L_Q(w_T)$ for every method, using $d = d_T^{\mathrm{M4}}$ for M4, to isolate the response correction (Appendix H).

5.4 Measuring positive-curvature response energy

For separate energy analysis, we use $(H_T)_+$, obtained by zeroing negative evaluation-Hessian eigenvalues. Averaging $(\bar{g}_T^{(i)})^\top (H_T)_+ \bar{g}_T^{(i)}$ across candidates estimates FHSD in this metric. An HVP-based calculation returns an interval for this mean, adding two HVPs per candidate at each step until the requested relative width is reached. Appendix B.8 gives the construction and interval theorem, with numerical-operator conditions in Appendix B.9. Section 6.7 evaluates width and cost.

6 Experiments

We test whether cross-epoch alignment controls terminal response energy and whether retaining this structure supports replacement prediction. We then identify the remaining error sources, evaluate four-checkpoint refinement, and measure accuracy and cost as the training and candidate sets grow.

6.1 Setup

Data and models. We compare every candidate prediction with complete neighboring retraining under matched initialization and data order. For text, we use binary classification on the baseball and space categories of 20 Newsgroups (Lang, 1995). The primary study uses 96 training documents, 24 replacements and 128 evaluation documents. Each replacement substitutes a held-out example of the other class. We adapt BERT-tiny (Turc et al., 2019) and BERT-base using rank-four LoRA on the

Table 1: **Primary replacement predictions at $n = 96$.** RR entries of K count candidates times schedules. Sign denotes sign accuracy, Prec denotes precision at the requested k, and Cost is time relative to neighboring retraining. All rows have zero normalized regret.

Setting	K	NMAE ↓	ρ ↑	Sign ↑	Prec ↑	Cost ↓
BERT-tiny FB	24	0.0762	0.9983	1.000	1.000	0.065
BERT-base FB	24	0.1060	0.9957	1.000	1.000	0.026
BERT-tiny RR	12×2	0.0972	0.9860	0.917	1.000	0.184
BERT-base RR	12×2	0.0731	0.9860	1.000	1.000	0.083
Swin-T primary	16	0.0713	1.0000	1.000	1.000	0.744

Full-batch rows use one initialization. RR predictions and outcomes are averaged over two schedules for each candidate. Selection uses $k = 6$ for full-batch text, $k = 3$ for RR and $k = 10$ for vision. Three-run enlarged-data results appear in Appendix F.3.

final attention query and value projections plus a trainable head (Devlin et al., 2019; Hu et al., 2022). Full-batch runs share gradient-flow time 0.8. Random reshuffling (RR) uses batches of 48 and two schedules per candidate. The enlarged text study uses $n = 1{,}000$ and four BERT sizes, with three independent data and initialization runs per model.

For vision, we classify CIFAR-10 airplanes and automobiles (Krizhevsky, 2009) using a rank-two tanh adapter and head on fixed Swin-T features (Liu et al., 2021). We use 96 training images and 256 evaluation images, with 16 primary candidates and larger candidate sets drawn from a pool of 128 images. Ten independent softplus Fashion-MNIST models compare temporal aggregation within a fixed terminal eigenspace (Xiao et al., 2017). Six directions from three GroupNorm ResNet-20 models evaluate positive-curvature computation (He et al., 2016; Wu & He, 2018). Appendix E gives the configurations.

Comparisons. External methods compare alternative trajectory summaries. Component ablations remove the orthogonal response, curvature propagation, or temporal variation in forcing (Appendix E.6). The exact tangent uses the true Hessian at each base-path step. For RR, eight independent epoch-sign draws estimate diagonal-time linear energy. A separate sign-randomization comparison applies the quadratic readout to the randomized responses and measures squared-loss prediction error. For refinement, all methods use actual terminal loss evaluation on new runs with 16 replacements each. Swin-T crosses three seeds with three step sizes, and BERT-tiny uses three seeds at one step size (Appendix H.2).

Metrics. Magnitude error is $\mathrm{NMAE} = \sum_i |\widehat{\Delta}_i - \Delta_i| / \sum_i |\Delta_i|$. Spearman correlation measures ordering (Spearman, 1904), and sign accuracy measures agreement in the direction of the loss change. Moment NMAE uses schedule-averaged squared quantities. We select the k smallest predictions and report precision@k as the fraction in the retraining oracle set A^*. We use $k = 6$ for full-batch text, $k = 3$ for RR and $k = 10$ for primary vision selection. Normalized regret is $\mathrm{Reg}(A) = (|A|^{-1} \sum_{i \in A} \Delta_i - |A^*|^{-1} \sum_{i \in A^*} \Delta_i)/r$, with $r = \max_i \Delta_i - \min_i \Delta_i$. Cost is estimator time divided by total neighboring retraining time on the same candidates and hardware, excluding common base training and state storage.

6.2 Cross-epoch alignment shapes terminal response energy

We first test whether propagated visits reinforce at the terminal parameters, as identified by the visit-pair decomposition in Section 4. Cross-epoch pairs contribute over 90% of the measured linear-response energy on both reshuffled BERT models (Figure 1a). To isolate alignment from single-visit magnitude, we randomize epoch signs while preserving diagonal energy. This intervention increases squared quadratic NMAE about sixfold on BERT-tiny, with the same direction on BERT-base (Figure 1b). Alignment therefore contributes both to response magnitude and to agreement with squared retraining changes. Table 19 separates these two measurements, and Appendix F.9 gives the corresponding full-batch Swin-T energies.

We next isolate temporal aggregation on ten Fashion-MNIST models, holding each model's terminal eigenspace and propagation fixed. The full moment is closer to measured energy on all ten models than a peak envelope of the same forcing sequence (Table 19). Its median multiplicative error is 1.12, compared with 1.33 for that envelope. Taking separate peaks of direct forcing and curvature

Table 2: **External estimators on the primary Swin-T candidates.** All methods use the same 16 replacements and select six. Linear and quadratic columns report NMAE (↓) using the same response. Ordering and selection use the quadratic readout when available. TracIn uses checkpoint similarities (Score). All methods have zero normalized regret except TracIn (0.0043).

Method	Linear/Score	Quadratic	$\rho \uparrow$	Sign ↑	Prec@6 ↑
PATHGRAM	0.2773	0.0713	1.0000	1.000	1.000
SOURCE, 4 segments	0.2772	0.0784	1.0000	1.000	1.000
Damped IF, factor 1	0.3936	0.3366	0.9912	0.938	1.000
TracIn	0.4023	—	0.9941	0.875	0.833
Last-epoch IF	0.9766	0.9765	0.9824	0.875	1.000

residual further enlarges the error. Preserving the forcing sequence thus improves energy-magnitude prediction, while the same-path peak retains slightly higher rank correlation.

6.3 SHARED PROPAGATION PREDICTS REPLACEMENT EFFECTS

We next test whether the retained responses predict signed effects of complete replacements. Across the primary experiments, PATHGRAM achieves NMAE 0.071–0.106 and recovers every requested oracle set (Table 1). Figure 6 shows candidate-level magnitude agreement across beneficial and harmful replacements.

Propagation and terminal readout. Table 2 compares external estimators and readouts on the same Swin-T candidates. The quadratic term reduces NMAE by more than threefold for both PATHGRAM and four-segment SOURCE, while preserving their oracle selections. This common gain identifies terminal curvature as an important part of magnitude prediction for both trajectory summaries. The segment-count and stopping-time sweeps in Figure 11b–c separate two further questions: finer SOURCE segmentation approaches its exponential-propagation result without improving replacement accuracy, and PATHGRAM remains accurate across the measured training horizons.

Information supporting selection. Removing the orthogonal response in Eq. (3) lowers BERT-base precision@6 from one to 0.500. Removing curvature propagation or temporal variation in forcing also changes the selected sets (Table 12). The comparison connects selection quality to two retained ingredients: propagation of the full forcing sequence and accumulation of its orthogonal component.

6.4 ERROR DECOMPOSITION MOTIVATES FOUR-CHECKPOINT REFINEMENT

Accurate ordering leaves a further question: which response error should be reduced to improve predicted magnitudes? We separate propagation, terminal expansion and finite replacement by evaluating the approximate response, exact tangent and retrained displacement. The BERT measured-displacement readout errors are below 0.0007 (Table 18), directing attention to the response approximation. On a separate Swin-T study, the finite-replacement discrepancy dominates the terminal expansion remainder across all nine seed–step-size combinations (Figure 3). The signed decomposition in Eq. (20) also reveals partial compensation between propagation and tangent errors. Improving tangent propagation alone therefore need not improve the complete replacement prediction. This motivates measuring the update residual along the approximate neighboring path.

On new confirmation seeds, the resulting M4 correction lowers mean NMAE below 0.003 on both tasks (Table 3). Spending the same four-gradient budget on tail retraining yields much smaller gains. Distributing residual measurements along the path thus improves magnitude prediction more effectively than replaying its last four updates. M4 also preserves oracle choices and eliminates the observed sign errors (Table 29). A separate component study finds that the complete residual gives lower loss and parameter errors than either original-objective response or changed-example feedback alone on all nine development configurations (Table 28). Both contributions are therefore retained in the correction.

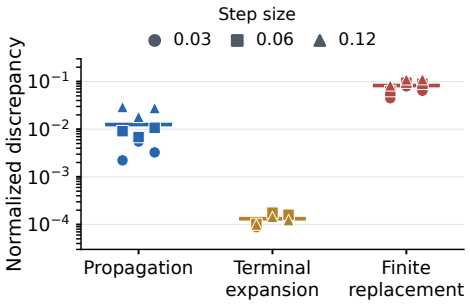


Figure 3: **Sources of loss error.** Normalized absolute discrepancies for PATHGRAM on Swin-T seeds 313–315, each at three step sizes. Markers show configurations and bars show means.

Table 3: **Four-checkpoint refinement.** Mean NMAE on new seeds with actual terminal loss readout for every method. Swin-T: three seeds, three step sizes each. BERT-tiny: three seeds, one step size. Replay uses four gradients from the shifted checkpoint.

Method	Swin-T	BERT-tiny
PATHGRAM	0.1034	0.0741
+ four-step replay	0.0945	0.0700
+ M4 correction	**0.0027**	**0.0023**

6.5 PREDICTION ACROSS TRAINING SIZES AND CANDIDATE SETS

We enlarge the text training set to $n = 1{,}000$ to test prediction with more data. Table 11 displays all three runs for each of four BERT sizes: eight recover the oracle top-six set, and the other four recover five members. The maximum normalized regret is 0.0136. The run-level comparison shows that candidate selection remains accurate across variation in magnitude error.

Enlarging the vision candidate set preserves oracle selection and similar NMAE while reusing the shared eigenspace (Table 10). A separate BERT panel varies backbone size at fixed adaptation settings and gradient-flow time. Its nonmonotone errors show that parameter count alone does not determine prediction quality (Figure 8).

6.6 COMPUTATION COST AND SPECTRAL RANK

Figure 9 relates prediction error to the cost of evaluating each candidate pool. Shared path and eigenspace computation costs less than neighboring retraining throughout these settings, with BERT ratios spanning 0.012–0.184 across settings (Tables 1 and 10). The rank sweep in Figure 11a explains why a compact eigenspace suffices on Swin-T: beyond five directions, NMAE changes little and the oracle selection is unchanged (Table 16).

For refinement, repeated Swin-T timings give a median paired speedup of $3.12\times$ over full retraining. A separate interleaved comparison gives a $5.44\times$ speedup over DeltaGrad, with both methods below 0.003 NMAE and both recovering every optimal action (Table 30).

6.7 EVALUATING POSITIVE-CURVATURE ENERGY

Finally, we evaluate FHSD's positive-curvature metric (Section 5.4). All six ResNet intervals reach relative width below 10% by 192 HVPs, with their contraction shown in Figure 10a. The leading 20 positive modes capture a median 16.4% of the 256-HVP midpoint energy (Figure 10b). Most energy lies beyond these modes, supporting complete-metric computation with a width-controlled budget. Table 15 gives the projected numerical intervals for BERT responses.

7 CONCLUSION

We identify cross-epoch alignment as a key component of replacement sensitivity. FHSD makes this structure explicit through an exact visit-pair decomposition and a sharp epoch-count bound under random reshuffling. Guided by this analysis, PATHGRAM preserves complete forcing sequences through shared curvature propagation. Experiments reveal strong reinforcement across epochs in reshuffled BERT models and accurate replacement selection on BERT and Swin-T. Four additional gradient measurements refine the approximate neighboring path and substantially improve magnitude prediction. Combining temporal interaction analysis with path residual correction thus provides a practical route from trajectory sensitivity to predicting the effects of finite data replacements.

AI USE STATEMENT

In this work, we used generative AI tools for tasks with required disclosure. Generative AI tools helped develop the theoretical models and conceptual frameworks, helped formulate the mathematical claims, provided critical ingredients for proving those claims, and assisted in the writing of the proofs. The mathematical claims, the proofs, and every experimental result were checked and verified by the authors. We additionally used generative AI tools for tasks with recommended disclosure, namely to aid and polish the writing and to edit the paper for readability. We have reviewed all AI-assisted work, including the mathematical derivations and the text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger Grosse. Training data attribution via approximate unrolling. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4381–4391, 2020.

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Jonathan Brophy, Zayd Hammoudeh, and Daniel Lowd. Adapting and evaluating influence-estimation methods for gradient-boosted decision trees. *Journal of Machine Learning Research*, 24(154): 1–48, 2023. URL `http://jmlr.org/papers/v24/22-0449.html`.

Tyler Chen, Thomas Trogdon, and Shashanka Ubaru. Randomized matrix-free quadrature: Unified and uniform bounds for stochastic lanczos quadrature and the kernel polynomial method. *SIAM Journal on Scientific Computing*, 47(3):A1733–A1757, 2025. doi: 10.1137/23M1600414.

Rishit Dagli, Abir Harrasse, Luke Zhang, Florent Draye, Amirali Abdullah, Bernhard Schölkopf, and Zhijing Jin. Stride: Training data attribution via sparse recovery from subset perturbations. *arXiv preprint arXiv:2606.05165*, 2026. URL `https://arxiv.org/abs/2606.05165`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

Alok Dutt, Leslie Greengard, and Vladimir Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40:241–266, 2000. doi: 10.1023/A:1022338906936.

Vitaly Feldman and Jan Vondrak. Generalization bounds for uniformly stable algorithms. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9747–9757, 2018.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2881–2891, 2020.

Andreas Frommer, Kathryn Lund, Marcel Schweitzer, and Daniel B. Szyld. The radau–lanczos method for matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 38(3):710–732, 2017. doi: 10.1137/16M1072565.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a Kronecker factored eigenbasis. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9550–9560, 2018.

Gene H. Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2): 205–224, 1965.

Gene H. Golub and Gérard Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton University Press, 2010.

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with SGD. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, 2008.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587, 2022.

Karthikeyan K and Anders Søgaard. Revisiting methods for finding influential examples. *arXiv preprint arXiv:2111.04683*, 2021. URL https://arxiv.org/abs/2111.04683.

Ayub Kharel, Ilja Kuzborskij, Patrick Rebeschini, and Yasin Abbasi-Yadkori. Generalization in nonlinear least squares via learned feature geometry. *arXiv preprint arXiv:2606.08799*, 2026.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1885–1894, 2017.

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

Ilja Kuzborskij and Christoph H. Lampert. Data-dependent stability of stochastic gradient descent. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2815–2824, 2018.

Ken Lang. NewsWeeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pp. 331–339, 1995.

Jin Hwa Lee, Matthew Smith, Maxwell Adam, and Jesse Hoogland. Influence dynamics and stagewise data attribution. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3c1edb917edb6cec59bf3a0afc03150e-Abstract-Conference.html.

Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5809–5819, 2020.

Yunwen Lei, Zimeng Wang, and Xiaoming Yuan. From convergence to generalization: Stability of stationary-point learning algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(9):10598–10609, 2026. doi: 10.1109/TPAMI.2026.3688491.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021. doi: 10.1109/ICCV48922.2021.00986.

Jackie Lok, Rishi Sonthalia, and Elizaveta Rebrova. Error dynamics of mini-batch gradient descent with random reshuffling for least squares regression. In *Proceedings of the 36th International Conference on Algorithmic Learning Theory*, volume 272 of *Proceedings of Machine Learning Research*, pp. 736–770, 2025.

Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2113–2122, 2015.

James Martens. Deep learning via Hessian-free optimization. In *International Conference on Machine Learning*, pp. 735–742, 2010.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417, 2015.

Andreas Maurer. A second-order look at stability and generalization. In *Proceedings of the 30th Conference on Learning Theory*, pp. 1461–1475, 2017.

Bruno Mlodozeniec, Isaac Reid, Sam Power, David Krueger, Murat Erdogdu, Richard E. Turner, and Roger Grosse. Distributional training data attribution: What do influence functions sample? In *Advances in Neural Information Processing Systems*, volume 38, pp. 11110–11144, 2025. doi: 10.52202/085713-0339.

Konstantinos E. Nikolakakis, Amin Karbasi, and Dionysis Kalogerias. Select without fear: Almost all mini-batch schedules generalize optimally. *arXiv preprint arXiv:2305.02247*, 2023.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113, 2023.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020.

Ittai Rubinstein and Samuel Hopkins. Rescaled influence functions: Accurate data attribution in high dimension. In *Advances in Neural Information Processing Systems 38*, pp. 167563–167592, 2025. doi: 10.52202/085713-5047.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8179–8186, 2022.

Andrea Schioppa, Katja Filippova, Ivan Titov, and Polina Zablotskaia. Theoretical and practical perspectives on what influence functions do. In *Advances in Neural Information Processing Systems 36*, pp. 27560–27581, 2023. doi: 10.52202/075280-1198.

Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*, 2019.

Simon Vary, Tyler Farghly, Ilja Kuzborskij, and Patrick Rebeschini. On-average stability of multipass preconditioned SGD and effective dimension. In *Proceedings of the 39th Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 6458–6495, 2026.

Andrew Wang, Elisa Nguyen, Runshi Yang, Juhan Bae, Sheila McIlraith, and Roger Grosse. Better training data attribution via better inverse hessian-vector products. In *Advances in Neural Information Processing Systems 38*, pp. 112895–112930, 2025a. doi: 10.52202/085713-3408.

Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *International Conference on Learning Representations*, 2025b.

Dennis Wei, Inkit Padhi, Soumya Ghosh, Amit Dhurandhar, Karthikeyan Natesan Ramamurthy, and Maria Chang. Final-model-only data attribution with a unifying view of gradient-based methods. In *Advances in Neural Information Processing Systems 38*, pp. 118221–118255, 2025. doi: 10.52202/085713-3562.

Yinjun Wu, Edgar Dobriban, and Susan Davidson. DeltaGrad: Rapid retraining of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10355–10366, 2020. URL `https://proceedings.mlr.press/v119/wu20b.html`.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pp. 3–19, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A Notation and Comparison with Related Formulations

This appendix supports the definitions in Sections 3–4 and the formulation comparison in Section 2.

Table 4: Core notation. Training curvature G_t and evaluation curvature H_T are distinct objects.

Symbol	Meaning
B_t, b_t, η_t	realized batch, batch size, step size
G_t, A_t	batch Hessian and update Jacobian $I - \eta_t G_t$
$\xi_t^{(i)}$	scaled forcing caused by candidate i
$\Phi_{t,s}$	ordered propagation from step s to t
$g_T^{(i)}$	scaled tangent terminal response
C_{su}	raw cross-time forcing moment
W_T	raw second moment of terminal responses
H_T, M_T	evaluation Hessian and a positive-semidefinite terminal metric
$D_T(M_T)$	finite-horizon stability dimension

A.1 Information retained by related formulations

Table 5 states which information about a one-sample perturbation each formulation retains. "Ordered path" means that an early hit is transported through the subsequent realized updates. "Cross-time" means that correlations between different hits remain in the raw second moment.

A.2 Replacement interpolation and response interpretation

This subsection explains the replacement construction and conditional moments in Sections 3–4.

Table 5: Information retained about a one-sample perturbation. Rows refer to stepwise stability (Hardt et al., 2016; Kuzborskij & Lampert, 2018) and stationary-point geometry (Lei et al., 2026; Kharel et al., 2026). They also cover multipass effective dimension (Vary et al., 2026), endpoint influence (Koh & Liang, 2017) and checkpoint gradient tracing (Pruthi et al., 2020). Entries concern the stated mathematical target of each cited formulation. In particular, stationary-point geometry describes learned outputs without an ordered finite-step replacement recurrence, so its finite-horizon and ordered-path entries are marked no. A no entry denotes information absent from that formulation rather than an impossibility for extensions of the method.

Approach	Finite T	Ordered path	Cross-time	Singular M_T	Cand. loss
Stepwise stability	yes	bound	no	yes	no
Stationary geometry	no	no	no	limited	no
Multipass eff. dim.	yes	contraction	implicit	pos. definite	no
Endpoint influence	endpoint	no	no	reg. inverse	yes
Checkpoint tracing	yes	no	sum	yes	proxy
FHSD and PATHGRAM	yes	yes	yes	yes	yes

Differentiating the realized update. Using the notation of Section 3, the base path follows

$$w_{t+1} = w_t - \eta_t F_t(w_t), \qquad F_t(w) = \frac{1}{b_t} \sum_{j \in B_t} \nabla \ell(w, z_j) + \lambda_{\mathrm{wd}} w. \tag{5}$$

Setting $\lambda_{\mathrm{wd}} = 0$ removes weight decay. The replacement interpolation is

$$F_t^{(i,\alpha)}(w) = F_t(w) + \frac{\alpha}{b_t} \mathbf{1}\{i \in B_t\} \big[\nabla \ell(w, z_i') - \nabla \ell(w, z_i)\big].$$

The base path is recovered at $\alpha = 0$. Differentiating the interpolated update and multiplying by n gives

$$g_{t+1}^{(i)} = (I - \eta_t G_t) g_t^{(i)} - \eta_t \xi_t^{(i)}, \qquad g_0^{(i)} = 0.$$

The derivative of the interpolated forcing through $w_t^{(i,\alpha)}$ is multiplied by α and therefore vanishes at the base path. This yields Eq. (1) with both its transport term and its current-visit forcing.

Conditional moments and terminal geometry. The dimension p counts trainable parameters. Conditioning on the realized schedule covers full-batch training and deterministic curricula as well as a realized random-reshuffling run, because it fixes which examples contribute to each batch gradient and therefore which update Jacobians transport a candidate's gradient changes from the steps at which that example is visited to the terminal parameters. The evaluation gradient $\nabla L_Q(w_T)$ and Hessian H_T provide the terminal loss readout. The positive-curvature metric weights directions with positive quadratic contributions to evaluation loss. Under fixed positive-semidefinite curvature and matched forcing second moments, FHSD is comparable to a weighted count of directions resolved within the horizon because the response factor in each eigendirection enters its squared energy through the finite-horizon filter derived in Appendix B.5, which discounts unresolved directions quadratically. The replace-one argument in Appendix B.1 relates expected replacement loss changes to the generalization gap under independent identically distributed sampling and a symmetric learning algorithm.

B PROOFS

This appendix proves the finite-horizon identities and curvature bounds in Sections 3–5.4.

B.1 REPLACE-ONE GENERALIZATION IDENTITY

This subsection derives the generalization identity associated with the replacement target in Section 3.

Replacement loss changes also connect to generalization through the standard replace-one construction (Bousquet & Elisseeff, 2002). For this connection, let $S = (Z_1, \ldots, Z_n)$ and $S' = (Z_1', \ldots, Z_n')$ be independent samples of independent identically distributed observations. Write $S^{i \leftarrow i'}$ for S with Z_i replaced by Z_i'. Let $\mathcal{A}$ be a symmetric learning algorithm whose output distribution is invariant to

permuting the training examples. The population and empirical risks are $R(w) = \mathbb{E}_Z[\ell(w, Z)]$ and $R_S(w) = n^{-1}\sum_i \ell(w, Z_i)$, respectively. Here Z is an independent observation from the same population. Define $\Delta_i \ell_{Z_i'} = \ell(\mathcal{A}(S^{i\leftarrow i'}), Z_i') - \ell(\mathcal{A}(S), Z_i')$ as the replacement loss change evaluated on the replacement example. The expected generalization gap then satisfies

$$\mathbb{E}\big[R(\mathcal{A}(S)) - R_S(\mathcal{A}(S))\big] = -\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[\Delta_i \ell_{Z_i'}\big], \tag{6}$$

where the expectation averages over the samples and algorithm randomness. The identity expresses the expected generalization gap as an average replacement loss change with the opposite sign. This section derives this identity by exchanging the original and replacement examples. Our fixed-Q experiments instead evaluate predictions of $\Delta_i L_Q$ for a realized training run.

Let $S = (Z_1, \ldots, Z_n)$ and $S' = (Z_1', \ldots, Z_n')$ be independent samples and let $S^{i\leftarrow i'}$ replace Z_i by Z_i'. For a symmetric learning algorithm $\mathcal{A}$ put $\Delta_i \ell_{Z_i'} = \ell(\mathcal{A}(S^{i\leftarrow i'}), Z_i') - \ell(\mathcal{A}(S), Z_i')$. The pair $(S^{i\leftarrow i'}, Z_i')$ has the same joint distribution as a training sample together with its ith observation, whereas Z_i' is independent of $\mathcal{A}(S)$. Therefore $\mathbb{E}\ell(\mathcal{A}(S^{i\leftarrow i'}), Z_i') = \mathbb{E}\ell(\mathcal{A}(S), Z_i)$ and $\mathbb{E}\ell(\mathcal{A}(S), Z_i') = \mathbb{E}R(\mathcal{A}(S))$. Averaging the first of these equalities over i and subtracting the second gives (6).

B.2 Matrix form and proof of Theorem 2

We prove the visit-pair identity through its matrix form. For arbitrary steps $t \geq s$, write $\Phi_{t,s} = A_{t-1}\cdots A_s$ with $\Phi_{s,s} = I$. Define the raw moments $W_T = \mathbb{E}[g_T^{(i)}(g_T^{(i)})^\top \mid \mathcal{F}]$ and $C_{su} = \mathbb{E}[\xi_s^{(i)}(\xi_u^{(i)})^\top \mid \mathcal{F}]$. These include both mean contributions and centered covariance. Then $D_T(M_T) = \mathrm{Tr}(M_T W_T)$, where Tr denotes trace. The diagonal-time matrix is $W_T^{\mathrm{diag}} = \sum_{s<T}\eta_s^2 \Phi_{T,s+1} C_{ss} \Phi_{T,s+1}^\top$, giving $D_T^{\mathrm{diag}}(M_T) = \mathrm{Tr}(M_T W_T^{\mathrm{diag}})$. The time-ordered replacement Gramian has the following equivalent form.

> Suppose the derivatives in Section 3 exist and $\mathbb{E}[\left\|\xi_t^{(i)}\right\|^2 \mid \mathcal{F}] < \infty$ for all $t < T$. The terminal response and its conditional second moment satisfy
>
> $$g_T^{(i)} = -\sum_{s=0}^{T-1}\eta_s \Phi_{T,s+1}\xi_s^{(i)},$$
> $$W_T = \sum_{s,u<T}\eta_s\eta_u \Phi_{T,s+1} C_{su} \Phi_{T,u+1}^\top,$$
>
> where both indices in the double sum range over the training steps from 0 to $T-1$. Consequently $D_T(M_T) = \sum_{s,u<T}\eta_s\eta_u \mathrm{Tr}(M_T \Phi_{T,s+1} C_{su} \Phi_{T,u+1}^\top)$ for every $M_T \succeq 0$.

The tangent recursion is $g_{t+1} = A_t g_t - \eta_t \xi_t$ with $g_0 = 0$. Repeated substitution gives $g_T = -\sum_{s=0}^{T-1}\eta_s \Phi_{T,s+1}\xi_s$. Taking the outer product yields

$$g_T g_T^\top = \sum_{s,u<T}\eta_s\eta_u \Phi_{T,s+1}\xi_s\xi_u^\top \Phi_{T,u+1}^\top, \tag{7}$$

and conditional expectation gives $W_T = \sum_{s,u<T}\eta_s\eta_u \Phi_{T,s+1} C_{su} \Phi_{T,u+1}^\top$. Finally $\mathbb{E}[g_T^\top M_T g_T] = \mathrm{Tr}(M_T W_T)$ for every $M_T \succeq 0$. No two transition matrices were exchanged in this derivation.

B.3 Batch intersections under reshuffling

This result formalizes why distinct visits within one epoch have zero cross moment.

Corollary 4 (Batch intersection)**.** *Let i be sampled uniformly from $[n]$ after fixing a realized schedule. Write $\zeta_t^{(i)} = \nabla\ell(w_t, z_i') - \nabla\ell(w_t, z_i)$ for the unscaled candidate gradient difference. Then $C_{su} = \frac{n}{b_s b_u}\sum_{i\in B_s\cap B_u}\mathbb{E}[\zeta_s^{(i)}(\zeta_u^{(i)})^\top \mid \mathcal{F}, i]$. Distinct batches in the same reshuffling epoch have $C_{su} = 0$, while cross-epoch intersections can retain nonzero moments.*

For a uniformly sampled index i, write $\xi_t^{(i)} = (n/b_t)\mathbf{1}\{i \in B_t\}\zeta_t^{(i)}$. Then

$$C_{su} = \frac{n^2}{b_s b_u}\mathbb{E}_i\Big[\mathbf{1}\{i \in B_s \cap B_u\}\zeta_s^{(i)}(\zeta_u^{(i)})^\top\Big] = \frac{n}{b_s b_u}\sum_{i \in B_s \cap B_u}\mathbb{E}\Big[\zeta_s^{(i)}(\zeta_u^{(i)})^\top \mid i\Big]. \tag{8}$$

Two distinct batches in one reshuffling epoch are disjoint, so their cross moment vanishes. The same index appears once in every later epoch, which permits nonzero cross-epoch terms.

B.4 Proof of Proposition 3

This subsection proves the cross-epoch amplification bound in Section 4. We restate the result before its proof.

> If every candidate index lies in at most K_{vis} of the batches $B_0, \dots, B_{T-1}$, then $0 \le D_T(M_T) \le K_{\mathrm{vis}} D_T^{\mathrm{diag}}(M_T)$ for every $M_T \succeq 0$. Under random reshuffling with E epochs, $K_{\mathrm{vis}} = E$. For every $E \ge 1$ and contraction factor $\rho \in (0,1)$, a strictly contractive scalar system with one visit per epoch attains $D_T = E\, D_T^{\mathrm{diag}}$.

Fix a candidate i that lies in the batches of times $s_1 < \cdots < s_{K_i}$, $K_i \le K_{\mathrm{vis}}$, and write $v_k = -\eta_{s_k}\Phi_{T,s_k+1}\xi_{s_k}^{(i)}$ and $\|x\|_M^2 = x^\top M x$ for $M = M_T \succeq 0$. Since $\xi_s^{(i)} = 0$ whenever $i \notin B_s$, Theorem 2 gives $g_T^{(i)} = \sum_k v_k$. Expanding the squares,

$$K_i \sum_k \|v_k\|_M^2 - \Big\|\sum_k v_k\Big\|_M^2 = \sum_{k<l}\|v_k - v_l\|_M^2 \ge 0,$$

so $0 \le (g_T^{(i)})^\top M g_T^{(i)} \le K_{\mathrm{vis}}\sum_{s<T}\eta_s^2(\xi_s^{(i)})^\top\Phi_{T,s+1}^\top M\Phi_{T,s+1}\xi_s^{(i)}$. The conditional expectation of the right side is $K_{\mathrm{vis}}\,\mathrm{Tr}(MW_T^{\mathrm{diag}})$, which proves $0 \le D_T(M) \le K_{\mathrm{vis}}\, D_T^{\mathrm{diag}}(M)$. Under random reshuffling each epoch partitions $[n]$ into batches, so every index lies in one batch per epoch and $K_{\mathrm{vis}} = E$. With every batch equal to $[n]$, $K_{\mathrm{vis}} = T$.

For equality, take $M = 1$ and fix $E \ge 1$, $\rho \in (0,1)$ and a nonzero scalar v. Use E scalar update Jacobians $A_s = \rho$, and choose step sizes $\eta_s > 0$ and forcing values $\xi_s = -v/(\eta_s\rho^{E-1-s})$ for $s = 0, \dots, E-1$. Every hit then has the same propagated contribution $-\eta_s\Phi_{E,s+1}\xi_s = v$. The full terminal response is Ev, so its energy is E^2v^2, while the diagonal-time energy retains the E individual energies and equals Ev^2. Their ratio is E, and since $|A_s| = \rho < 1$ every update is strictly contractive.

B.5 Finite-horizon spectral filter

This subsection develops the effective-dimension interpretation of FHSD from Section 4 and relates finite-horizon propagation to inverse curvature.

The word dimension has a concrete interpretation in a fixed-curvature example. Let $G_t = G \succeq 0$ and $\eta_t = \eta > 0$, with constant forcing $\xi_t^{(i)} = \xi$ over time. Assume $\eta G \preceq I$ and $\mathbb{E}[\xi\xi^\top \mid \mathcal{F}] = G$, and choose $M_T = G$. In this case the direction count bounds FHSD as

$$(1 - e^{-1})^2 N_T \le D_T(G) \le N_T, \qquad N_T = \sum_j \min\{1, (\eta T\lambda_j)^2\},$$

where λ_j are the eigenvalues of G and e is the base of the natural logarithm. Each direction contributes one to N_T once $\eta T\lambda_j \ge 1$ and contributes $(\eta T\lambda_j)^2$ below that threshold. Thus FHSD is comparable to a weighted count of directions resolved within the training horizon. This section derives the constants from Proposition 5.

Proposition 5 (Finite-horizon spectral filter)**.** *Let $G \succeq 0$, $A = I - \eta G$, and $\xi_t = \xi$ for T steps. For an eigenvalue λ with $0 \le \eta\lambda \le 1$, the response filter is $r_T(\lambda) = (1 - (1-\eta\lambda)^T)/\lambda$ for $\lambda > 0$ and $r_T(0) = \eta T$, and it obeys*

$$(1 - e^{-1})\min\{\eta T, \tfrac{1}{\lambda}\} \le r_T(\lambda) \le \min\{\eta T, \tfrac{1}{\lambda}\}, \tag{9}$$

where $1/0$ is interpreted as $+\infty$.

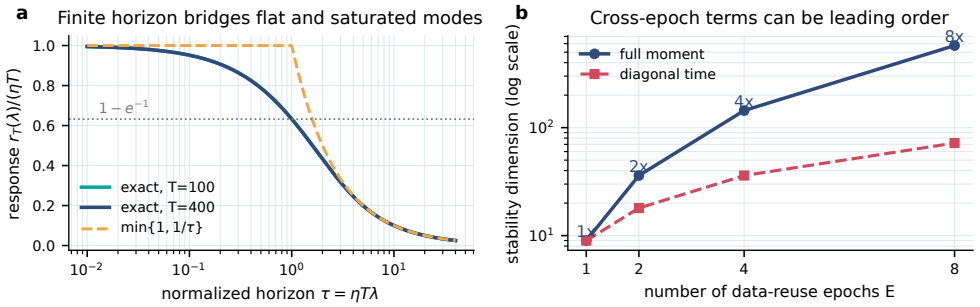


Figure 4: Two finite-horizon effects. (a) In a fixed positive-semidefinite mode, the response interpolates between the short-horizon branch ηT and the saturated branch $1/\lambda$, and it remains finite at $\lambda = 0$. (b) Repeated aligned hits produce E^2 full energy and only E diagonal-time energy.

Let $x = \eta\lambda \in [0, 1]$. For $\lambda > 0$, $r_T(\lambda) = \eta \sum_{j=0}^{T-1}(1-x)^j = (1-(1-x)^T)/\lambda$. The geometric sum gives $r_T \le \eta T$, and since the numerator is at most one, $r_T \le 1/\lambda$. For the lower bound use $(1-x)^T \le e^{-Tx}$. If $Tx \le 1$, concavity of $1 - e^{-y}$ on $[0, 1]$ gives $1 - e^{-Tx} \ge (1 - e^{-1})Tx$. If $Tx \ge 1$ then $1 - e^{-Tx} \ge 1 - e^{-1}$. Division by λ proves $r_T(\lambda) \ge (1 - e^{-1}) \min\{\eta T, 1/\lambda\}$. At $\lambda = 0$ the geometric sum is ηT.

Counting interpretation. Suppose in addition that $\mathbb{E}[\xi\xi^\top \mid \mathcal{F}] = G$, that the eigenvalues λ_j of G lie in $[0, 1/\eta]$ and that $M_T = G$. Then $g_T = -r_T(G)\xi$, $W_T = r_T(G)Gr_T(G)$ and $D_T(G) = \sum_j \lambda_j^2 r_T(\lambda_j)^2$. Multiplying (9) by λ_j gives $(1-e^{-1}) \min\{1, \eta T\lambda_j\} \le \lambda_j r_T(\lambda_j) \le \min\{1, \eta T\lambda_j\}$, so $D_T(G)$ lies between $(1 - e^{-1})^2$ and 1 times $\sum_j \min\{1, (\eta T\lambda_j)^2\}$. Each term of this sum equals one when $\lambda_j \ge 1/(\eta T)$ and equals $(\eta T\lambda_j)^2$ below this threshold. The effective dimension $\sum_j \lambda_j/(\lambda_j + \gamma)$ at $\gamma = 1/(\eta T)$ has the same threshold, although it discounts small eigenvalues linearly rather than quadratically.

B.6 Error from shared curvature

The following identity quantifies the curvature approximation in Section 5.

Proposition 6 (Reference propagation error)**.** *The exact and approximate scaled tangent responses satisfy* $g_T^{(i)} - \bar{g}_T^{(i)} = -\sum_{s<T} \eta_s \Phi_{T,s+1}(G_s - \bar{G})\bar{g}_s^{(i)}$.

Let $\bar{A}_t = I - \eta_t \bar{G}$ and $e_t = g_t - \bar{g}_t$. Subtracting the true and reference recursions gives $e_{t+1} = A_t g_t - \bar{A}_t \bar{g}_t = A_t e_t - \eta_t(G_t - \bar{G})\bar{g}_t$. Since $e_0 = 0$, unrolling produces $e_T = -\sum_{s<T} \eta_s \Phi_{T,s+1}(G_s - \bar{G})\bar{g}_s$. A direct consequence is

$$\|e_T\| \le \sum_{s<T} \eta_s \|\Phi_{T,s+1}\|_{\mathrm{op}} \left\|(G_s - \bar{G})\bar{g}_s\right\|, \tag{10}$$

so the bound weights deviations from the reference curvature by the accumulated reference response.

B.7 From tangent energy to complete replacement

Write $z_T^{(i)} = n(w_T^{(i,1)} - w_T) = g_T^{(i)} + r_T^{(i)}$ for the scaled complete-replacement displacement. The following bound relates its energy to the tangent and nonlinear path remainder introduced in Section 5.

Theorem 7 (Complete-replacement energy bound)**.** *For $M_T \succeq 0$, write $D = D_T(M_T)$ and $R_{\mathrm{rem}} = \mathbb{E}[(r_T^{(i)})^\top M_T r_T^{(i)} \mid \mathcal{F}]$. Then*

$$\left(\max\{0, \sqrt{D} - \sqrt{R_{\mathrm{rem}}}\}\right)^2 \le \mathbb{E}[(z_T^{(i)})^\top M_T z_T^{(i)} \mid \mathcal{F}] \le \left(\sqrt{D} + \sqrt{R_{\mathrm{rem}}}\right)^2. \tag{11}$$

Let $z = g + r$ be random vectors and define the conditional Hilbert-space norm $\|z\|_{M,2} = (\mathbb{E}[z^\top M z \mid \mathcal{F}])^{1/2}$ for $M \succeq 0$. The triangle inequality gives $\|z\|_{M,2} \le \|g\|_{M,2} + \|r\|_{M,2}$ and the reverse-triangle inequality gives $\|z\|_{M,2} \ge \max\{0, \|g\|_{M,2} - \|r\|_{M,2}\}$. Squaring both relations proves (11). If the evaluation Hessian is L_3-Lipschitz on the segment from w_T to $w_T + z/n$, Taylor's theorem also

gives $|\tau| \le L_3 \|z\|^3/(6n^3)$ for the terminal remainder of the evaluation-loss expansion in Section 5. That control is separate from the path remainder r.

B.8 Positive-curvature interval and proof

This subsection constructs and proves the energy interval used in Section 5.4. For a pool $\mathcal{I}$ of K candidates, define $\widehat{D}_T(M_T) = K^{-1}\sum_{i\in\mathcal{I}}(\bar{g}_T^{(i)})^\top M_T \bar{g}_T^{(i)}$ under the uniform empirical candidate law. For $H = H_T$ and a scaled response $v = \bar{g}_T^{(i)}$, define $q_+(v) = v^\top H_+ v = (v^\top H v + v^\top\sqrt{H^2}v)/2$. The square root supplies absolute curvature, cancelling negative eigenvalues.

For nonzero v, k Golub–Kahan steps starting at $v/\|v\|$ produce a bidiagonal B_k and projected positive-semidefinite matrix $T_k = B_k^\top B_k$ (Golub & Kahan, 1965). Let T_k^{R} be its Gauss–Radau modification with a fixed eigenvalue zero. For either matrix T with eigenpairs (θ_j, s_j), define the quadrature value $\mathcal{Q}(T) = \|v\|^2\sum_j (e_1^\top s_j)^2\sqrt{\theta_j}$, where e_1 is the first coordinate vector (Golub & Meurant, 2010; Frommer et al., 2017).

Theorem 8 (Positive-curvature interval)**.** *For real symmetric H, the Gauss rule $\mathcal{G}_k = \mathcal{Q}(T_k)$ and zero-node Radau rule $\mathcal{R}_k = \mathcal{Q}(T_k^{\mathrm{R}})$ satisfy $\mathcal{R}_k \le v^\top\sqrt{H^2}v \le \mathcal{G}_k$ in exact arithmetic. Hence $L_k \le q_+(v) \le U_k$, where $L_k = \max\{0, (v^\top H v + \mathcal{R}_k)/2\}$ and $U_k = \max\{0, (v^\top H v + \mathcal{G}_k)/2\}$.*

Each step adds two HVPs and reuses the existing prefix. Set $\underline{L}_k = \max_{j\le k} L_j$ and $\overline{U}_k = \min_{j\le k} U_j$. Stop at zero width or when $2(\overline{U}_k - \underline{L}_k)/(\overline{U}_k + \underline{L}_k) \le \tau$ for requested tolerance τ. Averaged endpoints enclose $\widehat{D}_T((H_T)_+)$ for the fixed approximate responses. If symmetric numerical $\widehat{H}$ has known Frobenius error $\left\|\widehat{H} - H\right\|_F \le \varepsilon$, expand its interval downward and upward by $\varepsilon\|v\|^2$ to enclose the energy for H, clipping the lower bound at zero (Appendix B.9).

Let H be real symmetric, so that $H_+ = (H+\sqrt{H^2})/2$. For a nonzero v, let μ_v be the spectral measure of H^2 induced by v. Then $v^\top\sqrt{H^2}v = \|v\|^2\int_0^\infty \sqrt{x}\, d\mu_v(x)$. Golub–Kahan bidiagonalization with starting vector $v/\|v\|$ generates a right basis Q_k and a bidiagonal B_k with $Q_k^\top H^2 Q_k = B_k^\top B_k \succeq 0$ in exact arithmetic. Thus the projected measure has nonnegative nodes. For $f(x) = \sqrt{x}$ on $(0,\infty)$, the sign pattern of the derivatives makes the Gaussian quadrature remainder nonpositive. Therefore the Gauss rule is an upper bound and the Gauss–Radau rule with fixed left node zero is a lower bound. The case of a zero eigenvalue follows by applying the argument to $f_\epsilon(x) = \sqrt{x+\epsilon}$ and letting $\epsilon \downarrow 0$. The signed term $v^\top H v$ is evaluated in closed form. Adding this term and dividing by two gives the interval for $v^\top H_+ v$. Intersecting an interval with all valid earlier intervals preserves coverage and yields a nested sequence.

For completeness, the zero-node Radau modification changes only the last diagonal entry of T_k. Before Krylov termination, write $A = T_k[1:k-1, 1:k-1]$ and let β be its coupling to the last coordinate. For $k > 1$, set $(T_k^{\mathrm{R}})_{kk} = \beta^2 e_{k-1}^\top A^{-1} e_{k-1}$, retaining every other entry. Its Schur complement is zero, so T_k^{R} is positive semidefinite with a zero eigenvalue. For $k = 1$, take $T_1^{\mathrm{R}} = [0]$. At Krylov termination the projected spectral measure gives the exact quadratic form. The factor $\|v\|^2$ in both quadrature rules restores the scale removed by normalizing the starting vector.

B.9 Operator perturbation bound

This subsection gives the operator perturbation bound used in Section 5.4. The Frobenius norm is $\|E\|_F = (\sum_{a,b} E_{ab}^2)^{1/2}$.

Proposition 9 (Operator perturbation bound)**.** *Suppose H and $\widehat{H}$ are symmetric and $\|E\|_F \le \varepsilon$ for $E = \widehat{H} - H$. If $[\widehat{L}_k, \widehat{U}_k]$ contains $v^\top(\widehat{H})_+ v$, then $[\max\{0, \widehat{L}_k - \varepsilon\|v\|^2\},\ \widehat{U}_k + \varepsilon\|v\|^2]$ contains $v^\top H_+ v$.*

Let $\widehat{H} = H + E$ with H and $\widehat{H}$ symmetric. The positive part is the Frobenius projection onto the positive-semidefinite cone, and metric projection onto a closed convex set is nonexpansive, so

$\left\|H_+ - (\widehat{H})_+\right\|_F \le \|E\|_F$. If $\|E\|_F \le \varepsilon$ then

$$\left|v^\top H_+ v - v^\top (\widehat{H})_+ v\right| \le \left\|H_+ - (\widehat{H})_+\right\|_{\mathrm{op}} \|v\|^2 \le \varepsilon \|v\|^2 . \tag{12}$$

Padding both endpoints of an interval for $\widehat{H}$ by this amount proves the transfer statement.

B.10 Visit-pair interpretation and spectral propagation

This subsection develops the interpretation of the identities in Section 4.

Definition 1 measures the squared replacement response at the terminal model. We now identify how individual visits combine to produce that response. We derive the complete sum over visit pairs and characterize its cross-epoch terms under random reshuffling. We then relate finite-horizon propagation to inverse curvature and connect the tangent response to a complete replacement.

The complete path Gramian. Equation (1) separates propagation from the forcing created at each visit. To explain their contribution to FHSD, we unroll this recursion before taking its second moment. Theorem 2 applies to finite paths with time-varying curvature, including singular and indefinite batch Hessians. The proof unrolls (1) and then takes the outer product and conditional expectation. Each summand in the response transports one visit's contribution through all remaining updates. Squaring that sum pairs every visit with every other visit, producing the complete path Gramian W_T. Thus FHSD retains whether propagated visits reinforce or cancel in the terminal metric. The matrices A_t can act differently when their order is reversed. Keeping the ordered products therefore preserves information about the actual training sequence. The response recursion follows standard unrolled differentiation (Mlodozeniec et al., 2025). Here we use its complete conditional second moment to resolve interactions between visits on a realized path.

For comparison, retaining only the terms that pair each visit with itself gives

$$W_T^{\mathrm{diag}} = \sum_{s<T} \eta_s^2 \Phi_{T,s+1} C_{ss} \Phi_{T,s+1}^\top,$$

where W_T^{diag} is the diagonal-time approximation used in our comparisons. This matrix adds squared visit contributions without measuring their alignment. It equals W_T when $C_{su} = 0$ for all distinct times s and u. A replacement supplies the forcing at all of its visits, so their raw cross moments retain this shared source of variation. In particular, raw moments include mean contributions as well as centered covariances.

What random reshuffling removes and retains. Equation (2) includes every pair of training steps. We next identify which pairs can contribute when each epoch partitions the data into disjoint batches. Because one replacement acts only when its index is visited, the intersection of two batches determines their cross moment. The proof substitutes the forcing definition and averages its indicators over the uniformly sampled index. The corollary identifies terms that are zero within an epoch. Every example appears again in later epochs, preserving the pairs that describe cross-epoch interactions. We can therefore omit within-epoch cross-batch terms when computing the full moment.

Corollary 4 identifies the surviving terms without specifying their total magnitude. We bound that magnitude using the maximum number of visits to each candidate. Let $D_T^{\mathrm{diag}}(M_T) = \mathrm{Tr}(M_T W_T^{\mathrm{diag}})$ denote the sum of the single-visit energies in the terminal metric. The proof bounds the squared norm of a sum by the number of visits times the sum of squared norms. The proposition shows that cross-epoch alignment can amplify the diagonal-time energy by as much as the number of epochs. For equality, set every update Jacobian to ρ and choose the forcing so that each propagated contribution equals the same scalar $v \neq 0$. The full energy is then $E^2 v^2$, while the diagonal-time energy is $E v^2$. Conversely, two propagated contributions v and $-v$ give zero full energy and positive diagonal-time energy. Both reinforcement and cancellation concern the directions after propagation, even when every individual update contracts perturbations. Section 6.2 tests this alignment by comparing complete contributions with diagonal-time estimates under random reshuffling.

Flat modes retain a finite response. The visit contributions in (2) contain ordered propagation over a finite number of updates. To relate them to the inverse Hessian used by influence functions, consider

a quadratic objective with constant curvature (Koh & Liang, 2017). Let $G \succeq 0$ and $A = I - \eta G$, with step size $\eta > 0$ and constant forcing over T steps. For an eigenvalue λ satisfying $0 \leq \eta\lambda \leq 1$, the factor multiplying the negative forcing is

$$r_T(\lambda) = \begin{cases} [1 - (1 - \eta\lambda)^T]/\lambda, & \lambda > 0, \\ \eta T, & \lambda = 0, \end{cases}$$

where $r_T(\lambda)$ is the response accumulated along that eigendirection. The factor sums the effect of constant forcing over the remaining updates. Proposition 5 bounds it between $(1-e^{-1})\min\{\eta T, 1/\lambda\}$ and $\min\{\eta T, 1/\lambda\}$. At $\lambda = 0$, these bounds use the convention $1/0 = +\infty$. The filter therefore retains the finite response ηT in flat directions. For fixed positive λ, it approaches $1/\lambda$ as the horizon grows. This recovers the inverse-curvature factor after the response in that direction has saturated. The finite-horizon formula also represents directions whose responses are still accumulating. For varying curvature we retain the ordered propagation in (2). Section 6.3 compares its shared approximation with the unpropagated path and endpoint-forcing alternatives defined in Section 6.

From tangent energy to nonlinear replacement. Theorem 2 gives the exact derivative at the interpolation endpoint $\alpha = 0$. Our prediction target instead evaluates the complete replacement at $\alpha = 1$. To connect these endpoints, write the scaled parameter change as $z_T^{(i)} = n(w_T^{(i,1)} - w_T) = g_T^{(i)} + r_T^{(i)}$. Here $r_T^{(i)}$ is the nonlinear response remainder along the training path. Theorem 7 expresses its effect on the terminal energy. The proof applies the triangle and reverse-triangle inequalities to the metric-weighted responses. The square root of R_{rem} bounds how far the complete replacement's root-mean-square response can depart from that of the tangent. When $R_{\mathrm{rem}} = 0$, the complete replacement energy equals FHSD. Positive semidefiniteness supplies the nonnegative squared lengths needed for these inequalities.

To translate this parameter response into loss, suppose the evaluation loss is three-times differentiable. Its terminal Taylor expansion gives

$$\Delta_i L_Q = \frac{1}{n} \nabla L_Q(w_T)^\top z_T^{(i)} + \frac{1}{2n^2} (z_T^{(i)})^\top H_T z_T^{(i)} + \tau_i,$$

where τ_i is the Taylor remainder of the evaluation loss. The linear term reads out the response along the evaluation gradient. The quadratic term accounts for curvature, and τ_i contains the higher-order loss terms. This loss remainder is distinct from the path-response remainder $r_T^{(i)}$. Since $H_T \preceq (H_T)_+$, choosing $M_T = (H_T)_+$ bounds the average positive quadratic contribution through Theorem 7. For signed prediction we retain the full evaluation Hessian H_T. Section 5 uses this expansion to turn a shared approximation of the tangent response into candidate loss predictions.

B.11 Positive-curvature projection and quadrature

This subsection gives the projection and quadrature construction used in Section 5.4.

HVPs also support Hessian-free optimization (Martens, 2010). Structured curvature approximations use Kronecker factors (Martens & Grosse, 2015) and eigenbasis corrections (George et al., 2018). Here we use matrix-function evaluation (Higham, 2008) through Golub–Kahan projection and quadrature (Golub & Kahan, 1965; Golub & Meurant, 2010; Frommer et al., 2017). Recent work also analyzes randomized matrix-free quadrature (Chen et al., 2025).

The empirical FHSD in Section 5 requires positive-curvature quadratic forms when $M_T = (H_T)_+$. Computing them directly from all Hessian eigenvectors would require a full spectral decomposition. We instead reduce each quadratic form to a function of a positive-semidefinite matrix and evaluate it through HVPs. We then construct lower and upper bounds that determine when to stop the computation.

Reducing the positive part to a matrix square root. For this section, write $H = H_T$ for the real symmetric evaluation Hessian. Let $v \in \mathbb{R}^p$ be the queried response vector, such as $\bar{g}_T^{(i)}$ in the empirical FHSD. The signed quadratic $v^\top H v$ combines positive and negative curvature contributions that can cancel. A few leading positive eigenpairs retain only part of the energy across positive-curvature directions. We therefore evaluate the complete positive-curvature energy $q_+(v) = v^\top H_+ v$.

Using $H_+ = (H + \sqrt{H^2})/2$, we write this target as

$$q_+(v) = \frac{1}{2}\big(v^\top H v + v^\top \sqrt{H^2} v\big),$$

where $\sqrt{H^2}$ denotes the positive-semidefinite square root of H^2. The signed term costs one HVP, while the square-root term is a matrix-function quadratic form. This combination cancels the negative-curvature contributions and retains the positive ones. For $v = 0$, the target is zero and requires no iteration.

Computing a positive-semidefinite projection. The square-root term above depends on the full spectrum of H^2. To approximate it using HVPs, we use Golub–Kahan bidiagonalization (Golub & Kahan, 1965). This method alternates products with H and $H^\top$ to build a small bidiagonal representation. Because H is symmetric, both products are HVPs and the resulting projection of H^2 preserves nonnegative curvature. For nonzero v, start from the unit vector $q_1 = v/\|v\|$ and run k steps. Let Q_k denote the right orthonormal basis and B_k the bidiagonal projection. In exact arithmetic their relation is

$$\mathsf{Q}_k^\top H^2 \mathsf{Q}_k = \mathsf{B}_k^\top \mathsf{B}_k \succeq 0.$$

The small matrix is positive semidefinite because it has the form $\mathsf{B}_k^\top \mathsf{B}_k$. We thus apply the standard bidiagonalization tool to the square-root representation of H_+ while preserving the sign structure needed for bounds.

Bounding the energy and choosing the computation budget. The projection $\mathsf{B}_k^\top \mathsf{B}_k$ makes the square-root term accessible through a small matrix. To obtain an interval for the full quadratic form, we use Gauss and Gauss–Radau quadrature (Golub & Meurant, 2010; Frommer et al., 2017). These rules approximate a weighted sum of square roots using nodes and weights obtained from the projection. The nodes approximate eigenvalues of H^2, and the weights describe the contribution of v at those values. For this function, Gauss gives an upper bound and Gauss–Radau with a fixed node at zero gives a lower bound. The proof expresses the quadratic form as a weighted sum over eigenvalues and uses the signs of the square-root function's derivatives. The theorem bounds all positive-curvature contributions along v using the same HVP sequence. Symmetry makes H^2 positive semidefinite, providing the nonnegative spectral support required by the quadrature rules. The role of k is computational: more steps refine the interval without changing the target metric.

C THE PATHGRAM PROCEDURE

This appendix specifies the shared propagation procedure and cost comparison described in Section 5. The reported estimator-to-retraining ratios exclude base-path training and state storage. For K candidates and T updates, forcing projections cost $O(KTpq)$, leading-coordinate updates $O(KTq)$ and orthogonal accumulation $O(KTp)$, in addition to gradient evaluation and shared eigenspace preparation.

Projecting the reference recursion onto U and its orthogonal complement gives the coordinate updates. Here I_q is the identity on the leading subspace.

$$a_{t+1}^{(i)} = (I_q - \eta_t \Lambda) a_t^{(i)} - \eta_t U^\top \xi_t^{(i)},$$
$$h_{t+1}^{(i)} = h_t^{(i)} - \eta_t (I - P) \xi_t^{(i)},$$

Table 6: Work shared across K candidate replacements. The HVP count for the terminal eigenspace depends on the iterative eigensolver tolerance.

Method	Training paths	Shared HVP	Candidate HVP
Neighbor retraining	K	0	0
Exact path tangent	1	0	KT
Endpoint influence	1	iterative solve	K gradients
PATHGRAM	1	eigenspace	K

Algorithm 1 PATHGRAM for many candidate replacements

Require: Base training set and schedule, candidates $\mathcal{I}$, rank q
1: Train the base model and retain the states needed to evaluate candidate forcing
2: Compute the leading terminal training eigenspace (U, Λ) using HVPs
3: **for** each training step t **do**
4: Evaluate $\{\xi_t^{(i)} : i \in \mathcal{I}\}$ in candidate batches
5: Update $a_t^{(i)}$ and $h_t^{(i)}$ by (3)
6: **end for**
7: **for** each candidate $i \in \mathcal{I}$ **do**
8: Set $\bar{g}_T^{(i)} = U a_T^{(i)} + h_T^{(i)}$
9: Compute (4) with an evaluation-loss HVP
10: **end for**
11: **return** signed loss predictions and optional FHSD intervals

D OFF-PATH QUERY COMPLEXITY

This appendix characterizes the information needed to predict a complete replacement from a recorded base path.

Theorem 10 shows why a base-path derivative transcript cannot uniformly determine an unrestricted neighboring trajectory. The result is a worst-case statement for a smooth function class.

Theorem 10 (Matched off-path query complexity)**.** *For every integer $m \geq 1$ there is a family of nonnegative, bounded, C^∞ empirical risks on $\mathbb{R}^{m+1}$ and a bounded C^∞ evaluation loss such that the following hold.*

1. *Every family member has the same base gradient-descent path and the same finite-order derivative transcript in a neighborhood of that path.*
2. *Any deterministic adaptive algorithm using $q < m$ arbitrary finite-order off-path queries has worst-case terminal-loss error at least $1/4$.*
3. *Any randomized algorithm using $q < m$ such queries has worst-world expected error at least $1/4$.*
4. *A budget of m gradient queries at fixed public points recovers the neighboring terminal loss.*

Proof. Write the state as $w = (t, x_1, \ldots, x_m)$ and choose $0 < r < 1/2$. Define the compactly supported bump

$$\rho_r(u) = \begin{cases} \exp\left(1 - \dfrac{1}{1 - (u/r)^2}\right), & |u| < r, \\ 0, & |u| \geq r, \end{cases} \tag{13}$$

which is C^∞, satisfies $\rho_r(0) = 1$, and has disjoint translates centered at the integers. Index each world by $\sigma = (\sigma_1, \ldots, \sigma_m) \in \{-1, +1\}^m$ and define the base risk and the common replacement increment

$$R_\sigma(w) = 2 + \sum_{j=1}^{m} \rho_r(t - j)\left[-\sin(t - j) - \sigma_j \sin(x_j)\right], \qquad h(w) = 1 - \rho_r(t) \sin t. \tag{14}$$

At most one translated bump is active at any point, hence $0 \leq R_\sigma \leq 4$ and $0 \leq R_\sigma + h \leq 6$.

Run unit-step gradient descent from the origin. Every R_σ is constant in a neighborhood of the origin, so the base path remains at the origin in all worlds. The increment in (14) moves the neighboring state from the origin to $(1, 0, \ldots, 0)$. At the public state whose time coordinate is j, the jth term of R_σ moves the time coordinate to $j + 1$ and the coordinate x_j to σ_j, so after $m + 1$ updates the neighboring terminal state is $(m + 1, \sigma_1, \ldots, \sigma_m)$. Use the bounded evaluation loss

$$L(w) = \frac{1}{2} + \frac{1}{4} \prod_{j=1}^{m} \sin\left(\frac{\pi x_j}{2}\right). \tag{15}$$

At the neighboring terminal state this loss equals $1/2 + (1/4)\prod_j \sigma_j$, whose two possible values are $1/4$ and $3/4$.

Because the time slabs are disjoint, a finite-order query at any point can depend on at most one bit σ_j. After $q < m$ adaptive queries at least one bit remains unobserved. Fix all other bits and flip the unobserved one. The entire query transcript is unchanged while the terminal losses differ by $1/2$, so one of the two absolute errors is at least $1/4$. For randomized algorithms, draw σ uniformly from $\{-1, +1\}^m$. Conditional on any transcript containing fewer than m bits the parity remains balanced, and the minimum conditional expected absolute error for a variable equally likely to be $1/4$ and $3/4$ is $1/4$. Yao's principle then yields a world with expected error at least $1/4$. For the matching upper bound, query ∇R_σ at $p_j = (j, 0, \ldots, 0)$ for $j = 1, \ldots, m$. The x_j component equals $-\sigma_j$, so these m queries recover every bit, the neighboring terminal state and (15). □

E Experimental Configuration

This appendix provides the data splits and numerical settings for the comparisons in Section 6.

Table 7: Experimental settings. K is the number of candidate replacements evaluated against neighboring retraining, RR denotes random reshuffling, and the BERT and Swin-T rows adapt the final attention block with LoRA and a rank-two residual adapter respectively. The scaled study of rows two and four appears in Table 10.

Model	Data	Trainable component	Optimizer	n	K	Primary question
Softplus MLP	Fashion-MNIST	all parameters	full batch	512	10 bases	cross-time moment
BERT-tiny/base	20 Newsgroups	LoRA + head	full batch	96	24	signed effect
BERT-tiny/base	20 Newsgroups	LoRA + head	RR SGD	96	12×2	cross-epoch reuse
Swin-T	CIFAR-10 binary	adapter + head	full batch	96	16	cross-family
ResNet-20	CIFAR-10	all parameters	momentum	6000	3×2	positive curvature

The ResNet-20 base models use SGD with learning rate 0.05 and momentum 0.9. Weight decay is 5×10^{-4} and batch size is 128. The learning rate drops after epochs 20 and 27. Each of three independently initialized models contributes two replacement directions, giving six matrix-function evaluations.

E.1 Transfer configurations at $n = 96$

This subsection supports the primary prediction comparisons in Section 6.3.

Table 8 records the optimization horizons and trainable dimensions. All BERT configurations use LoRA rank and scale equal to four, weight decay 10^{-3}, and the same 96-example training split. Full-batch BERT uses 24 candidate pairs. The reshuffling configurations use 12 candidate pairs with two fixed schedules and eight Rademacher probes for the diagonal-time moment. Swin-T uses ten terminal training-Hessian modes.

Table 8: Transfer-training configurations. The product ηT is matched within each BERT full-batch or reshuffling comparison.

Configuration	Batch	Epochs	Steps	Learning rate	Trainable dim.	Spectral rank
BERT-tiny, full batch	96	80	80	0.01000	2,306	6
BERT-base, full batch	96	320	320	0.00250	13,826	6
BERT-tiny, reshuffling	48	80	160	0.00500	2,306	6
BERT-base, reshuffling	48	320	640	0.00125	13,826	6
Swin-T adapter	96	80	80	0.06000	4,610	10

The 20 Newsgroups task uses the `rec.sport.baseball` and `sci.space` categories, and evaluation contains 128 examples. The Swin-T task uses the CIFAR-10 airplane and automobile classes with 48 training images per class and 256 evaluation images. Its fixed backbone maps images to normalized 768-dimensional pooled features before the trainable adapter.

E.2 SCALED STUDY

This subsection specifies the enlarged settings evaluated in Section 6.5.

The enlarged text study retains the evaluation procedure and model hyperparameters while changing the training-set size. BERT-tiny uses $\eta = 0.01$ with 80 steps. BERT-mini and BERT-medium use $\eta = 0.005$ with 160 steps. BERT-base uses $\eta = 0.0025$ with 320 steps. These settings match the corresponding rows of the controlled scale panel. All scaled text runs use $n = 1{,}000$ and 24 candidate replacements with $k = 6$. LoRA rank and scale are four. Maximum sequence length is 96 and weight decay is 10^{-3}. Eigensolver tolerance is 10^{-3} with at most 160 iterations. Three independent repetitions pair data seeds $170, 171, 172$ with initialization seeds $5, 6, 7$. All four scales use all three. The training and replacement pools are drawn from the 1,190 training documents in these two classes.

The vision study uses a fixed feature cache with 64 pool images per class instead of 24. It keeps the data seed and the evaluation seed. The balanced training prefix contains 96 images and the evaluation set contains 256 images. The settings remain $\eta = 0.06$ with 80 steps, adapter rank two and $k = 10$. The 48 pool images of the primary study are a subset of the 128 enlarged pool images. Because the candidate pool is permuted before pairs are drawn, enlarging the pool re-draws the candidate set. The Swin-T enlarged-pool row in Table 10 is therefore a fresh $K = 16$ draw from the 128-image pool. It serves as the within-study reference for the $K = 64$ and $K = 96$ rows. The Swin-T adapter row of the primary block uses the original 48-image pool.

E.3 POSITIVE-CURVATURE COMPUTATION

This subsection supplies the numerical settings and interval-width measurements summarized in Section 6.7.

The ResNet-20 experiment uses HVP budgets in $\{96, 128, 160, 192, 224, 256\}$, corresponding to 48 through 128 Golub–Kahan steps. Full reorthogonalization is applied to the small basis. Table 9 reports relative widths across all six directions. Every direction reaches 10% width at either 160 or 192 HVPs. Reported numerical brackets include the implementation's finite-precision padding, whose largest value is 2.31×10^{-8} relative to the estimate. Proposition 9 describes the separate operator-level condition needed to transfer such a bracket to an ideal exact HVP operator. We compare the leading 20 positive eigenmodes with the midpoint of each 256-HVP interval. Their captured energy ranges from 8.1% to 22.0% across the six directions, with median 16.4%. This comparison measures the positive-curvature contribution distributed beyond the leading modes.

Table 9: Relative width of the positive-curvature intervals across six ResNet directions.

HVPs	Median width	Maximum width
96	0.23	0.27
128	0.14	0.17
160	0.10	0.11
192	0.07	0.08
224	0.05	0.06
256	0.041	0.047

E.4 SHARED PROPAGATION IMPLEMENTATION

This subsection details the shared eigenspace and response computation in Section 5.

A direct forward evaluation of every candidate response in (1) uses one Hessian–vector product (HVP) per step and candidate. We reduce this cost by propagating candidate contributions through one shared leading Hessian eigenspace. We call the resulting estimator PATHGRAM and retain each candidate's complete forcing sequence. We describe this shared propagation and its error identity before constructing the signed loss prediction and empirical FHSD.

Sharing curvature across candidates. The repeated HVPs in (1) arise from applying the changing batch Hessian to each candidate response. We replace these Hessians with a common low-rank representation of the full training objective's Hessian at w_T, including weight decay. Let $U \in \mathbb{R}^{p \times q}$

contain orthonormal eigenvectors for its q largest algebraic eigenvalues and let the diagonal matrix $\Lambda \in \mathbb{R}^{q\times q}$ contain their eigenvalues. The rank q controls how many curvature directions receive explicit propagation. This section gives the spectral ranks and eigensolver settings used in the experiments. We obtain this eigenspace using HVPs (Pearlmutter, 1994). An HVP differentiates the gradient in a vector direction, allowing the eigensolver to access curvature without forming the Hessian. Set $P = UU^\top$ for the projector onto the leading subspace and $\bar{G} = U\Lambda U^\top$ for the shared reference Hessian.

We retain the forcing in (1) and replace only its propagation Hessian by $\bar{G}$. The approximate scaled tangent response follows $\bar{g}_{t+1}^{(i)} = (I - \eta_t \bar{G})\bar{g}_t^{(i)} - \eta_t \xi_t^{(i)}$ with $\bar{g}_0^{(i)} = 0$. To evaluate this recursion efficiently, split the response into its leading-subspace coordinates $a_t^{(i)} = U^\top \bar{g}_t^{(i)}$ and orthogonal component $h_t^{(i)} = (I - P)\bar{g}_t^{(i)}$. Both components start at zero and evolve as

$$a_{t+1}^{(i)} = (I_q - \eta_t \Lambda) a_t^{(i)} - \eta_t U^\top \xi_t^{(i)},$$
$$h_{t+1}^{(i)} = h_t^{(i)} - \eta_t (I - P)\xi_t^{(i)},$$

where I_q is the identity matrix on the q-dimensional leading subspace. The leading component propagates earlier contributions according to their reference curvature before adding the current forcing. The orthogonal component accumulates forcing while leaving its directions unchanged. Reconstructing $\bar{g}_T^{(i)} = U a_T^{(i)} + h_T^{(i)}$ therefore retains both components of the candidate response. Setting $\bar{G} = 0$ recovers the unpropagated-path response, while dropping $h_T^{(i)}$ gives the top-subspace alternative. These are the component comparisons in Section 6.

Error from the shared propagation. Equation (3) replaces the true curvature sequence with a common reference. We now identify which deviations from that reference affect the terminal response. Subtracting the exact and reference recursions gives the following deterministic identity.

$$g_T^{(i)} - \bar{g}_T^{(i)} = -\sum_{s<T} \eta_s \Phi_{T,s+1}(G_s - \bar{G})\bar{g}_s^{(i)}.$$

The proof unrolls the difference recursion in Appendix B. The factor $(G_s - \bar{G})\bar{g}_s^{(i)}$ measures how curvature mismatch acts on the response accumulated by step s. The ordered product $\Phi_{T,s+1}$ then transports this discrepancy to the terminal model. For a given path, choosing the reference eigenspace changes the curvature mismatch represented in this identity. Thus the approximation depends on matching curvature along accumulated response directions. If $G_s = \bar{G}$ at every step, the reference and exact tangent responses coincide.

Signed prediction and candidate averages. Equation (3) supplies a terminal response estimate for each candidate. We turn it into the target loss change using the Taylor expansion in Section 5. Substituting $\bar{g}_T^{(i)}/n$ for the parameter displacement gives the prediction

$$\widehat{\Delta_i L_Q} = \frac{1}{n}\nabla L_Q(w_T)^\top \bar{g}_T^{(i)} + \frac{1}{2n^2}(\bar{g}_T^{(i)})^\top H_T \bar{g}_T^{(i)},$$

where $\widehat{\Delta_i L_Q}$ estimates the signed effect of fully replacing example i. The linear term measures change along the evaluation gradient, while the quadratic term accounts for the evaluation curvature along the estimated displacement. The quadratic term requires one evaluation-loss HVP per candidate and retains both signs of curvature. Negative predictions indicate expected improvement, and smaller values rank candidates more favorably.

To summarize sensitivity over a finite candidate set $\mathcal{I}$, let $K = |\mathcal{I}|$ denote its size. We compute an empirical estimate of Definition 1 directly from the responses as

$$\widehat{D}_T(M_T) = \frac{1}{K}\sum_{i\in\mathcal{I}} (\bar{g}_T^{(i)})^\top M_T \bar{g}_T^{(i)},$$

where each summand is an approximate scaled tangent response energy. Equal weights specify the uniform empirical candidate law. This average avoids forming the full response matrix W_T. For $M_T = (H_T)_+$, Section 5.4 supplies a matrix-free interval for each summand.

Procedure and cost. Equations (3) and (4) specify the propagation and readout used in Algorithm 1 of Appendix C. We train the base model and retain its states for candidate-gradient evaluation. We then compute the shared terminal eigenspace and accumulate each candidate's response along the recorded schedule. Finally we apply the terminal loss expansion to obtain signed predictions. All estimator inputs come from the base path and the proposed replacement examples.

Direct comparison trains K neighboring models for T steps each. PATHGRAM uses one base path and one shared terminal eigenspace. It evaluates at most KT candidate gradient differences and uses K terminal HVPs for signed prediction. Only visits to a candidate contribute nonzero forcing to its response. After projecting the forcing, the leading-coordinate recurrence costs $O(KqT)$ arithmetic. Projection and orthogonal-component accumulation additionally act on the p-dimensional vectors, without storing a $p \times p$ matrix. A candidate pool shares the cost of the base path and eigenspace. Section 6.6 reports timing comparisons on the same candidates and hardware. The resulting reference responses also define the empirical FHSD whose positive-curvature summands we evaluate in Section 5.4.

E.5 INTERVAL INTERSECTION AND STOPPING

This subsection describes the stopping procedure for the intervals in Section 5.4.

To retain the information from all completed steps, we intersect their intervals as

$$\underline{L}_k = \max_{j \le k} L_j, \qquad \overline{U}_k = \min_{j \le k} U_j,$$

where $\underline{L}_k$ and $\overline{U}_k$ are the lower and upper endpoints after intersection. The intersected intervals remain valid and are nested as the HVP budget grows. We stop at a requested relative width measured against the interval midpoint. A zero-width interval already determines the value of the quadratic form. This section gives the budgets and requested accuracy used in the experiments.

For numerical products represented by a nearby symmetric operator $\widehat{H} = H + E_H$, an operator-error bound transfers the interval to H. Here E_H denotes the symmetric perturbation and $\|E_H\|_F$ is its Frobenius norm. Padding both endpoints by $\|E_H\|_F \|v\|^2$ covers $v^\top H_+ v$, with the lower endpoint clipped at zero (Proposition 9). This follows from nonexpansiveness of projection onto the positive-semidefinite cone in Frobenius norm. Averaging the candidate intervals bounds the empirical FHSD from Section 5. These intervals complete its positive-curvature evaluation, whose interval width and HVP cost we assess in Section 6.

E.6 EXPERIMENTAL DESIGN

This subsection expands the setting choices and comparison definitions in Section 6.

We evaluate the introduction's claims about cross-epoch interactions and the prediction of replacement effects on a fixed evaluation set Q. Section 6.2 tests the visit-pair decomposition of Theorem 2 by comparing complete and diagonal-time contributions. Section 6.3 assesses signed loss predictions and the resulting choices of candidate corrections. Section 6.4 connects the error decomposition to four-checkpoint refinement. Section 6.5 examines larger training and candidate sets, and Section 6.6 evaluates computation costs. Section 6.7 tests the positive-curvature computation of Theorem 8.

Data and models. We use binary text and image classification to compare predictions with complete neighboring retraining. For text we use the baseball and space categories of 20 Newsgroups (Lang, 1995). The primary full-batch comparison uses 96 training documents drawn uniformly at random and 24 candidate replacements. We measure each replacement's loss change on 128 evaluation documents. This setting makes candidate-level comparisons with the exact path tangent computationally accessible.

We adapt BERT-tiny (Turc et al., 2019) and BERT-base through rank-four LoRA modules on the final attention layer's query and value projections, with a trainable classification head (Devlin et al., 2019; Hu et al., 2022). Using the same adaptation location isolates propagation across different backbone sizes. We match full-batch horizons at gradient-flow time 0.8 so that the step-size comparisons cover the same accumulated training time. For random reshuffling (RR), we retain the data and adapted

modules and partition each epoch into batches of 48 examples. This comparison isolates repeated visits under changing batch order.

For vision we hold an ImageNet-pretrained Swin-T backbone fixed and train a residual tanh adapter with a binary head (Liu et al., 2021; Krizhevsky, 2009). We use CIFAR-10 (Krizhevsky, 2009) airplanes and automobiles to test replacement prediction with a different modality and adaptation mechanism. The adapter has rank two. We use 96 balanced training images and 256 evaluation images. The primary candidate set and the enlarged candidate pools appear in Table 10.

We enlarge text training to $n = 1{,}000$ and include BERT-mini and BERT-medium (Turc et al., 2019) to test whether prediction quality persists with more training data. Each scaled text configuration uses three independent data and initialization runs to measure variation across training instances. For vision we draw candidate sets from a common pool of 128 images to study the effect of sharing computation across more replacements. We also use a separate BERT scale comparison with matched adaptation settings and its own candidate draws. This section gives the splits and optimization settings in Tables 7 and 8.

To compare temporal aggregation, we use softplus multilayer perceptrons on Fashion-MNIST with fixed terminal top-20 eigenspaces and the same spectral propagation (Xiao et al., 2017). We evaluate ten independent base models to compare magnitude accuracy and ordering across training instances. For positive-curvature computation we use GroupNorm ResNet-20 on CIFAR-10, which permits evaluation of the metric through HVPs (Pearlmutter, 1994) (He et al., 2016; Wu & He, 2018). Three independently initialized models contribute six replacement directions to measure variation in interval width. The moment comparison holds the terminal spectrum and propagation rule fixed, while the ResNet study evaluates the positive-curvature calculation through HVPs.

Reference methods. We compare PATHGRAM with component alternatives to identify which information supports prediction and selection. The *top subspace* alternative drops the forcing orthogonal to the leading terminal eigenspace, testing the contribution of the complement in (3). The *unpropagated path* accumulates each forcing vector without curvature propagation, testing the effect of propagating earlier visits. The *endpoint forcing* alternative repeats the last observed forcing at every visit without propagation, testing the contribution of forcing variation. The exact path tangent (Mlodozeniec et al., 2025) uses the true base-path Hessian at every step and serves as an oracle for the derivative response. We apply the same terminal loss expansion to this response when comparing signed predictions. Every neighboring retraining (Bousquet & Elisseeff, 2002) run shares its base model's initialization and realized data order so that the target remains the replacement effect $\Delta_i L_Q$.

The *diagonal-time* comparison retains only the $s = u$ terms of (2), isolating the effect of visit-pair interactions. For squared linear PATHGRAM predictions under RR, we estimate this quantity using eight independent random epoch-sign draws per candidate and schedule. Each candidate is evaluated over two realized schedules to compare the effect across data orders. We separately apply these signs before the quadratic readout and compare squared predictions with squared retraining loss changes after schedule averaging. On Fashion-MNIST we compare the full second moment of propagated base-path forcing with two peak envelopes. The same-path envelope uses the peak of each modal forcing sequence. The separated envelope takes peaks of the neighboring-path direct forcing and curvature residual separately (Appendix F.9). For curvature we compare $v^\top H_+ v$ with the signed quadratic (Pearlmutter, 1994) and the leading 20 positive eigenmodes (Higham, 2008) to assess the information retained by simpler summaries.

Prediction and selection metrics. For predicted and measured loss changes, normalized mean absolute error is $\mathrm{NMAE} = \sum_i |\widehat{\Delta}_i - \Delta_i| / \sum_i |\Delta_i|$. Lower NMAE means more accurate magnitudes relative to the measured signal. Spearman correlation (Spearman, 1904) measures agreement in candidate ordering, and sign accuracy is the fraction of predictions with the correct loss-change sign. Higher values are better for both measures. Evaluation accuracy is the fraction of correctly classified examples in Q, with higher values indicating better base-model performance. For the moment comparison we apply NMAE to the schedule-averaged squared quantities. For the Fashion-MNIST comparison, multiplicative error is $\max(\widehat{E}/E, E/\widehat{E})$, where E is the measured projected replacement energy and $\widehat{E}$ is its estimate. Values closer to unity indicate better magnitude agreement.

We select the k candidates with the smallest predicted loss changes to evaluate the correction decision. Precision@k is the selected fraction belonging to the oracle set A^* of the smallest retraining loss changes, so higher is better. This oracle uses all neighboring outcomes and specifies the optimal selection for the measured target. We use $k = 6$ for full-batch text and $k = 3$ for RR text. For the primary and enlarged vision sets we use $k = 10$. The external Swin-T comparison uses the same primary candidates with $k = 6$ for every method. These choices compare selection from each candidate pool at its stated retrieval size.

To measure the consequence of a selection error, let $r = \max_i \Delta_i - \min_i \Delta_i$ be the measured loss-change range. The normalized regret of a selected set A is

$$\mathrm{Reg}(A) = \frac{|A|^{-1} \sum_{i \in A} \Delta_i - |A^*|^{-1} \sum_{i \in A^*} \Delta_i}{r}.$$

Lower regret means less additional loss relative to oracle selection. Unlike set overlap, it distinguishes missing a nearly equivalent correction from missing a substantially better one. This connects prediction accuracy to the benefit of the selected corrections.

Computation metrics. We report estimator time divided by the time for all neighboring retraining runs on the same candidates and hardware. A smaller ratio means greater savings from shared computation. For a curvature interval $[L, U]$, we measure relative width as $(U - L)/|\widehat{q}_+|$ with midpoint $\widehat{q}_+ = (L + U)/2$. Smaller widths give more precise positive-curvature estimates. We report width distributions over the replacement directions and the HVP budget needed to reach 10% width, measuring the cost of the requested numerical accuracy.

F Additional Results

This appendix gives the candidate-level results and component comparisons summarized in Section 6.

F.1 Temporal moments under random reshuffling

This subsection supports the cross-epoch comparison in Section 6.2. Energy ratios use squared linear predictions. Moment NMAE compares schedule-averaged squared quadratic predictions with squared retraining changes. Downward arrows indicate smaller preferred error.

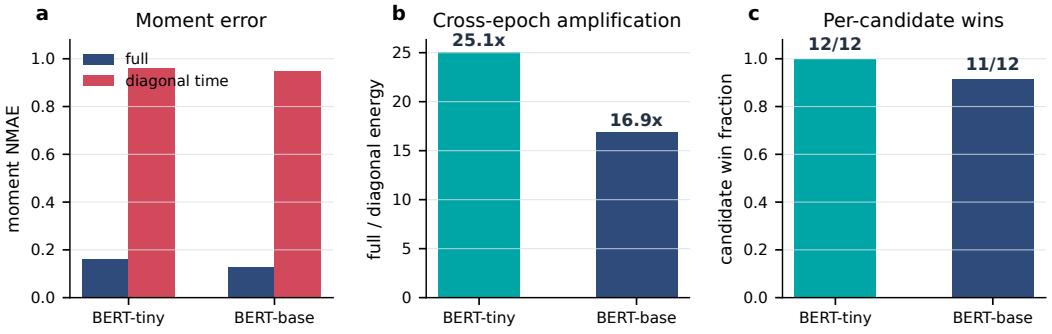


Figure 5: Temporal correlation under random reshuffling. (a) NMAE for squared predictions from the quadratic loss expansion, using the original epoch contributions (full) or independent sign randomization (plotted as "random sign"). Predictions and retraining loss changes are squared before averaging over schedules. (b) Ratio of squared linear predictions, averaged over candidates and schedules, to their estimated diagonal-time part. (c) Fraction of candidates with smaller absolute error under the original contributions.

Figure 5 extends the main-text comparison with amplification ratios and the fraction of candidates improved by preserving epoch alignment. The two BERT models agree in all three views: cross-epoch terms amplify the linear-response energy, and retaining their alignment improves squared-loss prediction for most candidates. Energy ratios use the evaluation-gradient readout and average over candidates and schedules. By Theorem 2 and Corollary 4, their excess over the diagonal-time part consists of cross-epoch terms.

F.2 SIGNED PREDICTIONS ON THE FULL-BATCH SETTINGS

This subsection supports the signed-loss comparison in Section 6.3.

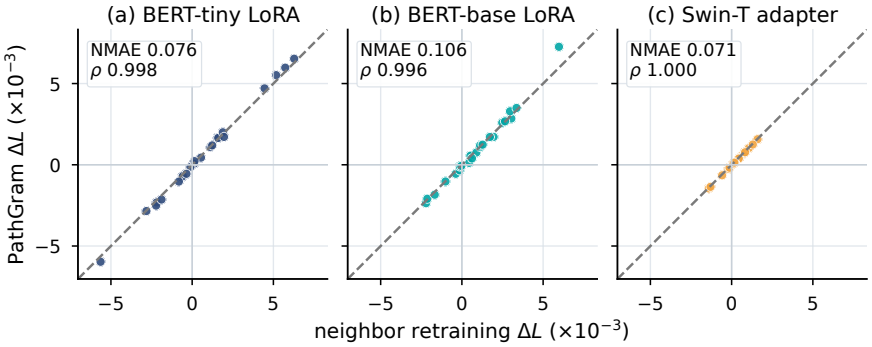


Figure 6: Predicted and measured signed terminal-loss changes on the three full-batch settings, in units of 10^{-3} on both axes. The dashed diagonal is exact agreement and ρ is Spearman rank correlation.

Figure 6 displays every primary full-batch candidate behind Table 1. The diagonal permits direct inspection of magnitude agreement, while the spread on either side of zero separates beneficial from harmful replacements. This candidate-level view complements the aggregate errors and selection metrics.

F.3 ENLARGED TRAINING SETS AND CANDIDATE POOLS

This subsection expands the training-size and candidate-pool comparisons in Section 6.5.

Table 10 summarizes the enlarged settings, including exact-tangent errors, signal ranges and cost ratios. Table 11 reports every enlarged-text run, making seed variation and oracle-set recovery directly visible. Each row evaluates 24 replacements and selects six under its own matched retraining outcomes. Eight runs recover all six oracle members, and the other four recover five. The per-run comparison distinguishes magnitude accuracy from ordering and selection.

Table 10: Complete enlarged-setting comparison. Acc. is base-model evaluation accuracy (averaged across seeds for text). Rows marked reference are the $n = 96$ settings of Table 1. The primary Swin-T pool holds 48 images and the rebuilt pool holds 128. The setting column gives the number of data and initialization seeds for each scaled text row. Multi-seed rows give the mean with the population standard deviation in parentheses. The gold signal range r is given in units of 10^{-3} (seed-wise minimum and maximum where more than one seed is run). The Tangent column is the exact path-tangent reference of Section 6, ρ is Spearman rank correlation and Prec is precision@k with $k = 6$ for text and $k = 10$ for vision.

Model	Setting	n	K	r	NMAE ↓	Tangent	ρ ↑	Sign ↑	Prec	Cost ↓	Acc.
BERT-tiny	reference	96	24	11.92	0.0762	0.0533	0.9983	1.000	1.000	0.065	0.820
BERT-base	reference	96	24	8.18	0.1060	0.0723	0.9957	1.000	1.000	0.026	0.734
BERT-tiny	3 seeds	1000	24	0.46–0.73	0.060 (0.022)	0.0069	0.9968	0.972	0.944	0.058	0.839
BERT-mini	3 seeds	1000	24	0.75–1.42	0.0280 (0.0056)	0.0096	0.9977	0.986	0.944	0.044	0.857
BERT-med.	3 seeds	1000	24	0.80–1.62	0.160 (0.055)	0.0169	0.9835	0.958	0.889	0.043	0.878
BERT-base	3 seeds	1000	24	0.42–1.58	0.079 (0.027)	0.0099	0.9968	0.986	1.000	0.012	0.734
Swin-T	primary pool	96	16	3.00	0.0713	0.0795	1.0000	1.000	1.000	0.744	0.957
Swin-T	rebuilt pool	96	16	3.48	0.0813	0.0880	0.9971	1.000	1.000	0.343	0.957
Swin-T	scaled pool	96	64	3.92	0.0801	0.0888	0.9993	1.000	1.000	0.078	0.957
Swin-T	scaled pool	96	96	3.92	0.0771	0.0857	0.9996	0.990	1.000	0.151	0.957

The quadratic readout lowers NMAE in eight of the twelve runs, with relative reductions of 1.3%–20.0%, and raises it by at most 6.4% in the other four. The signal ranges in Table 10 provide the scale of the corresponding replacement effects. For vision, the same table separates the primary candidate

Table 11: **Every enlarged-text run.** Each data and initialization seed uses $n = 1{,}000$, 24 replacements and top-six selection. Tangent is the exact path-tangent NMAE, ρ is Spearman correlation, Hits counts recovered oracle members, and Cost is estimator time relative to neighboring retraining. Regret uses the within-run retraining signal range.

Model	Seed	NMAE	Tangent	ρ	Hits	Regret	Cost
BERT-tiny	170	0.0377	0.0051	0.9957	6/6	0.0000	0.035
BERT-tiny	171	0.0901	0.0083	0.9948	5/6	0.0007	0.077
BERT-tiny	172	0.0513	0.0072	1.0000	6/6	0.0000	0.062
BERT-mini	170	0.0328	0.0070	0.9974	6/6	0.0000	0.035
BERT-mini	171	0.0201	0.0095	0.9991	6/6	0.0000	0.039
BERT-mini	172	0.0311	0.0123	0.9965	5/6	0.0017	0.059
BERT-medium	170	0.0927	0.0141	0.9843	6/6	0.0000	0.021
BERT-medium	171	0.1615	0.0213	0.9852	5/6	0.0013	0.056
BERT-medium	172	0.2264	0.0154	0.9809	5/6	0.0136	0.054
BERT-base	170	0.0416	0.0099	0.9974	6/6	0.0000	0.013
BERT-base	171	0.0890	0.0092	0.9957	6/6	0.0000	0.010
BERT-base	172	0.1066	0.0107	0.9974	6/6	0.0000	0.013

pool from the enlarged pool used for the candidate-count comparison. All three enlarged-pool settings recover their requested oracle sets while retaining a shared eigenspace.

F.4 Replacement decisions and reduced baselines

This subsection gives the component comparisons supporting the selection result in Section 6.3.

To identify the information supporting correction selection, Table 12 removes the orthogonal response, curvature propagation or temporal forcing variation from the primary BERT calculation. PathGram recovers the oracle top-six set on both models. The unpropagated and endpoint-forcing alternatives use uncalibrated amplitudes, so this table evaluates their rankings and decisions. Their Spearman correlations range from 0.060 to 0.538, showing the effect of removing propagation or forcing variation.

Table 12: Full-batch top-six replacement decisions at $n = 96$.

Model	Method	Precision@6	Norm. regret	Spearman	Sign acc.
BERT-tiny	PathGram	1.000	0.0000	0.998	1.000
	Top subspace	0.833	0.0099	0.890	0.750
	Unpropagated path	0.500	0.1508	0.538	0.542
	Endpoint forcing	0.333	0.1543	0.515	0.542
BERT-base	PathGram	1.000	0.0000	0.996	1.000
	Top subspace	0.500	0.0864	0.716	0.875
	Unpropagated path	0.167	0.2140	0.077	0.667
	Endpoint forcing	0.167	0.2140	0.060	0.667

Figure 7 contrasts the decision effects of the component removals in Table 12. The orthogonal component has a larger effect on the BERT-base selection, while retaining the full propagated forcing sequence improves both models. Table 13 supplies the corresponding schedule-averaged decisions under RR.

Table 13: RR top-three replacement decisions using original visit contributions and schedule-averaged loss changes.

Model	Precision@3 ↑	Norm. regret ↓	Spearman ↑	Sign ↑
BERT-tiny	1.000	0.0000	0.9860	0.917
BERT-base	1.000	0.0000	0.9860	1.000

F.5 Controlled scale panel

This subsection supports the model-size comparison summarized in Section 6.5.

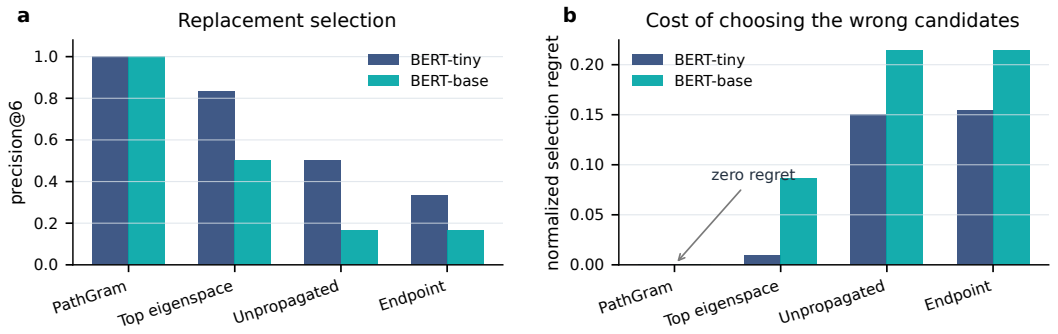


Figure 7: Replacement selection from signed-loss predictions. (a) Precision at the requested set size. (b) Normalized regret as defined in Section 6. The complete path and its orthogonal complement are both needed to preserve the oracle decision on the larger BERT model.

We use a separate BERT panel to compare backbone sizes under matched adaptation settings. Table 14 holds the data task, LoRA rank, adapted layer type, initialization rule and gradient-flow time fixed. Each model adapts its final attention block. The panel uses its own candidate draw, with median absolute loss signal 2.22×10^{-3} for the BERT-tiny comparison, compared with 1.21×10^{-3} in the primary full-batch study. Its errors vary nonmonotonically with model size, motivating the path-based interpretation below.

Table 14: Controlled scale panel with a separate candidate draw and matched adaptation settings. Parameter counts include the fixed backbone.

Model	Total parameters	Trainable dim.	η	Steps	Eval. acc.	NMAE	Sign ↑
BERT-tiny	4,388,226	2,306	0.0100	80	0.820	0.112	1.000
BERT-mini	11,175,170	4,610	0.0050	160	0.766	0.136	0.750
BERT-medium	41,382,402	9,218	0.0050	160	0.836	0.207	1.000
BERT-base	109,496,066	13,826	0.0025	320	0.734	0.074	1.000

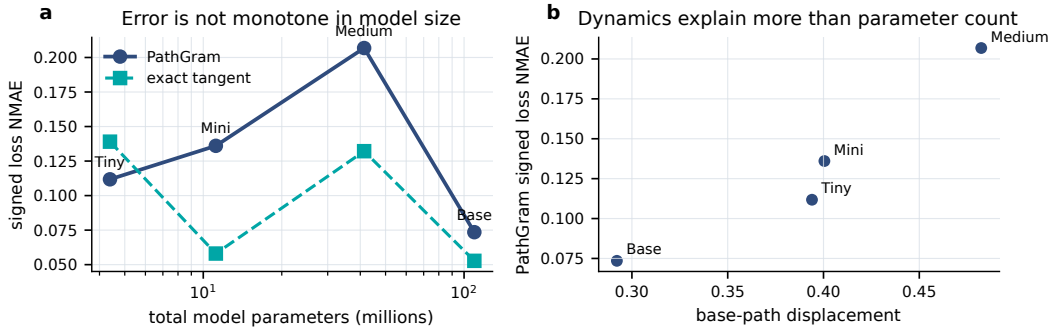


Figure 8: Controlled BERT scale comparison. All models use the same data task and the same type of final-attention LoRA adaptation with matched gradient-flow time. Prediction error varies nonmonotonically with total parameter count under matched adaptation locations and gradient-flow time.

Prediction error varies nonmonotonically with model size. To separate backbone size from adaptation choices, we compare BERT scales with matched data and LoRA settings at the same gradient-flow time. This comparison uses its own candidate draws, as specified in Table 14 and Figure 8. Within this panel NMAE rises to 0.207 on BERT-medium and then falls to 0.074 on BERT-base. Proposition 6 identifies curvature mismatch acting on accumulated responses as the source of propagation error. That quantity depends on the path and reference eigenspace as well as the model. Parameter count is therefore an incomplete summary of approximation quality in this comparison.

F.6 AMORTIZATION

This subsection supports the cost comparison in Section 6.6.

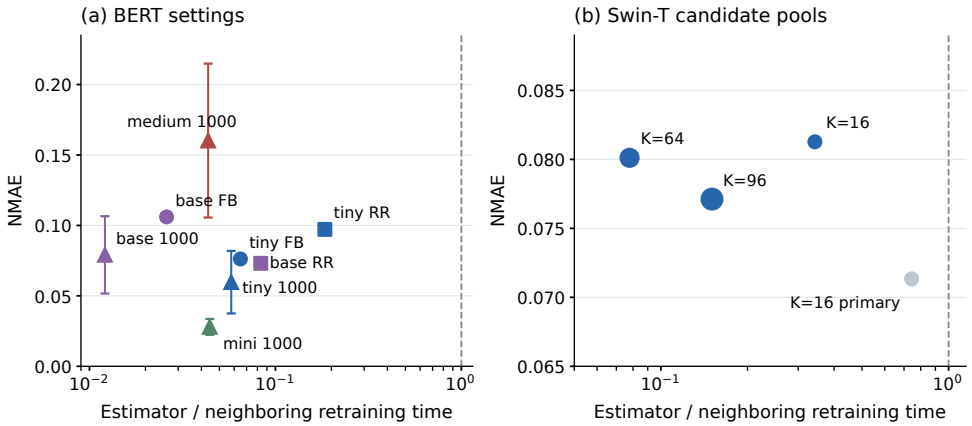


Figure 9: Amortized computation across primary and enlarged settings. Horizontal position is estimator time relative to neighboring retraining and vertical position is signed-loss NMAE. (a) BERT full-batch (circles), RR (squares) and $n = 1{,}000$ (triangles) settings. Triangles show means over three seeds, with vertical bars giving population standard deviations of NMAE. (b) Swin-T candidate pools, with marker area increasing with candidate count K. Gray marks the primary pool and blue the enlarged pool. The dashed line is equal cost.

Figure 9 places the unrefined primary and enlarged experiments on a common accuracy–cost display. Each marker uses matched candidate pools for prediction and retraining, so movement along the horizontal axis reflects their measured relative cost. Panel (a) separates the BERT protocols and their seed variation. In panel (b), the candidate-count encoding shows where shared preparation is amortized across more replacements. Refinement timings use their separate repeated protocol in Table 30.

F.7 POSITIVE-CURVATURE INTERVALS

This subsection supports the positive-curvature evaluation in Section 6.7.

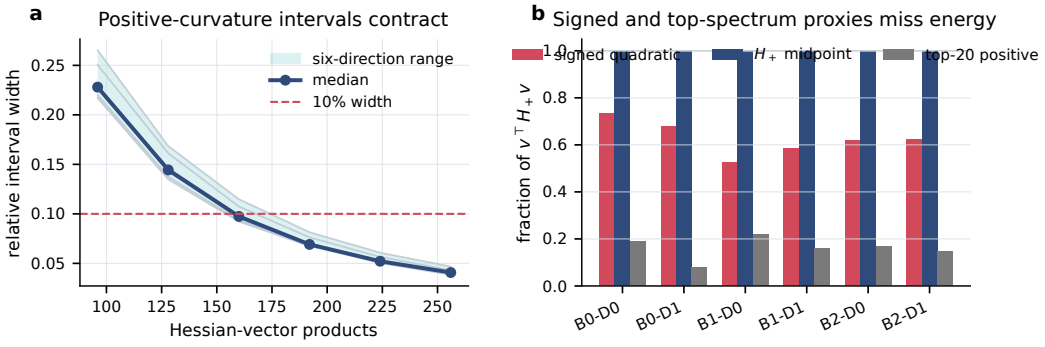


Figure 10: Matrix-free evaluation of $v^\top H_+ v$ on GroupNorm ResNet-20. (a) Relative interval width contracts with the HVP budget. Shading spans six directions from three independent base models, and the solid curve shows their median. (b) The signed quadratic and the leading-20 positive eigenspace differ materially from the full positive-curvature energy.

Positive-curvature intervals narrow with the HVP budget. To evaluate the metric in Theorem 8, we measure interval width as the HVP budget increases. Table 9 and Figure 10 report the width distribution across directions at each budget. All six ResNet directions reach relative width below 10% by 192 HVPs. Intersecting the intervals from successive budgets gives nested intervals for every

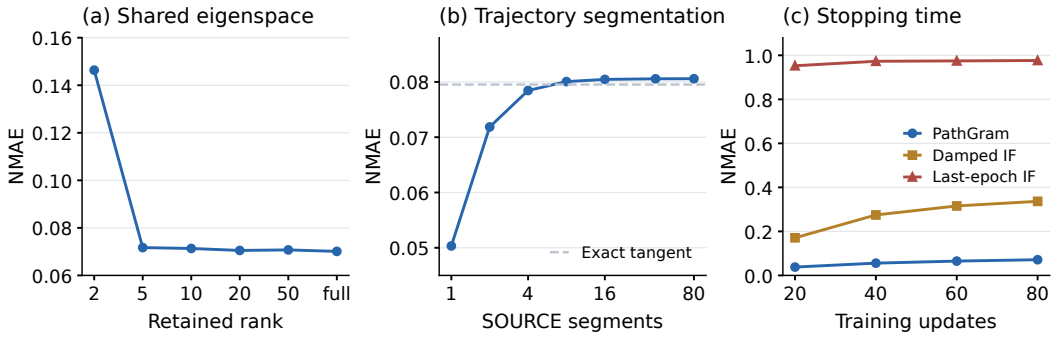


Figure 11: **Swin-T sensitivity comparisons.** (a) Retained spectral rank, including the full terminal spectrum. (b) SOURCE segment count, with the exact discrete tangent as a reference. (c) Stopping time for PATHGRAM, damped IF (factor one) and last-epoch IF. Every panel shows quadratic-readout NMAE. Panels vary one factor at a time under the conditions of Tables 16, 22 and 23, respectively.

direction. Each additional budget reuses the earlier products and refines the same positive-curvature target, as described in Section 5.4. The observed narrowing supports choosing the computation budget through the requested interval width.

The same figure compares the positive-curvature metric with simpler spectral summaries. The signed quadratic (Pearlmutter, 1994) differs from the positive-curvature energy because negative and positive directions can cancel. The leading 20 positive eigenmodes (Higham, 2008) capture 8.1%–22.0% of the 256-HVP midpoint energy across the six directions, with median 16.4%. The remaining positive directions therefore contribute most of the measured energy. The interval computation therefore evaluates the complete positive-curvature metric required by FHSD. This component comparison verifies the positive-curvature calculation used to evaluate empirical FHSD.

We also evaluate empirical positive-curvature energy for 24 approximate replacement responses on each full-batch BERT model. Table 15 reports projected numerical interval endpoints for the candidate mean $\widehat{D}_T((H_T)_+)$. The responses retain the scaling n of Section 3, and the energy uses that scaling directly. The reported widths measure quadrature precision conditional on the fixed candidate responses and implemented projected operator. The numerical intervals use the unpadded endpoints, with transfer to the exact Hessian governed by the operator-perturbation bound in Appendix B.9.

Table 15: Projected numerical intervals for empirical BERT positive-curvature energy. Each row averages 24 fixed approximate responses and uses 768 HVPs in total. Endpoints are rounded outward, and relative widths are calculated before rounding.

Setting	Lower endpoint	Upper endpoint	Relative width (%)
BERT-tiny FB	4.2894	4.2914	0.044
BERT-base FB	7.2862	7.2897	0.046

F.8 TERMINAL READOUT COMPARISON

This subsection supports the linear and quadratic readout comparison in Section 6.3.

To isolate the terminal readout, we compare linear and quadratic expansions on the same full-batch text responses. The quadratic term lowers NMAE from 0.1303 to 0.0762 on BERT-tiny and from 0.2507 to 0.1060 on BERT-base. The exact path tangent with quadratic readout gives NMAEs 0.0533 and 0.0723. Applying that readout to the measured neighboring displacement gives errors below 0.0007 (Table 18). This comparison motivates separating propagation and finite-replacement discrepancies from the terminal expansion in Appendix H.3.

Shared spectral rank. Table 16 varies the retained terminal spectral rank on the Swin-T task while holding the candidates, the readout and the initialization fixed. Figure 11a shows the marked error reduction between ranks two and five, followed by a nearly flat curve. Oracle selection is unchanged from rank five through the full spectrum. The full row retains every terminal eigendirection and still propagates fixed terminal curvature.

Table 16: Swin-T replacement NMAE and selection as the retained terminal spectral rank varies. Rank correlation and precision@6 refer to the quadratic readout. The full column retains every eigendirection of the terminal Hessian.

Rank	Linear NMAE	Quadratic NMAE	ρ	Prec
2	0.2926	0.1463	0.9912	0.8333
5	0.2778	0.0717	1.0000	1.0000
10	0.2773	0.0713	1.0000	1.0000
20	0.2765	0.0705	1.0000	1.0000
50	0.2767	0.0707	1.0000	1.0000
full	0.2760	0.0701	1.0000	1.0000

Curvature-rank sensitivity at larger step sizes. A separate BERT-base panel examines rank sensitivity for step sizes with $\eta\lambda_{\max} > 1$. We adapt the final attention query and value projections with rank-four LoRA and pair $n \in \{64, 96\}$ with $\eta \in \{0.02, 0.05\}$ for 40 updates and four replacements per configuration. The trainable dimension is 13,826, and the four values of $\eta\lambda_{\max}$ are 1.241, 1.353, 1.982 and 2.792. Table 17 reports cost and ordering across the retained ranks. The median estimator-to-retraining cost ratio grows from 0.273 at rank 2 to 3.11 at rank 48. Whole-panel execution time grows from 227 to 511 seconds, with reported peak memory of 678 MiB throughout. The median within-configuration rank correlation stays at 0.20 at every rank, while individual configurations range from -1.00 to 1.00. At the larger step size, shared-response and exact-tangent NMAEs reach 5.7×10^{17} and 3.9×10^{5}, respectively. Within this panel, retaining more curvature directions raises cost without improving median within-configuration ordering. The Swin-T comparison above evaluates rank selection on the primary prediction task.

Table 17: Retained curvature rank on the BERT panel whose step sizes lie outside the range $\eta\lambda \leq 1$. The cost ratio is the median over the four configurations, the wall-clock time covers the whole panel, and rank correlation is pooled over all sixteen directions.

Rank	Cost ratio	Wall time (s)	ρ	Sign	Peak memory (MiB)
2	0.2729	227.49	0.5676	0.8750	678
6	0.3855	234.87	0.5676	0.8750	678
12	0.7058	274.25	0.4853	0.8750	678
24	1.4511	344.00	0.4853	0.8750	678
48	3.1131	510.61	0.4941	0.9375	678

Response error decomposition. Table 18 separates shared-propagation error from path remainder on the three recorded settings. Write e_{prop} for the median over candidates of $\left\|\bar{g}_T^{(i)} - g_T^{(i)}\right\| / \left\|g_T^{(i)}\right\|$ and e_{path} for the median of $\left\|g_T^{(i)} - z_T^{(i)}\right\| / \left\|z_T^{(i)}\right\|$. The last column applies the quadratic terminal readout of (4) to the measured displacement $z_T^{(i)}$ itself, so it isolates the readout from any response approximation. These medians measure relative parameter discrepancies, complementing the mean squared remainder R_{rem} in Theorem 7.

Table 18: Median relative response errors and the loss error of the measured displacement under the same quadratic readout. Lower values indicate closer agreement.

Setting	e_{prop}	e_{path}	Measured-displacement NMAE
BERT-tiny	0.0887	0.0441	0.00028
BERT-base	0.0706	0.0473	0.00064
Swin-T	0.0177	0.0203	0.00012

F.9 Moment aggregation on Fashion-MNIST and cross-step energies on Swin-T

This subsection supports the temporal-aggregation comparison in Section 6.2.

We compare aggregation rules in a common terminal top-20 eigenspace on ten independently trained Fashion-MNIST models. For each candidate, the full moment propagates base-path replacement

Table 19: **Retaining visit interactions improves squared-response estimates.** The upper block compares BERT models under RR. The energy ratio divides the mean squared linear prediction by its diagonal-time estimate. The NMAE columns compare squared quadratic predictions with squared retraining loss changes after averaging over two schedules. The lower block summarizes ten independent Fashion-MNIST base models. Multiplicative errors are medians and are better closer to unity. Spearman correlation measures ordering across base models. Arrows indicate the preferred direction for error and correlation.

RR model	Energy ratio	Original NMAE ↓	Random-sign NMAE ↓
BERT-tiny	about 25×	0.161	0.960
BERT-base	about 17×	0.129	0.947
Fashion-MNIST estimate	**Mult. error**	**Spearman ↑**	
Full second moment	1.12×	0.939	
Same-path peak	1.33×	0.964	
Separated peak	12.92×	0.661	

forcing using the fixed terminal eigenvalues and then evaluates its positive-curvature energy. The measured target is $\sum_j \lambda_j^+ [n u_j^\top (w_T^{(i,1)} - w_T)]^2$, using the fully retrained neighboring displacement and the same terminal eigenpairs (u_j, λ_j).

The two peak capacities summarize forcing magnitudes within this eigenspace. Let $\xi_{t,j}$ be the base-path forcing in mode j and let $r_j = \eta \sum_{t=0}^{T-1} |1 - \eta\lambda_j|^t$ be its absolute propagation gain. The same-path peak capacity is $\sum_j \lambda_j^+ [r_j \max_t |\xi_{t,j}|]^2$. For the separated reference, we measure the direct forcing $f_{t,j}$ and curvature-mismatch residual $e_{t,j}$ in the scaled neighboring-displacement recursion. Its capacity is $\sum_j \lambda_j^+ [r_j(\max_t |f_{t,j}| + \max_t |e_{t,j}|)]^2$. Each base model contributes one recorded replacement. We report median multiplicative errors and rank correlations across the ten base models.

The full moment has median multiplicative error 1.12, compared with 1.33 for the same-path peak and 12.92 for the separated peak (Table 19). It is closer to measured energy on every base model. The same-path peak has higher rank correlation, separating agreement in ordering from agreement in magnitude. Preserving the temporal forcing sequence therefore improves energy-magnitude prediction over these peak summaries.

Cross-step energies on the recorded vision path. Every candidate appears in every full batch of the Swin-T path, so its visit count equals the 80 updates. Write $c_{is} = \frac{\eta}{n} \nabla L_Q(w_T)^\top \Phi_{T,s+1} \xi_s^{(i)}$ for the projected contribution of visit s to candidate i. The mean of $(\sum_s c_{is})^2$ over candidates is 6.7366×10^{-7} and the mean of $\sum_s c_{is}^2$ is 1.0166×10^{-8}, a ratio of 66.2677 with interval $[53.91, 71.64]$. Cross-step pairs therefore carry 98.5% of the mean squared linear readout, and that ratio reaches 82.8% of the visit-count bound in Proposition 3. These energies use the evaluation-gradient metric $M_T = \nabla L_Q(w_T) \nabla L_Q(w_T)^\top$, linking the visit-pair decomposition to the linear loss prediction.

F.10 SWIN COMPARISON WITH EXTERNAL INFLUENCE METHODS

We compare external influence methods on the Swin-T task with $n = 96$ and the same 16 replacement candidates. All predictions target the signed evaluation-loss change under matched initialization and data order. The comparison uses data seed 731 and initialization seed 2 with 80 updates at step size 0.06. The adapter rank is 2 and the PATHGRAM spectral rank is 10. We select six candidates in this comparison and evaluate precision@6 and normalized regret against the same retraining outcomes.

We implement damped influence functions (Koh & Liang, 2017) using exact Hessian–vector products and conjugate gradients with relative tolerance 10^{-10}. The damping factors are 0.1, 1 and 10 times $1/(\eta T)$. Positive-definiteness adjustment gives actual damping values 0.1422, 0.2083 and 2.0833. TracIn uses all 80 pre-update checkpoints with weights η/n (Pruthi et al., 2020). Last-epoch SGD-influence restricts propagation to the final epoch (Hara et al., 2019). SOURCE uses four equal segments and averages exact Hessian actions and replacement gradients at each segment's endpoints

before exponential propagation (Bae et al., 2024). Table 20 separates these response approximations from the linear or quadratic terminal readout.

Table 20: Swin-T replacement NMAE with linear and matched quadratic readouts. Lower values indicate closer agreement with the same neighboring retraining outcomes.

Method	Linear NMAE	Quadratic NMAE
PATHGRAM	0.2773	0.0713
SOURCE, four segments	0.2772	0.0784
Last-epoch SGD-influence	0.9766	0.9765
Damped IF, factor 0.1	0.5053	0.7930
Damped IF, factor 1	0.3936	0.3366
Damped IF, factor 10	0.8503	0.8465

The readout comparison in Table 20 extends Table 2 across damping choices. For both path-covering estimators, the terminal quadratic term supplies the principal magnitude improvement while retaining exact oracle-set recovery. TracIn's checkpoint-similarity score and the IF responses are evaluated against the same neighboring outcomes.

Rank correlation and paired differences. Table 21 compares their absolute prediction errors with quadratic PATHGRAM on matched candidates. TracIn retains its checkpoint-similarity score, and the IF entries use the quadratic terminal readout. All four mean paired differences are positive, with intervals above zero, indicating larger magnitude error for these external settings on the recorded Swin-T path.

Table 21: Swin-T paired NMAE differences against quadratic PATHGRAM on the same 16 replacements. TracIn uses checkpoint similarities and IF uses quadratic readout. Positive values indicate larger error than PATHGRAM. Brackets give percentile intervals from resampling matched candidate pairs.

Method	ΔNMAE	Interval
TracIn	0.3310	$[0.1830,\ 0.5205]$
Damped IF, factor 0.1	0.7217	$[0.3701,\ 1.4128]$
Damped IF, factor 1	0.2652	$[0.2212,\ 0.3172]$
Damped IF, factor 10	0.7752	$[0.7119,\ 0.8138]$

Segment count. Table 22 varies the number of stationary segments while holding the candidates, the terminal readout and the initialization fixed. Figure 11b displays the complete sweep and its approach to a plateau. The exact discrete tangent provides a separate reference for the endpoint of this comparison. The segment sweep uses exponential propagation, explaining its distinct eighty-segment value. At the four-segment setting the paired difference between the two path-covering estimators is 0.0071 with interval $[0.0036, 0.0113]$. Intervals reported here and below quantify candidate sensitivity through paired resampling conditional on the recorded path and stated readouts.

Table 22: Swin-T replacement NMAE as the number of stationary segments varies. All rows share the candidates, initialization and terminal readout of Table 20. The last row propagates the exact discrete tangent instead of a segment exponential.

Segments	Linear NMAE	Quadratic NMAE
1	0.2650	0.0503
2	0.2748	0.0719
4	0.2772	0.0784
8	0.2777	0.0801
16	0.2779	0.0805
40	0.2779	0.0806
80	0.2779	0.0806
Exact discrete tangent	0.2777	0.0795

Shorter horizons. Table 23 stops the recorded path at four points and rebuilds the evaluation gradient, the evaluation Hessian and every neighboring retraining outcome at each one. Influence damping follows the factor-one rule above, which makes it depend on the stopping point. Quadratic PATHGRAM NMAE ranges from 0.0379 to 0.0713 across the four stopping points, supporting prediction at multiple stages of the recorded path. Its linear error at twenty updates is 0.3235 against 0.3220 for damped influence, while its quadratic error stays below every alternative at all four stopping points.

Table 23: Swin-T replacement NMAE at four stopping points of the recorded path. Evaluation gradients, evaluation Hessians and neighboring retraining outcomes are recomputed at each stopping point.

Method	20 updates	40 updates	60 updates	80 updates
PATHGRAM, linear	0.3235	0.3355	0.3055	0.2773
PATHGRAM, quadratic	0.0379	0.0558	0.0651	0.0713
Damped IF factor 1, linear	0.3220	0.3743	0.3908	0.3936
Damped IF factor 1, quadratic	0.1709	0.2747	0.3157	0.3366
Last-epoch, linear	0.9529	0.9732	0.9747	0.9766
Last-epoch, quadratic	0.9525	0.9729	0.9746	0.9765

F.11 SCHEDULE-SPECIFIC BASE LOSSES AND RR SELECTIONS

We compare the base evaluation losses for full-batch training and each realized RR schedule in Table 24. The losses differ within each model despite equal classification accuracies on the 128 evaluation examples. Classification accuracy counts discrete decisions and can remain unchanged when predicted probabilities change. These measurements accompany the independently trained schedule-specific base models used for the replacement comparisons. Candidate predictions and retraining loss changes are averaged over the two schedules before selecting the three smallest values.

Table 24: Evaluation losses and accuracies of separately trained full-batch and RR base models.

Model	Schedule	Evaluation loss	Evaluation accuracy
BERT-tiny	Full batch	0.5161	0.8203
BERT-tiny	RR 701	0.5158	0.8203
BERT-tiny	RR 1701	0.5160	0.8203
BERT-base	Full batch	0.6009	0.7344
BERT-base	RR 701	0.6010	0.7344
BERT-base	RR 1701	0.6010	0.7344

For BERT-tiny the predicted ordering selects candidates $(6, 3, 0)$ and the retraining oracle orders them as $(6, 0, 3)$. For BERT-base both orderings select $(6, 11, 10)$. Each model therefore has precision@3 equal to one and zero normalized regret, as reported in Table 13. Set recovery concerns membership rather than the internal order of the selected candidates. Each candidate's prediction averages its two schedule-specific evaluations.

G SCOPE OF REPLACEMENT PREDICTION

This appendix relates the query complexity in Appendix D to the measured prediction variation in Section 6.5.

The bounded smooth family in Appendix D shares the base gradient-descent path and all finite-order derivatives near it while encoding m signs along neighboring paths, with their parity determining a bounded terminal loss that cannot be recovered from the shared base-path information alone. With fewer than m off-path finite-order queries, any adaptive algorithm has worst-case absolute error at least $1/4$. This includes randomized algorithms. Matching gradient queries at public points recover the loss. The construction concerns unrestricted smooth risks, with observable low-dimensional responses or controlled displacement providing additional structure for practical prediction.

On the primary full-batch text settings, the terminal quadratic term removes 42% and 58% of the linear error. The variation on enlarged settings is reported in Section 6.5.

Across the enlarged BERT runs, the population standard deviations of NMAE are 0.022 around 0.060 for BERT-tiny and 0.027 around 0.079 for BERT-base. BERT-medium varies by 0.055 around 0.160. For BERT-base, the run with median absolute loss change 5.7×10^{-5} has NMAE 0.107. Reporting both the signal and normalized error distinguishes the prediction scale from its relative accuracy. Equation (6) connects expected replacement loss changes to generalization under the replace-one coupling. FHSD conditions on a realized training path and resolves the visits and loss directions that contribute to its replacement sensitivity.

H Finite-Replacement Refinement and Error Measurements

This appendix develops the four-checkpoint refinement and the additional measurements in Section 6.4. We first specify the correction, then give its experimental settings and explain the signed error decomposition.

H.1 Gradient residuals along an approximate neighboring path

Consider full-batch gradient descent on the original regularized objective F and its one-example replacement $F^{(i)}$. Write $d_t = \bar{g}_t^{(i)}/n$ for the physical PathGram displacement and $x_t = w_t + d_t$ for the approximate neighboring path. The gradient residual measures its departure from an update on $F^{(i)}$:

$$R_t = \nabla F^{(i)}(x_t) + \frac{x_{t+1} - x_t}{\eta_t} = \nabla F^{(i)}(w_t + d_t) - \nabla F(w_t) + \frac{d_{t+1} - d_t}{\eta_t}. \tag{16}$$

The original gradient is available from $(w_t - w_{t+1})/\eta_t$. The first PathGram step equals the replacement gradient difference, so $R_0 = 0$. For the 80-update experiments, we evaluate R_t at $t \in \{20, 40, 59, 79\}$ and interpolate linearly between these points and $(0, 0)$. With $\widetilde{R}_t$ denoting that interpolation, we compute one correction sweep:

$$c_0 = 0, \qquad c_{t+1} = (I - \eta_t \bar{G}) c_t - \eta_t \widetilde{R}_t, \qquad d_T^{\mathrm{M4}} = d_T + c_T. \tag{17}$$

We use the same leading eigenspace and retain the orthogonal response in both recursions. The four gradient locations are determined by the original approximate path before this sweep.

This is an application of residual correction to a discrete training path, following the numerical principle of correcting an approximate trajectory through its update discrepancy (Dutt et al., 2000). To see the connection, let $\delta_t = w_t^{(i,1)} - w_t$ be the actual replacement displacement and set $e_t = \delta_t - d_t$. Subtracting the approximate-path update from the actual update gives

$$e_{t+1} = e_t - \eta_t \big[\nabla F^{(i)}(x_t + e_t) - \nabla F^{(i)}(x_t)\big] - \eta_t R_t. \tag{18}$$

Equation (17) approximates the gradient difference in this relation using the shared curvature and the residual using four temporal samples. The residual includes the shared-curvature discrepancy and the response of the replaced objective to a finite parameter displacement.

For the refinement comparisons, every response method uses

$$f(d) = L_Q(w_T + d) - L_Q(w_T) \tag{19}$$

as its terminal readout. M4 predicts $f(d_T^{\mathrm{M4}})$, while the unrefined response predicts $f(d_T)$. This common evaluation isolates the response correction from the choice of terminal readout. The primary experiments in Table 1 retain the quadratic readout of Eq. (4).

H.2 Configurations, comparisons and independent runs

Both refinement tasks use 96 balanced training examples, 16 independently evaluated replacement candidates and 80 full-batch gradient-descent updates. Each candidate replaces one example with a held-out example of the other class. All replacement models start from the same initialization as their corresponding base model.

For vision, we use the same Swin-T feature task and rank-two tanh adapter described in Section 6.1. The adapter and head have 4,610 trainable parameters, and the query set contains 256 images. We cross each initialization/candidate seed with step sizes 0.03, 0.06 and 0.12. The data split remains fixed across these runs. Seeds 310–312 select the four-checkpoint budget and supply the component comparisons below. Seeds 313–315 independently measure the signed errors of the unrefined estimators. Seeds 410–412 evaluate the fixed correction, producing nine confirmation runs and 144 full neighboring retraining outcomes.

For text, we use BERT-tiny on baseball versus space, with rank-four LoRA on the final attention query and value projections. The trainable head and LoRA parameters total 2,306 parameters. We use a fixed split with 96 training documents, 20 replacement-pool documents and 120 query documents. Initialization seeds 10510–10512 pair with candidate seeds 510–512. The step size is 0.01, giving three confirmation runs and 48 neighboring retraining outcomes. We transfer the four checkpoint times and the single correction sweep without retuning. Both tasks use weight decay 0.001, spectral rank $q = 10$ and double precision. Text runs use an RTX 3090, with training and query microbatches of 16 and 32, respectively.

We compare M4 with PATHGRAM, one- and four-segment SOURCE-style propagation, and the exact discrete tangent. The segment implementation uses endpoint-averaged Hessian actions and replacement gradients, as in Appendix F.10. Two short-replay comparisons perform four gradient-descent updates on the replaced objective, starting at time $T - 4$. One starts from the base checkpoint, and the other starts from that checkpoint plus the PATHGRAM displacement. The latter shares the same response preparation as M4 and uses the same number of additional full gradients.

We implement a same-size replacement version of DeltaGrad (Wu et al., 2020) using cached original gradients and Hessian BFGS actions. Exact base-objective gradients are evaluated through update index 20 and then every second update. At other steps, two retained curvature pairs approximate the base-gradient change, and the new-minus-old example gradient supplies the replacement term. The refresh period is selected from $\{2, 5, 10\}$ on seeds 310–312, with all other settings fixed before confirmation. The BFGS scale has lower bound 0.001, and the nonconvex fallback uses threshold $L = 1$.

Each NMAE is computed within a run and then averaged equally across runs. The CPU confirmation study has three independent initialization/candidate seeds, each observed at three step sizes. Table 25 gives all response comparisons, and Table 26 gives seed-level variation.

Table 25: **Complete refinement comparison.** Extension of Table 3 to SOURCE-style responses and the exact tangent. Mean NMAE on new confirmation seeds, with actual terminal loss evaluation for every method. Swin-T uses three seeds at three step sizes, and BERT-tiny uses three seeds at one step size.

Method	Swin-T features	BERT-tiny LoRA
PATHGRAM response	0.1034	0.0741
SOURCE-style, 1 segment	0.0977	0.0967
SOURCE-style, 4 segments	0.1128	0.0584
Exact discrete tangent	0.1134	0.0608
PATHGRAM + last-four-step replay	0.0945	0.0700
Four-checkpoint correction (M4)	**0.0027**	**0.0023**

Table 26: Per-seed refinement NMAE. Replay starts from the PATHGRAM-shifted checkpoint. Each Swin-T entry averages three step sizes, and BERT-tiny uses one. Within each task, seeds share the data split.

Task and candidate seed	PATHGRAM response	Last-four-step replay	M4
Swin-T, 410	0.0857	0.0780	0.0024
Swin-T, 411	0.1204	0.1103	0.0033
Swin-T, 412	0.1041	0.0952	0.0024
BERT-tiny, 510	0.0728	0.0678	0.0023
BERT-tiny, 511	0.0812	0.0766	0.0021
BERT-tiny, 512	0.0683	0.0654	0.0024

H.3 SIGNED PREDICTION ERROR DECOMPOSITION

We measure loss discrepancies at the approximate response, exact tangent and fully retrained displacement. Let $v = g_T^{(i)}/n$ be the exact tangent displacement and let $q(d)$ be the quadratic loss change in Eq. (4). For an approximate displacement $\widehat{d}$, its signed error decomposes as

$$q(\widehat{d}) - \Delta_i L_Q = \underbrace{q(\widehat{d}) - q(v)}_{\text{propagation}} + \underbrace{q(v) - f(v)}_{\text{terminal expansion}} + \underbrace{f(v) - \Delta_i L_Q}_{\text{finite replacement}}. \tag{20}$$

We compute each absolute component with the same run-level NMAE denominator. The signed components sum to the total error before taking absolute values. Table 27 summarizes the nine configurations from error-measurement seeds 313–315, each evaluated at three step sizes.

Table 27: Normalized absolute loss discrepancies on independent Swin-T runs. Each entry averages three seeds and three step sizes. The final column measures the total prediction error after signed combination.

Response	Propagation	Terminal expansion	Finite replacement	Total NMAE
PATHGRAM	0.0125	0.00013	0.0824	0.0749
SOURCE-style, 1 segment	0.0522	0.00013	0.0824	0.0725

For each response, the propagation component opposes the exact tangent's total prediction error on 107 of 144 candidates. This 74.3% opposition identifies partial compensation between the propagation and tangent errors. The exact tangent with quadratic readout has mean NMAE 0.0824 on these runs. Figure 3 shows the variation across seeds and step sizes. The quantities here are loss discrepancies, while Table 18 reports relative parameter errors.

H.4 INFORMATION RETAINED BY THE CORRECTION

We separate the original objective's response from the feedback contributed by the changed example. Write $D_i = F^{(i)} - F$. Along the PATHGRAM path, the residual in Eq. (16) equals $B_t + J_t$, where

$$\begin{aligned} B_t &= \nabla F(w_t + d_t) - \nabla F(w_t) - \bar{G} d_t, \\ J_t &= \nabla D_i(w_t + d_t) - \nabla D_i(w_t). \end{aligned} \tag{21}$$

The linear approximation to the first component is $B_t^{\text{lin}} = (\nabla^2 F(w_t) - \bar{G}) d_t$. We apply the same interpolation and propagation separately to each retained combination, then evaluate the actual terminal loss. Table 28 uses seeds 310–312 at the three development step sizes. Parameter error divides the mean displacement-error norm by the mean retrained-displacement norm within each run.

Table 28: Correction components on the development runs. All corrections use the same four checkpoints. Both metrics are averaged equally across nine runs.

Retained residual information	Query NMAE	Parameter error
Unrefined PATHGRAM	0.0720	0.0327
Original-objective response B	0.0918	0.0219
Changed-example feedback J	0.0208	0.0251
Linear response and feedback $B^{\text{lin}} + J$	0.0097	0.0031
Complete residual $B + J$	0.0023	0.0027

The complete residual gives lower query and parameter errors than either individual component on all nine runs. The query changes induced by the two individual components have opposite signs on 137 of 144 candidates. Their joint effect therefore explains the correction more directly than either component's isolated change in query loss.

H.5 SINGLE-REPLACEMENT SELECTION

For the confirmation tasks, we also choose the single candidate with the smallest predicted loss change, including a no-change option with value zero. Let a be that action and a^* the minimizer of the measured retraining changes over the same options. We evaluate the regret $\Delta_a L_Q - \Delta_{a^*} L_Q$.

This selects one actual intervention from each candidate set. Table 29 reports the number of runs recovering an optimal action and the candidate-level sign errors.

Table 29: Single-replacement selection with actual terminal loss readout. Optimal choices are counted across nine Swin-T runs or three BERT-tiny runs. Sign errors are counted over 144 and 48 candidates, respectively.

	Swin-T features		BERT-tiny LoRA	
Method	Optimal	Sign errors	Optimal	Sign errors
PATHGRAM response	9/9	1	3/3	2
SOURCE-style, 1 segment	9/9	2	3/3	1
SOURCE-style, 4 segments	9/9	2	3/3	2
Exact discrete tangent	9/9	3	3/3	2
PATHGRAM + last-four-step replay	9/9	1	3/3	2
Base + last-four-step replay	6/9	21	1/3	15
Four-checkpoint correction (M4)	9/9	0	3/3	0

Both the unrefined response and M4 recover the optimal action in every confirmation run, with zero regret. The refinement improves magnitude agreement and eliminates the observed sign errors while preserving these choices. The primary top-k comparisons retain their separate definition in Section 6.1.

H.6 REPEATED EXECUTION TIMES

We measure CPU execution times with the data, model and original training path already in memory. Each method recomputes its required candidate gradients, curvature information, propagation and terminal loss evaluation. For M4, this includes the original response preparation and all four replacement-objective gradients per candidate. Each method receives one warm-up and five measured repetitions per configuration, in randomized interleaved order.

Table 30: CPU time per 16 candidates. The first block gives the median of the nine run-level median times. The second block is a separate interleaved Swin-T comparison and reports medians of 45 measurements per method (five repetitions at each of nine configurations). NMAE averages the corresponding nine confirmation runs.

Method	Seconds	NMAE
PATHGRAM response	0.1835	0.1034
Four-checkpoint correction (M4)	0.2641	0.0027
PATHGRAM + last-four-step replay	0.2231	0.0945
Complete neighboring retraining	0.8189	0
M4, DeltaGrad comparison	0.2619	0.0027
DeltaGrad adaptation	1.4252	0.0012
Complete neighboring retraining, DeltaGrad comparison	0.8834	0

In the first comparison, we calculate the retraining-to-M4 time ratio within each run before taking the median, obtaining $3.12\times$. M4 is faster than full neighboring retraining in all nine runs. The separate Swin-T comparison reports both time and NMAE for M4 and DeltaGrad in Table 30. M4 is 5.44 times faster, with NMAE 0.0027 compared with 0.0012 for DeltaGrad. Both recover the optimal action in all nine Swin-T runs and have zero sign errors across the 144 candidates. The base-checkpoint four-step replay, executed without response or eigenspace preparation, takes 0.0461 seconds in a separate repeated measurement. Its mean NMAE is 0.8712. All these costs exclude the common base training, feature extraction and data loading.