How One Training Example Changes Others: Exact Route Attribution for BatchNorm

Anonymous authors
Paper under double-blind review

Abstract

In models with batch normalization, an example affects learning through both its own loss and the shared statistics used by other examples. Training-data attribution measures how individual examples influence a model, but a total influence score does not explain how that influence arises. We introduce Edge-Game Replay (EGR) to separate these contributions as they propagate through training. EGR intervenes on loss participation, normalization statistics, later batch membership, and running-buffer updates, and uses exact Shapley values over complete training replays to allocate the final evaluation-loss difference. We find that shared statistics and loss participation make contributions of comparable magnitude. Across 16 CIFAR-10 ResNet runs, adding statistic contributions to the loss contribution reduces mean absolute prediction error by 66.9% after 393 updates and raises Kendall correlation from 0.391 to 0.822 on a disjoint evaluation set. The resulting score ranks deletions more accurately than TracInCP, last-layer influence functions, and single-model TRAK at all three comparison horizons. Eight CIFAR-100 WideResNet runs also show improved two-update prediction. The decomposition also guides the choice among loss masking, exclusion from normalization statistics, and example deletion.

1 Introduction

Training-data attribution estimates how individual training examples affect model predictions, helping identify which data to inspect or remove (Koh & Liang, 2017; Pruthi et al., 2020; Bae et al., 2024; Deng et al., 2026). With batch normalization (BatchNorm), an example influences training through both its supervised loss and the mean and variance used to normalize its batch (Ioffe & Szegedy, 2015). Removing it changes the activations and gradients of the remaining examples, which we call its *peers*. These changes propagate through later updates and alter the final model.

Loss masking and deletion distinguish these two pathways: masking removes an example's supervised loss while retaining its contribution to batch statistics, whereas deletion removes both. Ghost samples exploit this distinction to provide normalization statistics without a supervised loss (Pfahler & Morik, 2021). The quantitative question is how much each pathway contributes to the final evaluation loss after its effects propagate through training. Contributions with opposite signs may cancel in the aggregate deletion effect, obscuring whether an example influences learning through supervision or shared statistics. Separating these pathways provides a way to analyze example influence and design selective interventions.

We introduce *Edge-Game Replay* (EGR), which decomposes deletion effects by intervening on *routes* through the training computation. The routes control loss participation, normalization statistics, later batch membership, and running-buffer updates. Later membership captures the change in batch composition when deletion precedes batching (Wu et al., 2023); running buffers carry normalization statistics into evaluation. EGR replays each combination of interventions through subsequent training and uses exact Shapley values to divide the final evaluation-loss difference among routes, including their interactions.

We find that shared statistics contribute at a comparable magnitude to loss participation across the original, merged, and layer-split route partitions. Their aggregate effect increases with the number of affected peers, even as the effect on each peer decreases with batch size.

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.

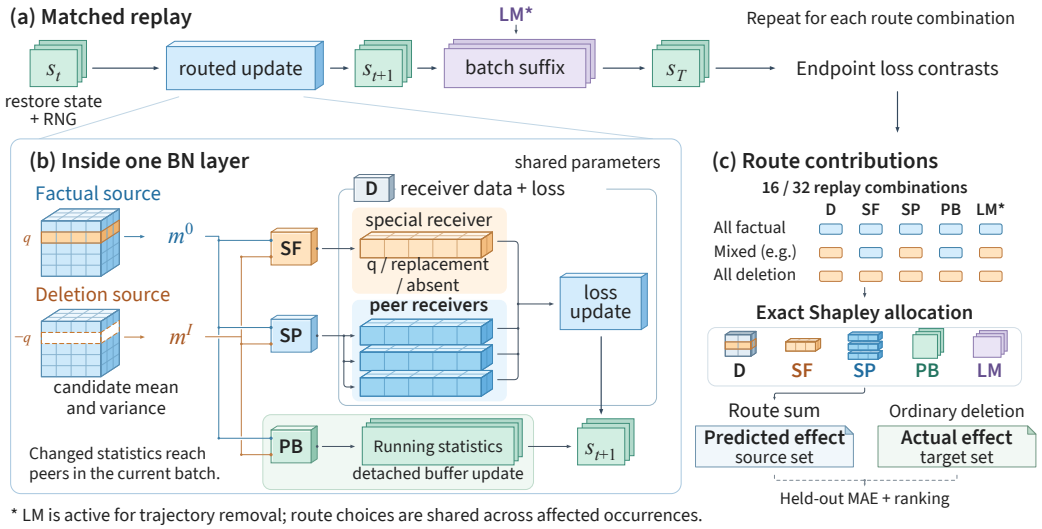


Figure 1: Overview of Edge-Game Replay. (a) Route interventions propagate through subsequent training updates. (b) Receiver and source streams separate loss participation from normalization statistics. (c) Shapley contributions allocate the source-set loss difference; their sums predict deletion effects on the target set.

The decomposition improves prediction of deletion effects and guides training interventions. Across 16 CIFAR-10 ResNet runs, adding statistic contributions improves prediction at every tested horizon: after 393 updates, mean absolute error decreases by 66.9% and Kendall correlation increases from 0.391 to 0.822 on a disjoint evaluation set. The resulting score outperforms TracInCP, last-layer influence functions, and single-model TRAK in deletion ordering. Results on eight CIFAR-100 WideResNet runs further demonstrate improved two-update prediction. In a noisy-image task with an exposure shift, an EGR-guided policy improves accuracy over ordinary training and high-loss filtering across 16 new models; restricting its actions to loss masking and buffer exclusion improves accuracy further.

2 Related Work

Batch composition and shared statistics. Cross-example dependence makes loss participation one of several ways an example affects learning. BatchNorm couples normalized activations across examples (Ioffe & Szegedy, 2015; Wang & Li, 2025); XK-FAC retains this dependence in curvature approximations (Lee et al., 2020), whereas DAVINZ avoids BatchNorm for neural tangent kernel (NTK)-based data valuation (Wu et al., 2022). ECIF separates positive-pair loss from an example's role as a negative for other pairs (Xie et al., 2026). Batch construction changes the optimization problem (Wu et al., 2023), and Reload studies how deletion changes retained examples' statistics and gradients (Newatia et al., 2026).

Path allocation and training-process explanations. Shapley Flow allocates prediction differences to graph edges (Wang et al., 2021). Exact Path Kernel represents a predictor along its gradient-descent path (Bell et al., 2023). ExPLAIND extends training-process decomposition to modern optimizers and attributes contributions across examples, components, and steps (Eichin et al., 2026). EGR instead allocates the endpoint evaluation-loss difference between factual and deletion replays to training routes.

Influence through training and deletion. Influence functions estimate local responses to data perturbations (Koh & Liang, 2017), while trajectory methods estimate influence along realized training trajectories (Hara et al., 2019; Chen et al., 2021; Pruthi et al., 2020; Bae et al., 2024; Wang et al., 2025b). TrackStar incorporates optimizer-state corrections (Chang et al., 2025). Coupled SGD runs support unlearning (Ullah et al., 2021), and trajectory-specific leave-one-out comparisons quantify

deletion effects (Wang et al., 2025b). Influence scores can lose predictive accuracy as training continues, even with matched randomness (Schioppa et al., 2023).

Data units and attribution targets. Datamodels predict outputs of models trained on subsets (Ilyas et al., 2022), while Data Shapley values individual examples (Ghorbani & Zou, 2019). f-INE compares outcome distributions under training randomness (Panda et al., 2026). AirRep models group influence, and ProxySPEX estimates higher-order interactions among examples (Sun et al., 2025; Butler et al., 2025). Concept-TRAK and MOTIVE target visual concepts and motion (Park et al., 2026; Wu et al., 2026); GUDA uses group unlearning for diffusion-model attribution (Murata et al., 2026). EGR instead decomposes one example's deletion effect into training-route contributions. Appendix M extends these comparisons.

3 EDGE-GAME REPLAY

3.1 DEFINING THE DELETION EFFECT

We measure influence after the intervention's effects propagate through subsequent training. Let q be the *focal example* to remove and $s_t = (\theta_t, o_t, m_t)$ the pre-intervention state (parameters, optimizer state, and BatchNorm running buffers). The replay ends at T. Write $J_E(s)$ for the mean loss on E at state s in evaluation mode.

We consider two removal rules, $I \in \{\text{once}, \text{trajectory}\}$. *Single-occurrence* removal deletes q from one update and keeps later batches fixed. *Trajectory* removal deletes q before batching in each replay epoch; the first example of the next batch fills the affected batch, and the remaining sequence is rebatched. To isolate the deletion effect, factual and deletion replays start from the same state s_t, share the realized random choices, and run to T. Their endpoint evaluation-loss difference defines the ordinary deletion benefit

$$H^{\text{ord}}_{q,I}(E) = J_E(s^0_T) - J_E(s^I_T), \tag{1}$$

where s^0_T and s^I_T are the factual and deletion terminal states. Positive benefit means that removal lowers loss. EGR computes contributions on a source evaluation set E_{src} and uses them to predict deletion benefit on a disjoint target set E_{tgt}. This tests whether the measured influence transfers to other evaluation examples. Both sets are distinct from the batches supplying normalization statistics.

3.2 SEPARATING LOSS RECEIVERS FROM STATISTIC SOURCES

Separating these pathways requires independent control over loss participation and normalization-statistic sources. Each BatchNorm layer therefore processes three activation streams sharing the current replay's parameters. The *receiver stream* contains the examples whose losses drive the update. Two *statistic-source streams*, formed from the factual and deletion batches, supply their respective means and variances. Table 1 defines the factual and deletion choices for each route.

A *coalition* S specifies the routes assigned deletion settings: $c_g(S) = I$ if $g \in S$, and $c_g(S) = 0$ otherwise. These settings remain fixed across affected occurrences. The SF route selects the statistic source for the focal receiver $r^\star(S)$. This receiver is q when D is factual. When D uses its deletion setting, the receiver is the replacement under trajectory deletion and is absent under single-occurrence deletion. At layer ℓ, source $a \in \{0, I\}$ supplies mean μ^a_ℓ and variance $(\sigma^a_\ell)^2$, with dependence on the coalition and replay step implicit. Receiver r is normalized by

$$a_{\ell r}(S) = \begin{cases} c_{SF}(S), & r = r^\star(S), \\ c_{SP}(S), & \text{otherwise}, \end{cases} \qquad h^R_{\ell r}(S) = \gamma_\ell \frac{z^R_{\ell r}(S) - \mu^{a_{\ell r}(S)}_\ell}{\sqrt{(\sigma^{a_{\ell r}(S)}_\ell)^2 + \epsilon}} + \beta_\ell. \tag{2}$$

Here $z^R_{\ell r}$ and $h^R_{\ell r}$ are receiver activations before and after normalization; γ_ℓ, β_ℓ are learned affine parameters, and $\epsilon > 0$ is the numerical-stability constant. Each source stream normalizes itself with its own moments before entering the next layer. The supervised loss is computed on receivers, with gradients flowing through their selected moments to the source activations. The PB route selects detached source moments for updating the running buffers. Appendix A gives the variance convention and complete stream construction.

Table 1: Each route selects a factual or deletion computation within a complete replay.

Route	Factual choice	Deletion choice
D	Factual receivers, targets, loss mask, and mean-loss divisor	Removal-rule receivers, targets, loss mask, and mean-loss divisor
SF	Factual-source moments for the focal or replacement receiver	Deletion-source moments for the focal or replacement receiver
SP	Factual-source moments for the peers	Deletion-source moments for the peers
LM	Factual later batches	Later batches after removal
PB	Factual-source moments for running-buffer updates	Deletion-source moments for running-buffer updates

Propagating the intervention. SGD carries gradient changes into parameters and optimizer state, while PB selects running-buffer updates. The updated parameters determine the next batch's activations and moments. Each coalition evolves its own state so that its endpoint loss includes effects propagated through subsequent training.

Within each coalition, streams with identical inputs and operations share outputs. When receiver and selected source streams coincide, they use native BatchNorm. This implementation reproduces ordinary factual and deletion training for the empty and full coalitions, respectively. Appendix A specifies the implementation and precision comparisons.

3.3 Allocating route interactions

The effect of changing peer statistics depends on which receivers contribute loss. EGR averages the change in final loss caused by switching each route to its deletion setting over all route permutations. The active route set $\mathcal{P}_I$ contains D, SF, SP, PB for single-occurrence deletion and additionally LM for trajectory deletion. Let $R_{q,I}(S)$ denote the complete replay for coalition $S \subseteq \mathcal{P}_I$. The coalition value is the source-set endpoint loss difference from the all-factual replay:

$$v_{q,I}(S) = J_{E_{\mathrm{src}}}(s_T^{R_{q,I}(S)}) - J_{E_{\mathrm{src}}}(s_T^{R_{q,I}(\varnothing)}). \tag{3}$$

Writing $p = |\mathcal{P}_I|$ and suppressing the indices q, I, the Shapley allocation is (Shapley, 1953)

$$\phi_g = \sum_{S \subseteq \mathcal{P}_I \setminus \{g\}} \frac{|S|!(p - |S| - 1)!}{p!} [v(S \cup \{g\}) - v(S)], \qquad \sum_{g \in \mathcal{P}_I} \phi_g = v(\mathcal{P}_I). \tag{4}$$

Reading an allocation. The sign of ϕ_g shows whether switching route g to deletion raises or lowers final loss on average over other route settings. Each contribution includes its share of interaction effects.

For an interaction of size 1 requiring D, SF, and SP, Shapley assigns $1/3$ to each and zero to PB. Merging SF and SP gives $1/2$ each to D and statistics; alternatively, splitting SP into two necessary subroutes gives four shares of $1/4$. The total remains one, but its allocation depends on the route partition (Appendix F).

To measure how different routes improve prediction of deletion effects, we form three scores from the same Shapley allocation:

$$\begin{aligned} \mathcal{A}_{\mathrm{base}} &= \begin{cases} \{D\}, & I = \text{once}, \\ \{D, LM\}, & I = \text{trajectory}, \end{cases} & \mathcal{A}_{\mathrm{stats}} &= \mathcal{A}_{\mathrm{base}} \cup \{SF, SP\}, \\ \mathcal{A}_{\mathrm{full}} &= \mathcal{P}_I, & \widehat{H}_{\mathcal{A}} &= -\sum_{g \in \mathcal{A}} \phi_g. \end{cases} \end{aligned} \tag{5}$$

We refer to these as the baseline, statistic-aware, and full scores. The minus sign makes loss reductions correspond to positive benefits (Equation 1). The full score equals the source-set deletion benefit; its target-set error measures transfer between evaluation sets. Computing all route contributions requires 2^p coalition replays: 16 for single-occurrence deletion and 32 for trajectory deletion.

3.4 Batch size and aggregate peer effects

Batch size affects both averaging and the number of peers exposed to changed moments. We separate these effects to compare aggregate statistic contributions with loss participation. For a batch of $B \geq 2$ examples, let a_j be example j's scalar activation in one channel, $\mu = B^{-1} \sum_j a_j$ the mean, and $r = B^{-1} \sum_j a_j^2$ the mean square. Removing example i while holding the remaining activations fixed gives

$$\mu_{-i} - \mu = \frac{\mu - a_i}{B-1}, \qquad r_{-i} - r = \frac{r - a_i^2}{B-1}, \tag{6}$$

and, for bounded activations, $\sigma^2_{-i} - \sigma^2 = O(1/B)$ because $\sigma^2 = r - \mu^2$. With fixed $\epsilon > 0$ and the uniform derivative bounds and local Lipschitz conditions in Appendix B, a fixed peer's normalized activation and unaveraged loss gradient change by $O(1/B)$. For K affected peers, averaging their gradient changes over B receivers gives

$$\|\Delta \bar{g}_{SP,K}\| = O(K/B^2). \tag{7}$$

For $K = B - 1$, this bound is $O(1/B)$, matching the order of the bound for removing one bounded loss-gradient term and reaveraging with statistics fixed. The aggregate peer effect depends on whether individual gradient changes align or cancel.

4 Prediction and Propagation of Deletion Effects

We first test whether combining loss and statistic contributions improves prediction of deletion effects versus using loss contributions alone or standard attribution scores. Controlled interventions then examine the roles of affected peers, later batch composition, and interactions across layers.

Data and evaluation. The main benchmark applies single-occurrence deletion to 16 independently trained CIFAR-10 ResNet-18 checkpoints (He et al., 2016; Krizhevsky, 2009). Each checkpoint provides eight focal deletions in each of three batches, replaying each deletion along two continuation sequences. Evaluation occurs immediately after intervention and after 2, 10, 34, 130, and 393 updates. Appendix D specifies the sampling and training configuration.

The CIFAR-10 and CIFAR-100 prediction studies use disjoint 5,000-image test halves for source and target evaluation. Mean absolute error (MAE) measures error in predicted deletion benefit on the mean cross-entropy (CE) scale; Kendall τ_b measures rank correlation between predicted and actual deletion benefits (Koh et al., 2019). Independent training runs receive equal weight; Appendix C defines bootstrap intervals and resampling units.

Attribution baselines. We compare against all-parameter TracInCP (Pruthi et al., 2020), damped last-layer influence functions (IF) (Koh & Liang, 2017), and regularized single-model TRAK (Park et al., 2023), evaluated after 2, 34, and 393 continuation updates (Appendix E). To compare MAE on a common cross-entropy scale, we fit a monotone affine map from scores to deletion benefits using 15 models and evaluate on the remaining model, holding out each model in turn.

4.1 Statistic routes improve prediction through continued training

Comparing $D + SF + SP$ with D measures the predictive gain from normalization statistics. The gains persist across all six horizons (Figure 2). After 393 updates, MAE falls from 0.0319 to 0.0106, a 66.9% reduction, and Kendall τ_b rises from 0.391 to 0.822. Both improvements are positive in all 16 checkpoints, with 95% intervals $[0.0177, 0.0251]$ for MAE reduction and $[0.374, 0.497]$ for ordering gain.

The statistic-aware score ranks deletions more accurately than TracInCP, last-layer IF, and single-model TRAK at all three comparison horizons (Table 2a). Every model shows positive ordering gains in each comparison (Holm-adjusted $p < 0.001$). Including the buffer contribution in full EGR further improves short-horizon ordering.

At 393 updates, the $D + SF + SP$ score achieves calibrated MAE 0.0106, versus 0.0419 for TracInCP, IF, and TRAK. Appendix E reports calibrated errors at all three horizons.

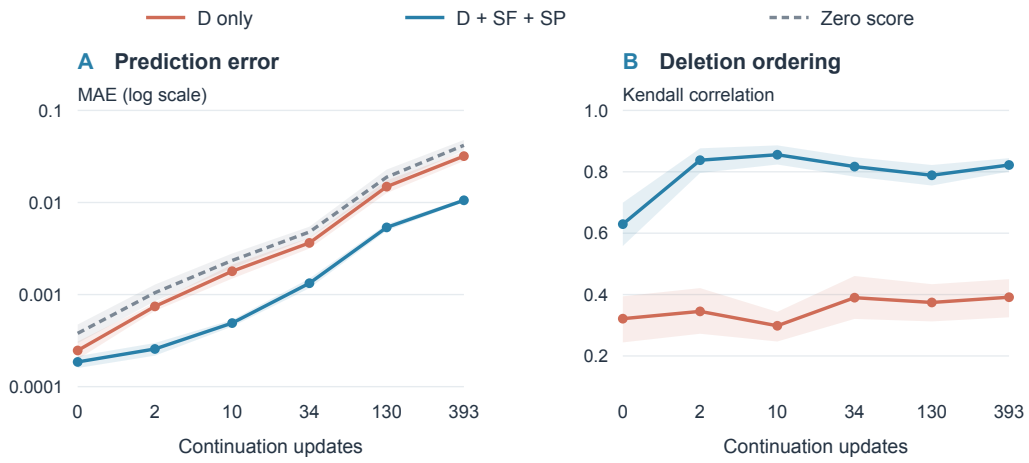


Figure 2: Prediction of deletion effects on CIFAR-10 ResNet-18. (A) MAE on a log scale, including a constant-zero predictor. (B) Kendall τ_b. Bands show 95% bootstrap intervals. Continuation horizons are equally spaced categories; 0 denotes the intervention update.

Table 2: Prediction of deletion effects. (a) Kendall τ_b before calibration on CIFAR-10 ResNet-18. (b) Two-update prediction on CIFAR-100 WideResNet-28-10; MAE is multiplied by 10^3. Bold marks the best value in each column; dashes denote unevaluated baselines.

	(a) CIFAR-10: $\tau_b \uparrow$			(b) CIFAR-100: 2 updates	
Method	2	34	393	MAE $\downarrow$	$\tau_b \uparrow$
TracInCP (all parameters)	0.164	−0.005	−0.004	–	–
IF (damped last layer)	0.121	−0.010	0.023	–	–
TRAK (single model)	0.126	0.013	0.019	–	–
EGR: D	0.345	0.390	0.391	2.69	0.219
EGR: $D + SF + SP$	0.838	**0.817**	**0.822**	1.56	**0.750**
Full EGR	**0.874**	0.816	**0.822**	**1.49**	0.746

Deciding whether to delete also requires predicting whether removal helps or hurts. At 393 updates, the uncalibrated $D + SF + SP$ score correctly predicts the sign of target deletion benefit in 91.54% of cases, compared with 73.18% for D and 44.92–49.61% for the standard baselines (Appendix Table 11).

To examine the roles of batch context and source output, we modify the baseline variants. Computing TracInCP's focal gradient in its original batch raises Kendall τ_b from 0.164 to 0.201 after two updates; at 393 updates, it is 0.010. Using CE-based source features for TRAK gives $\tau_b = 0.064$ and 0.012 at these horizons (Appendix Table 10).

Eight CIFAR-100 WideResNet-28-10 runs test whether the prediction gain extends to a wider model and a 100-class task (Zagoruyko & Komodakis, 2016). Adding $SF + SP$ to D reduces two-update MAE by 41.9% and raises Kendall τ_b by 0.531, improving both metrics in every run (Table 2b; Appendix G).

Training phase. To examine how the usefulness of statistics changes during training, we pair four CIFAR-10 runs at epochs 30 and 60. Adding $SF + SP$ to D lowers MAE in all four runs at epoch 30 and one at epoch 60, while the ratio of mean $|\phi_{SP}|$ to mean $|\phi_D|$ rises from 0.87 to 5.15 (Appendix I).

Mean absolute contributions decrease for all routes, but at different rates: D shrinks by a factor of 150.7, SP by 25.4, and PB by 4.93. The larger relative contributions of statistic routes therefore

reflect their slower decrease. Deletion rankings also change between the two epochs, with pooled within-run Kendall $\tau_b = 0.117$ for matched examples.

4.2 Small replay errors amplify through training

To distinguish deletion effects from numerical drift, we compare shared computations with separately evaluated moments. Shared computations exactly reproduce factual states and losses at all tested horizons in the four-route and layer-split games. Separately computed moments introduce rounding differences; after 393 updates, the mean absolute error in source-set deletion benefit reaches 0.0587, exceeding the mean absolute source–target benefit difference of 0.0106 (Appendix Table 5).

To test whether amplification depends on error direction, we perturb factual and deletion states immediately after the intervention update using the observed numerical error, its negative, or a random vector with matched tensor norms. After 393 updates, the perturbed deletion branches have mean ℓ_2 parameter distances of 9.206, 9.206, and 9.197 from the ordinary deletion branch. The resulting scores have Kendall τ_b values of 0.016, 0.022, and -0.076, versus 0.822 without perturbation (Appendix A.4).

4.3 Propagation through peers and later batch membership

The peer-statistic contribution affects training through peers' gradients. GroupNorm and frozen BatchNorm remove dependence on shared current-batch statistics, providing controls for this mechanism. Both yield zero allocations for SF, SP, and PB, while D remains active; LM also remains active in the GroupNorm trajectory experiment (Appendices J–K).

Batch size and affected peer count. We vary batch size while tracking the same seven peers, then vary the affected-peer count K at fixed $B = 128$ (Appendix H). The first comparison measures how

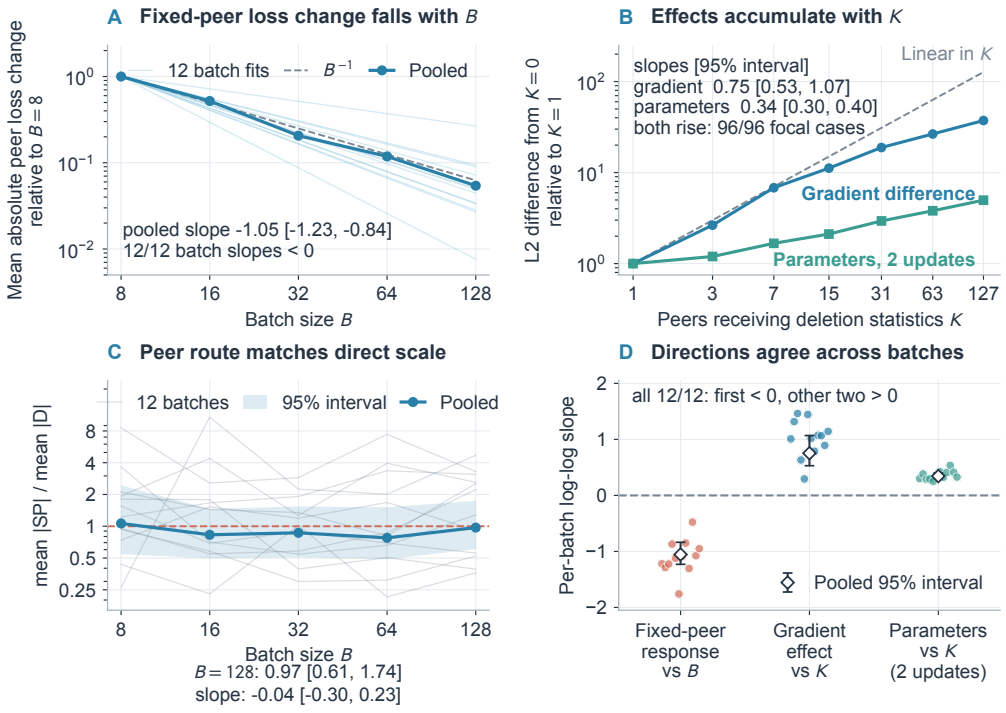


Figure 3: Batch size and aggregate peer effects. (A) Mean absolute pre-update peer loss change versus B. (B) Gradient and parameter ℓ_2 distances from $K = 0$, normalized by their $K = 1$ values. Parameters are measured after two continuation updates. (C) Mean $|\phi_{SP}|$/mean $|\phi_D|$. (D) Per-batch log–log slopes. Intervals are 95% hierarchical bootstrap intervals.

batch averaging reduces individual loss changes; the second measures how peers' gradient changes combine in an update.

Mean absolute pre-update loss change among the fixed peers scales as $B^{-1.055}$, with a slope interval of $[-1.230, -0.836]$ (Figure 3). The aggregate peer contribution remains comparable to loss participation: at $B = 128$, mean $|\phi_{SP}|$/mean $|\phi_D|$ is 0.971 $[0.609, 1.742]$.

Tracking the changed gradients into later parameters shows how accumulation persists through continued training. Log–log slopes against peer count are 0.753 $[0.530, 1.069]$ at the intervention update and 0.345 $[0.303, 0.401]$ after two continuation updates; the latter is positive in all 12 batches. Increasing K from 1 to 127 multiplies gradient distance by 37.31 and parameter distance by 4.97.

This accumulation also affects evaluation loss: the mean absolute target-loss difference from $K = 0$ grows from 8.53×10^{-5} at $K = 1$ to 7.10×10^{-4} at $K = 127$, a factor of 8.33 $[6.25, 14.11]$. To test whether statistic contributions are example-specific, we shuffle the $SF + SP$ score increments among focal examples while keeping D fixed. Compared with shuffled assignments, correct matching lowers MAE in 59 of 60 batch–size combinations and raises Kendall τ_b in 58 (Appendix H).

Deleting before batch construction. The trajectory study measures the contribution of changes in later batch composition. Across four MLP runs with three FashionMNIST replay epochs (Xiao et al., 2017), LM has the largest mean absolute contribution, 0.0495, versus 0.0195 for D. The corresponding values for SF and SP are 0.0187 and 0.0172. Adding $SF + SP$ to $D + LM$ lowers MAE and raises Kendall τ_b in every run.

4.4 Attribution across route partitions

Route partitions determine how interactions are divided among routes, including D. We vary the partition to compare the overall statistic contribution and identify opposing effects across layers.

We compare the original, merged, and layer-split partitions on the six-horizon benchmark. The merged game combines SF and SP into one statistic route; the layer-split game separates SP into early layers (stem and stages 1–2) and late layers (stages 3–4), including projection BatchNorm in its corresponding stage. We recompute all Shapley values for each partition and sum its statistic contributions before taking the absolute value. Appendix F specifies the averaging procedure.

After 393 updates, statistic contributions are 1.02–1.66 times the mean absolute magnitude of D across the three partitions (Table 3). Adding them to each partition's D score reduces MAE by 0.0202–0.0230 and raises Kendall τ_b by 0.431–0.492. Appendix Table 12 reports all horizons.

Table 3: Route partitions after 393 updates. Statistics / D is the source-set magnitude ratio defined in Appendix F. MAE gains are multiplied by 10^3; brackets show 95% bootstrap intervals.

Partition	Statistics / D	MAE gain	τ_b gain
Four routes	1.237 $[1.151, 1.322]$	21.3	0.431
Merged SF, SP	1.024 $[0.956, 1.092]$	20.2	0.463
Layer-split SP	1.662 $[1.568, 1.755]$	23.0	0.492

The mean absolute change in ϕ_D is 39.5% of its original mean absolute magnitude after merging SF and SP, and 47.5% after splitting SP. Switching D with other routes all factual or all at deletion yields unchanged loss differences across partitions; its Shapley allocation changes through intermediate coalitions.

The layer split reveals cancellation within the peer contribution. Summing early- and late-layer contributions before taking absolute values gives a mean magnitude of 0.0297, versus 0.0400 when their magnitudes are summed. This reflects substantial opposing contributions across layers.

Conditional effects within the game. For single-occurrence deletion, SF affects loss only when D is factual. After 393 updates, its mean absolute Shapley contribution is 0.0153, but its leave-one-route-out effect is zero: the focal receiver is absent in both compared replays.

5 USING ROUTES TO GUIDE TRAINING

Opposing loss and statistic contributions mean that deletion removes both beneficial and harmful effects. Selective intervention can preserve useful contributions. We use contribution signs to propose loss masking, statistic exclusion, and buffer exclusion, then evaluate their combinations to account for interactions. The policy selects the candidate that most reduces clean-source CE after one update relative to keeping the example; if none improves CE, it keeps the example unchanged (Appendix L).

Image noise changes gradients and shared statistics without changing labels, making it a test of where to intervene. We evaluate 16 new CIFAR-10 ResNet-18 initializations on a task selected in a four-model exploration using the same target images. During 64 updates with batches of eight, half the updates add Gaussian noise to one image. Evaluation uses 8,000 test images under a fixed brightness-and-contrast transformation, the *exposure shift*.

Ordinary training tests whether intervention helps; high-loss filtering tests a loss-based masking rule. The latter masks the focal loss when it exceeds every peer loss. EGR raises mean accuracy by 0.56 and 0.84 percentage points over these baselines, respectively (Table 4), with gains in 14 of 16 models against each. Adjusted paired intervals for these accuracy gains are $[0.21, 0.92]$ and $[0.42, 1.26]$; mean CE also decreases in both comparisons.

To examine which actions support the gain, we restrict EGR's actions to loss masking and buffer exclusion (D, PB). This policy achieves the highest mean accuracy, 85.09%, and lowest mean CE, 0.4739, in the comparison. Fixed statistic exclusion reaches 85.05% mean accuracy, also higher than the full policy.

Direct search tests whether the same actions are useful without contribution-based proposals. It selects the action with the lowest clean-source CE after one update. Searching the four D, PB actions achieves mean accuracy 85.01% and CE 0.4761; adding statistic exclusion and full deletion gives 85.00% and 0.4780. The four-action set yields higher mean accuracy and lower mean CE under both EGR and direct search.

To test whether the gains persist, we continue all models with ordinary training for 32 updates without added noise. EGR retains mean exposure-shift accuracy gains of 0.16 and 0.11 percentage points over ordinary training and high-loss filtering, respectively. However, high-loss filtering then achieves lower exposure-shift CE and higher clean-image accuracy than EGR (Appendix Table 22).

Table 4: Mean accuracy and CE across 16 models after 64 updates under the exposure shift. Appendix Table 21 reports all 22 methods.

Method	Accuracy (%) ↑	CE ↓
Ordinary training	84.42	0.4968
High-loss filtering	84.15	0.5024
EGR-guided policy	84.98	0.4803
Direct-route policy	84.45	0.4930
Direct search (six actions)	85.00	0.4780
EGR policy (D, PB actions)	85.09	0.4739
Always exclude focal statistics	85.05	0.4843

6 CONCLUSION

Shared normalization statistics contribute on the same scale as loss participation, with effects accumulating across affected peers. EGR separates these contributions as they propagate through training, improving prediction of deletion effects beyond three standard attribution baselines. The decomposition also guides selective interventions: the policy restricted to loss masking and buffer exclusion achieves the highest mean accuracy in our exposure-shift comparison.

AI USE STATEMENT

We used AI tools to identify relevant literature, assist in analyzing experimental data and interpreting the results, and refine the language of the manuscript for clarity and readability. All AI-assisted work was reviewed by the authors.

REPRODUCIBILITY STATEMENT

Code and experimental measurements are available at:
https://anonymous.4open.science/r/egr-review-9DB5/.

Section 3 defines the interventions, replay procedure, and attribution rule. Appendices A and B provide implementation details, numerical checks, and the derivation of the peer-gradient bound. Appendix C specifies random seeds and statistical procedures. Experiment-specific data splits, model and optimizer settings, and baseline definitions are documented in Appendices D–L.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. doi: 10.48550/arXiv.1607.06450. URL https://arxiv.org/abs/1607.06450.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B. Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, volume 35, pp. 17953–17967, 2022. doi: 10.52202/068431-1305. URL https://papers.nips.cc/paper_files/paper/2022/hash/7234e0c36fdbcb23e7bd56b68838999b-Abstract-Conference.html.

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger Grosse. Training data attribution via approximate unrolling. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66647–66686, 2024. doi: 10.52202/079017-2129. URL https://papers.nips.cc/paper_files/paper/2024/hash/7af60ccb99c7a434a0d9d9c1fb00ca94-Abstract-Conference.html.

Samyadeep Basu, Phillip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xHKVVHGDOEk.

Brian Wesley Bell, Michael Geyer, David Glickenstein, Amanda S. Fernandez, and Juston Moore. An exact kernel equivalence for finite classification models. In *Proceedings of the 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pp. 206–217. PMLR, 2023. URL https://proceedings.mlr.press/v221/bell23a.html.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy*, pp. 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019. URL https://doi.org/10.1109/SP40001.2021.00019.

Landon Butler, Abhineet Agarwal, Justin Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. ProxySPEX: Inference-efficient interpretability via sparse feature interactions in LLMs. In *Advances in Neural Information Processing Systems*, volume 38, pp. 72306–72340, 2025. doi: 10.52202/085713-2426. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/68dced1638d71d4d21598f79ae91262e-Abstract-Conference.html.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015. doi: 10.1109/SP.2015.35. URL https://doi.org/10.1109/SP.2015.35.

Tyler A. Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable influence and fact tracing for large language model pretraining. In *International Conference on Learning Representations*, pp. 40976–40997, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/65798a76cc176c29b6bfefe84b0a03ff-Abstract-Conference.html.

Jianhui Chen, Yuzhang Luo, and Liangming Pan. Mechanistic data attribution: Tracing the training origins of interpretable LLM units. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PQaxfoEcRc.

Yuanyuan Chen, Boyang Li, Han Yu, Pengcheng Wu, and Chunyan Miao. HyDRA: Hypergradient data relevance analysis for interpreting deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7081–7089, 2021. doi: 10.1609/aaai.v35i8.16871. URL https://ojs.aaai.org/index.php/AAAI/article/view/16871.

Junwei Deng, Weijing Tang, and Jiaqi W. Ma. A versatile influence function for data attribution with non-decomposable loss. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13256–13276. PMLR, 2025. URL https://proceedings.mlr.press/v267/deng25h.html.

Junwei Deng, Pingbang Hu, Suliang Jin, Hao Lu, Jiachen T. Wang, Shichang Zhang, and Jiaqi W. Ma. How faithful is trajectory-based data attribution? Error sources, remedies, and practical guidelines. *arXiv preprint arXiv:2605.18814*, 2026. doi: 10.48550/arXiv.2605.18814. URL https://arxiv.org/abs/2605.18814.

Ngoc Phu Doan, Chongyan Gu, and Ihsen Alouani. Batch normalization amplifies memorization and privacy risks. *arXiv preprint arXiv:2605.24420*, 2026. URL https://arxiv.org/abs/2605.24420.

Florian Eichin, Yupei Du, Philipp Mondorf, Maria Matveev, Barbara Plank, and Michael A. Hedderich. ExPLAIND: Unifying model, data, and training attribution to study model behavior. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=7G6x9QTaN4.

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. URL https://papers.nips.cc/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html.

Kostadin Garov, Dimitar I. Dimitrov, Nikola Jovanović, and Martin Vechev. Hiding in plain sight: Disguising data stealing attacks in federated learning. In *International Conference on Learning Representations*, pp. 24519–24545, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6ac754e9a4107bd00dc3d72275629d99-Abstract-Conference.html.

Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, 2021. doi: 10.1609/aaai.v35i13.17371. URL https://ojs.aaai.org/index.php/AAAI/article/view/17371.

Kelvin Guu, Albert Webson, Ellie Pavlick, Lucas Dixon, Ian Tenney, and Tolga Bolukbasi. Simfluence: Modeling the influence of individual training examples by simulating training runs. *arXiv preprint arXiv:2303.08114*, 2023. doi: 10.48550/arXiv.2303.08114. URL https://arxiv.org/abs/2303.08114.

Reza Habibi, Darian Lee, and Magy Seif El-Nasr. Symbolic mechanistic data attribution: Tracing training influence to learned behavioral policies. *arXiv preprint arXiv:2606.29171*, 2026. URL https://arxiv.org/abs/2606.29171.

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with SGD. In *Advances in Neural Information Processing Systems*, volume 32, pp. 4215–4224, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5f14615696649541a025d3d0f8e0447f-Abstract.html.

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. IEEE, 2016. doi: 10.1109/CVPR.2016.90. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/He_Momentum_Contrast_for_Unsupervised_Visual_Representation_Learning_CVPR_2020_paper.html.

Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W. Ma. Most influential subset selection: Challenges, promises, and beyond. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119778–119810, 2024. doi: 10.52202/079017-3806. URL https://papers.nips.cc/paper_files/paper/2024/hash/d8684e49752e06ac5e4b554b60ad212a-Abstract-Conference.html.

Andrew Ilyas and Logan Engstrom. MAGIC: Near-optimal data attribution for deep learning. *arXiv preprint arXiv:2504.16430*, 2025. doi: 10.48550/arXiv.2504.16430. URL https://arxiv.org/abs/2504.16430.

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. PMLR, 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1945–1953, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/c54e7837e0cd0ced286cb5995327d1ab-Abstract.html.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5255–5265, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/a78482ce76496fcf49085f2190e675b4-Abstract.html.

Atli Kosson, Dongyang Fan, and Martin Jaggi. Ghost noise for regularizing deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13274–13282, 2024. doi: 10.1609/aaai.v38i12.29228. URL https://ojs.aaai.org/index.php/AAAI/article/view/29228.

Zhi Kou, Kaichao You, Mingsheng Long, and Jianmin Wang. Stochastic normalization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16304–16314, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/bc573864331a9e42e4511de6f678aa83-Abstract.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Antoine Labatie, Dominic Masters, Zach Eaton-Rosen, and Carlo Luschi. Proxy-normalizing activations to match batch normalization while removing batch dependence. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16990–17006. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/8d2a5f7d4afa5d0530789d3066945330-Abstract.html.

Janghyeon Lee, Hyeong Gwon Hong, Donggyu Joo, and Junmo Kim. Continual learning with extended Kronecker-factored approximate curvature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9001–9010, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Lee_Continual_Learning_With_Extended_Kronecker-Factored_Approximate_Curvature_CVPR_2020_paper.html.

Jin Hwa Lee, Matthew Smith, Maxwell Adam, and Jesse Hoogland. Influence dynamics and stagewise data attribution. In *International Conference on Learning Representations*, pp. 35716–35747, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3c1edb917edb6cec59bf3a0afc03150e-Abstract-Conference.html.

Xiangru Lian and Ji Liu. Revisit batch normalization: New understanding and refinement via composition optimization. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3254–3263. PMLR, 2019. URL https://proceedings.mlr.press/v89/lian19a.html.

Jinkun Lin, Anqi Zhang, Mathias Lécuyer, Jinyang Li, Aurojit Panda, and Siddhartha Sen. Measuring the effect of training data on deep learning predictions via randomized experiments. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13468–13504. PMLR, 2022. URL https://proceedings.mlr.press/v162/lin22h.html.

Yilin Lyu, Liyuan Wang, Xingxing Zhang, Zicheng Sun, Hang Su, Jun Zhu, and Liping Jing. Overcoming recency bias of normalization statistics in continual learning: Balance and adaptation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 25475–25494. Curran Associates, Inc., 2023. doi: 10.52202/075280-1108. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/50ca96a1a9ebe0b5e5688a504feb6107-Abstract-Conference.html.

Bruno Kacper Mlodozeniec, Isaac Reid, Sam Power, David Krueger, Murat A. Erdogdu, Richard E. Turner, and Roger Baker Grosse. Distributional training data attribution: What do influence functions sample? In *Advances in Neural Information Processing Systems*, volume 38, pp. 10057–10091, 2025. doi: 10.52202/085713-0339. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0e8909cae8248c98279f6cd82074aa6d-Abstract-Conference.html.

Siqiao Mu and Diego Klabjan. Rewind-to-delete: Certified machine unlearning for nonconvex functions. In *Advances in Neural Information Processing Systems*, volume 38, pp. 114911–114947, 2025. doi: 10.52202/085713-3832. URL https://papers.nips.cc/paper_files/paper/2025/hash/a6bd9cbda87b508b8254b3a420448405-Abstract-Conference.html.

Naoki Murata, Yuhta Takida, Chieh-Hsin Lai, Toshimitsu Uesaka, Bac Nguyen, Stefano Ermon, and Yuki Mitsufuji. GUDA: Counterfactual group-wise training data attribution for diffusion models via unlearning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/66223.

Aviraj Newatia, Michael Cooper, Viet Nguyen, and Rahul G. Krishnan. Mitigating privacy risk via forget set-free unlearning. In *International Conference on Learning Representations*, pp. 47590–47636, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4f0998c1e548fc4adfb360a92b36d8cd-Abstract-Conference.html.

Elisa Nguyen, Minjoon Seo, and Seong Joon Oh. A Bayesian approach to analysing training data attribution in deep learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64155–64180. Curran Associates, Inc., 2023. doi: 10.52202/075280-2803. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ca774047bc3b46cc81e53ead34cd5d5a-Abstract-Conference.html.

Subhodip Panda, Dhruv Tarsadiya, Shashwat Sourav, Prathosh A. P., and Sai Praneeth Karimireddy. f-INE: A hypothesis testing framework for estimating influence under training randomness. In *International Conference on Learning Representations*, pp. 89939–89962, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/916ee60e315531d6b3954af8a8dc3437-Abstract-Conference.html.

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. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Yonghyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, Naoki Murata, Wei-Hsiang Liao, Woosung Choi, Kin Wai Cheuk, Junghyun Koo, and Yuki Mitsufuji. Concept-TRAK: Understanding how diffusion models learn concepts through concept attribution. In *International Conference on Learning Representations*, pp. 93014–93041, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/962a0f260ce5a591a40eeb0ef3978567-Abstract-Conference.html.

Lukas Pfahler and Katharina Morik. Explaining deep learning representations by tracing the training process. *arXiv preprint arXiv:2109.05880*, 2021. URL https://arxiv.org/abs/2109.05880.

Quang Pham, Chenghao Liu, and Steven C. H. Hoi. Continual normalization: Rethinking batch normalization for online continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vwLLQ-HwqhZ.

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. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *International Conference on Learning Representations*, pp. 70523–70563, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/afe1aa79e5eea7955f553c61a307273e-Abstract-Conference.html.

Keivan Rezaei, Mehrdad Saberi, Abhilasha Ravichander, and Soheil Feizi. Revisiting the past: Data unlearning with model state history. In *International Conference on Learning Representations*, pp. 112809–112829, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b760de0a1126a3df214b334901b2a986-Abstract-Conference.html.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2226–2234, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html.

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*, volume 36, pp. 27560–27581, 2023. doi: 10.52202/075280-1198. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/57bb27b9be6ad04019ae3cea2b540872-Abstract.html.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, number 28 in Annals of Mathematics Studies, pp. 307–318. Princeton University Press, 1953. doi: 10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

Raghav Singal, George Michailidis, and Hoiyi Ng. Flow-based attribution in graphical models: A recursive Shapley approach. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9733–9743. PMLR, 2021. URL https://proceedings.mlr.press/v139/singal21a.html.

Saurabh Singh and Shankar Krishnan. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11237–11246. IEEE, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Singh_Filter_Response_Normalization_Layer_Eliminating_Batch_Dependence_in_the_Training_CVPR_2020_paper.html.

Saurabh Singh and Abhinav Shrivastava. EvalNorm: Estimating batch normalization statistics for evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3633–3641, 2019. doi: 10.1109/ICCV.2019.00373. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Singh_EvalNorm_Estimating_Batch_Normalization_Statistics_for_Evaluation_ICCV_2019_paper.html.

Weiwei Sun, Haokun Liu, Nikhil Kandpal, Colin Raffel, and Yiming Yang. Enhancing training data attribution with representational optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 102274–102311, 2025. doi: 10.52202/085713-3421. URL https://papers.nips.cc/paper_files/paper/2025/hash/940a76f6cc2e23867421e1cf1c8da4f3-Abstract-Conference.html.

Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9269–9278. PMLR, 2020. URL https://proceedings.mlr.press/v119/sundararajan20b.html.

Lucile Ter-Minassian, Oscar Clivio, Karla Diazordaz, Robin J. Evans, and Christopher C. Holmes. PWSHAP: A path-wise explanation model for targeted variables. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34054–34089. PMLR, 2023. URL https://proceedings.mlr.press/v202/ter-minassian23a.html.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022a. doi: 10.1109/EuroSP53844.2022.00027. URL https://doi.org/10.1109/EuroSP53844.2022.00027.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4007–4022, Boston, MA, August 2022b. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/thudi.

Stelios Triantafyllou, Aleksa Sukovic, Yasaman Zolfimoselo, and Goran Radanovic. Counterfactual effect decomposition in multi-agent sequential decision making. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60072–60098. PMLR, 2025. URL https://proceedings.mlr.press/v267/triantafyllou25a.html.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A. Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Proceedings of the 34th Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 4126–4142. PMLR, 2021. URL https://proceedings.mlr.press/v134/ullah21a.html.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4105–4113. IEEE, 2017. doi: 10.1109/CVPR.2017.437. URL https://doi.org/10.1109/CVPR.2017.437.

Fei Wang and Baochun Li. Hear no evil: Detecting gradient leakage by malicious servers in federated learning. *arXiv preprint arXiv:2506.20651*, 2025. doi: 10.48550/arXiv.2506.20651. URL https://arxiv.org/abs/2506.20651.

Jiachen Tianhao Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data Shapley in one training run. In *International Conference on Learning Representations*, pp. 12358–12395, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/20fdaf67581e6d7157376d1ed584040a-Abstract-Conference.html.

Jiachen Tianhao Wang, Dawn Song, James Y. Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *International Conference on Learning Representations*, pp. 11236–11273, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1e0d38c676d5855bcfab7f6d29d20ad9-Abstract-Conference.html.

Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. Shapley Flow: A graph-based approach to interpreting model predictions. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 721–729. PMLR, 2021. URL https://proceedings.mlr.press/v130/wang21b.html.

Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. In Qiang Yang, Lixin Fan, and Han Yu (eds.), *Federated Learning: Privacy and Incentive*, volume 12500 of *Lecture Notes in Computer Science*, pp. 153–167. Springer International Publishing, Cham, 2020. ISBN 978-3-030-63076-8. doi: 10.1007/978-3-030-63076-8_11. URL https://doi.org/10.1007/978-3-030-63076-8_11.

David Xing Wu, Chulhee Yun, and Suvrit Sra. On the training instability of shuffling SGD with batch normalization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 37787–37845. PMLR, 2023. URL https://proceedings.mlr.press/v202/wu23x.html.

Xindi Wu, Despoina Paschalidou, Jun Gao, Antonio Torralba, Laura Leal-Taixé, Olga Russakovsky, Sanja Fidler, and Jonathan Lorraine. Motion attribution for video generation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zAl9heLw4q.

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. PMLR, 2020a. URL https://proceedings.mlr.press/v119/wu20b.html.

Yinjun Wu, Val Tannen, and Susan B. Davidson. PrIU: A provenance-based approach for incrementally updating regression models. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 447–462. Association for Computing Machinery, 2020b. doi: 10.1145/3318464.3380571. URL https://doi.org/10.1145/3318464.3380571.

Yuxin Wu and Kaiming He. Group normalization. In *Computer Vision – ECCV 2018*, volume 11217 of *Lecture Notes in Computer Science*, pp. 3–19. Springer, 2018. doi: 10.1007/978-3-030-01261-8_1. URL https://doi.org/10.1007/978-3-030-01261-8_1.

Yuxin Wu and Justin Johnson. Rethinking "batch" in BatchNorm. *arXiv preprint arXiv:2105.07576*, 2021. doi: 10.48550/arXiv.2105.07576. URL https://arxiv.org/abs/2105.07576.

Zhaoxuan Wu, Yao Shu, and Bryan Kian Hsiang Low. DAVINZ: Data valuation using deep neural networks at initialization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24150–24176. PMLR, 2022. URL https://proceedings.mlr.press/v162/wu22j.html.

Haocheng Xia, Jiayao Zhang, Qiheng Sun, Jinfei Liu, Kui Ren, Li Xiong, and Jian Pei. Computing Shapley values for dynamic data. *IEEE Transactions on Knowledge and Data Engineering*, 37 (6):3253–3271, June 2025. doi: 10.1109/TKDE.2025.3527380. URL https://doi.org/10.1109/TKDE.2025.3527380.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. doi: 10.48550/arXiv.1708.07747. URL https://arxiv.org/abs/1708.07747.

Huanyi Xie, Chenyang Ren, Khouloud Saadi, Shu Yang, Zhen Tan, Jingfeng Zhang, Lijie Hu, and Di Wang. Dissecting representation misalignment in contrastive learning via influence function. In *International Conference on Learning Representations*, pp. 62221–62256, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/65848e674f556cee56492c0b502eced8-Abstract-Conference.html.

Xiangzhe Xu, Hongyu Liu, Guanhong Tao, Zhou Xuan, and Xiangyu Zhang. Checkpointing and deterministic training for deep learning. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, CAIN '22, pp. 65–76. Association for Computing Machinery, 2022. doi: 10.1145/3522664.3528605. URL https://doi.org/10.1145/3522664.3528605.

Junjie Yan, Ruosi Wan, Xiangyu Zhang, Wei Zhang, Yichen Wei, and Jian Sun. Towards stabilizing batch statistics in backward propagation of batch normalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgGjRVKDS.

Zhuliang Yao, Yue Cao, Shuxin Zheng, Gao Huang, and Stephen Lin. Cross-iteration batch normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12331–12340, 2021. doi: 10.1109/CVPR46437.2021.01215. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yao_Cross-Iteration_Batch_Normalization_CVPR_2021_paper.html.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, 2016. doi: 10.5244/C.30.87. URL https://doi.org/10.5244/C.30.87.

Dewen Zeng, Wenlong Tian, Haozhao Wang, Jianfeng Lu, Weijun Xiao, and Zhiyong Xu. Ripple Shapley: Data influence attribution in one federated training run. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28085–28093, 2026. doi: 10.1609/aaai.v40i33.40034. URL https://ojs.aaai.org/index.php/AAAI/article/view/40034.

A Replay Implementation and Numerical Accuracy

Within each coalition, the streams in Equation 2 share parameters and pool BatchNorm2d moments across batch and spatial dimensions. The forward pass uses the biased minibatch variance σ_b^2; the running variance uses the unbiased estimate σ_u^2. For $n > 1$ activations per channel and BatchNorm momentum ρ,

$$\sigma_u^2 = \frac{n}{n-1}\sigma_b^2, \qquad \bar{\sigma}_{t+1}^2 = (1-\rho)\bar{\sigma}_t^2 + \rho\sigma_u^2. \tag{8}$$

Replay procedure. The inputs are a training state s_t, a focal example, a removal rule, the batch sequence and input tensors, and the source evaluation set. For each coalition S, we restore a separate copy of s_t and keep its route choices fixed until the endpoint. The replay proceeds as follows.

1. Construct the factual and deletion batches at each affected update. For single-occurrence deletion, the deletion batch omits the focal example. For trajectory deletion, remove it from the epoch order before batching; the first example of the next batch completes the affected batch.
2. Select receivers and labels using D. Single-occurrence deletion uses B or $B-1$ receivers. Trajectory deletion uses the factual batch or the rebatched batch, with the replacement as the focal receiver for SF.
3. Propagate the receiver and both statistic-source streams through the network using the coalition's current parameters. Each source normalizes itself with its own moments. Equation 2 selects the source for each receiver at every BatchNorm layer, retaining gradients through the selected moments.
4. Average CE over the selected receivers, backpropagate, and apply the optimizer update. Use PB to select the running-mean and running-variance updates computed in that forward pass. These buffer updates use detached moments; each layer's counter advances once.
5. Continue from the resulting parameters, optimizer state, and buffers. Single-occurrence deletion uses the fixed continuation batches. For trajectory deletion, LM selects the factual or rebatched suffix after each affected update, including its final partial batch. Repeat the intervention at the focal occurrence in each replay epoch.

At T, evaluate mean source-set CE in evaluation mode. Subtract the empty-coalition loss to obtain $v(S)$, then apply Equation 4 to the complete coalition table. All branches reuse the same realized input tensors and orders wherever their selected batches coincide.

A.1 Shared computation

Shared computation preserves native operations when the receiver and a statistic source coincide. Identical input tensors share the outputs of convolutions, normalization operations, activations, and residual additions; automatic differentiation accumulates gradients through these shared outputs. A receiver using another stream's moments is normalized explicitly. Each source computes candidate running-buffer updates from copies of the coalition's current buffers. The PB route commits the selected updates once, so evaluating both sources does not advance the coalition's buffers twice.

A.2 Precision comparison

We compare shared computation and separately computed moments against ordinary replay at 128 endpoints: four checkpoints, 16 deletions per checkpoint, and both factual and deletion branches after two continuation updates. Each comparison uses the same inputs and numerical precision. Agreement requires parameter and buffer differences within absolute and relative tolerances of 10^{-4}, identical counters, and a loss difference within $5 \times 10^{-5} + 10^{-4}|J_{\text{reference}}|$.

Shared computation matches ordinary replay at all 128 endpoints. Separate moment computation matches at 39 endpoints in FP32 with TF32, and at all 128 with TF32 disabled or with FP64. At one checkpoint, we compare shared computation with an independent implementation that computes each stream separately. In FP64, parameter, gradient, and momentum differences are of order 10^{-16} across the 16 original and 32 layer-split coalitions.

A.3 Numerical error and transfer between evaluation sets

Numerical error is the absolute difference between the full EGR score and ordinary source-set deletion benefit. Transfer error is the absolute difference between ordinary deletion benefits on the source and target sets. Error from separately computed moments exceeds transfer error from two updates onward (Table 5).

Table 5: Mean absolute numerical and source-to-target transfer errors, multiplied by 10^3.

Updates	Separate moments	Shared computation	Transfer error
0	0.0249	0	0.0748
2	0.226	0	0.197
10	1.46	0	0.482
34	5.22	0	1.33
130	25.5	0	5.36
393	58.7	0	10.6

A.4 Propagation of a small state perturbation

We use the six-horizon benchmark in Appendix D: 16 checkpoints, three probe batches per checkpoint, eight focal examples per batch, and two continuation sequences, giving 768 matched replay pairs. Immediately after the intervention update, each factual and deletion state receives the perturbation measured between its separate-moment and ordinary implementations, its negative, or one fixed random direction. The perturbations apply to parameters, optimizer momentum, and running means and variances, with counters fixed. Random perturbations use Rademacher entries, with each tensor's norm matched to the observed perturbation before FP32 storage. For the running-variance tensors taken together, the stored perturbation norm differs from the matched norm by at most 5.36%. All branches use the same continuation batches.

Table 6: Propagation of state perturbations. (a) Mean parameter distance from the ordinary deletion branch. (b) Full-score prediction MAE for target-set deletion benefit, multiplied by 10^3, using both perturbed endpoints to compute the source score. Reference denotes the unperturbed ordinary replay.

(a) Parameter ℓ_2 distance

Updates	Observed	Negative	Random
0	6.37×10^{-4}	6.37×10^{-4}	6.37×10^{-4}
2	0.00537	0.00527	0.00439
10	0.0434	0.0429	0.0386
34	0.318	0.317	0.306
130	2.498	2.497	2.477
393	9.206	9.206	9.197

(b) Full-score prediction MAE

Updates	Observed	Negative	Random	Reference	Zero score
0	0.0820	0.0791	0.0763	0.0748	0.379
2	0.337	0.299	0.284	0.197	1.05
10	1.57	1.58	1.36	0.482	2.35
34	5.26	5.02	4.70	1.33	4.78
130	25.7	26.2	25.1	5.36	18.8
393	58.1	65.2	57.5	10.6	41.8

B From Moment Changes to the Peer-Gradient Bound

We derive Equations 6 and 7 by bounding moment changes, their effect on one receiver, and the sum over affected receivers. Parameters, the receiver batch, and its mean-loss divisor B are fixed; routes other than SP remain factual.

Conditions. Let $a_j = a_j(\theta)$, $j = 1, \dots, B$, be the source activations, with $B \geq 2$ and focal index i. Assume $|a_j| \leq M$ and $\|\nabla_\theta a_j\| \leq L$, uniformly in j and B. The BatchNorm constant $\epsilon > 0$ is fixed. Affine parameters and their derivatives are bounded. The maps following normalization have bounded first derivatives that are Lipschitz in their activation arguments along the segment between factual and deletion activations. For a stack of layers, impose these conditions on each layer, with constants independent of B and the selected receiver count K. A ReLU network satisfies the required local smoothness within a region where its activation pattern stays fixed. We use Euclidean norms for vectors and the corresponding induced norms for Jacobians.

1. Changes in moments and their parameter derivatives. Write $\mu_0 = B^{-1}\sum_j a_j$, $r_0 = B^{-1}\sum_j a_j^2$, and $v_0 = r_0 - \mu_0^2$. Subscript 1 denotes moments after removing i, with divisor $B-1$, and $\delta x = x_1 - x_0$. Direct subtraction gives

$$\delta\mu = \frac{\mu_0 - a_i}{B-1}, \quad \delta r = \frac{r_0 - a_i^2}{B-1}, \quad \delta v = \delta r - (\mu_1 + \mu_0)\delta\mu. \tag{9}$$

The activation bound yields

$$|\delta\mu| \leq \frac{2M}{B-1}, \qquad |\delta r| \leq \frac{2M^2}{B-1}, \qquad |\delta v| \leq \frac{6M^2}{B-1}. \tag{10}$$

Differentiating the moment identities with respect to θ yields

$$\begin{aligned} \delta\nabla\mu &= \frac{\nabla\mu_0 - \nabla a_i}{B-1}, \\ \delta\nabla r &= \frac{\nabla r_0 - 2a_i\nabla a_i}{B-1}, \\ \delta\nabla v &= \delta\nabla r - 2\delta\mu\,\nabla\mu_1 - 2\mu_0\,\delta\nabla\mu. \end{aligned} \tag{11}$$

Here $\nabla = \nabla_\theta$ and $\delta\nabla x = \nabla x_1 - \nabla x_0$. Since $\|\nabla\mu_a\| \leq L$ and $\|\nabla r_a\| \leq 2ML$ for $a \in \{0, 1\}$, Equation 11 gives bounds $2L/(B-1)$, $4ML/(B-1)$, and $12ML/(B-1)$ for the three derivative differences, respectively. Also $\|\nabla v_a\| \leq 4ML$.

2. A fixed receiver's activation and loss gradient. For a retained receiver $j \neq i$, set $b_j^a = a_j - \mu_a$, $f(v) = (v+\epsilon)^{-1/2}$, and $u_j^a = b_j^a f(v_a)$. On $v \geq 0$, the absolute values of f, f', and f'' are bounded by $\epsilon^{-1/2}$, $\frac{1}{2}\epsilon^{-3/2}$, and $\frac{3}{4}\epsilon^{-5/2}$. Thus

$$\begin{aligned} |u_j^1 - u_j^0| &\leq \epsilon^{-1/2}|\delta\mu| + M\epsilon^{-3/2}|\delta v| = O((B-1)^{-1}), \\ \nabla u_j^a &= (\nabla a_j - \nabla\mu_a) f(v_a) + b_j^a f'(v_a)\nabla v_a. \end{aligned} \tag{12}$$

Subtracting the two gradient expressions and applying the triangle inequality to each product gives

$$\begin{aligned} \|\delta\nabla u_j\| \leq{}& \epsilon^{-1/2}\|\delta\nabla\mu\| + L\epsilon^{-3/2}|\delta v| + 2ML\epsilon^{-3/2}|\delta\mu| \\ &+ 6M^2 L\epsilon^{-5/2}|\delta v| + M\epsilon^{-3/2}\|\delta\nabla v\| = O((B-1)^{-1}). \end{aligned} \tag{13}$$

The bounded affine transform $h_j^a = \gamma u_j^a + \beta$ preserves both bounds. Write the unaveraged receiver loss as $\ell_j^a = \psi_j(\theta, h_j^a)$. The total gradient includes the downstream dependence on θ:

$$g_j^a = \partial_\theta\psi_j(\theta, h_j^a) + \partial_h\psi_j(\theta, h_j^a)\nabla h_j^a. \tag{14}$$

The partial derivative ∂_θ holds h fixed. Let L_θ and L_h bound the Lipschitz constants of the two partial derivatives as functions of h. Let C_ψ bound $|\partial_h\psi_j|$ and Q bound $\|\nabla h_j^0\|$. The chain rule then gives

$$\begin{aligned} |\ell_j^1 - \ell_j^0| &\leq C_\psi|\delta h_j| = O((B-1)^{-1}), \\ \|g_j^1 - g_j^0\| &\leq (L_\theta + L_h Q)|\delta h_j| + C_\psi\|\delta\nabla h_j\| \leq \frac{C_g}{B-1}. \end{aligned} \tag{15}$$

The constant C_g is independent of B and K.

3. Averaging over K affected receivers. Let $\mathcal{R}_K$ contain the K peers assigned deletion statistics. The source streams are independent of receiver selection, so receivers outside $\mathcal{R}_K$ retain their losses and gradients. The mean-gradient change is

$$\Delta\bar{g}_{SP,K} = \frac{1}{B}\sum_{j\in\mathcal{R}_K}(g_j^1 - g_j^0), \qquad \|\Delta\bar{g}_{SP,K}\| \le \frac{C_g K}{B(B-1)} = O(K/B^2). \tag{16}$$

Since $B-1 \ge B/2$, the bound holds uniformly for $0 \le K \le B-1$. Removing one bounded gradient term and renormalizing the remaining loss, with normalization statistics fixed, gives

$$\bar{g}_{-i} - \bar{g} = \frac{\bar{g} - g_i}{B-1}, \qquad \|\bar{g}_{-i} - \bar{g}\| \le \frac{2G}{B-1} \quad \text{if } \|g_j\| \le G. \tag{17}$$

When $K = B-1$, the peer and direct changes have bounds of the same order, $O(1/B)$.

Extension to the routed network. At deeper layers, the retained source activations can also change. Let e_ℓ bound their activation and parameter-Jacobian differences at layer ℓ. The next moment change has two parts: deletion from a fixed source batch and perturbation of its retained activations. The first is $O((B-1)^{-1})$ by the calculation above, and the second is $O(e_\ell)$ by boundedness and local smoothness. The normalized source streams therefore satisfy $e_{\ell+1} \le c_\ell e_\ell + d_\ell/(B-1)$, starting from $e_0 = 0$. At fixed depth, $e_\ell = O((B-1)^{-1})$. Applying the same argument to each selected receiver preserves Equation 15, and hence Equation 16. The constants depend on the architecture and local derivative bounds.

For BatchNorm2d with a fixed number of spatial positions per image, first average activations and squared activations within each image. Deleting an image changes the averages of these per-image moments by the same $1/(B-1)$ factors. The argument applies componentwise, with constants allowed to depend on spatial size and channel count.

C Experimental Protocols

ResNet architecture. The CIFAR-style ResNet-18 uses a 3×3 stride-one stem, no max pool, eight residual blocks, and 20 BatchNorm layers, including three on projection shortcuts. The horizon, batch-size, training-phase, controlled ResNet, and utility studies share this architecture; their training settings are specified in the corresponding sections.

Experimental repetitions. FashionMNIST and controlled CIFAR-10 each use four training runs with different random seeds. The training-phase and batch-size studies share the same four runs. The CIFAR-100 WideResNet, horizon, and utility studies use eight, sixteen, and sixteen runs, respectively. The supplementary README lists the seed values and their correspondence to run numbers in the tables.

Table 7: Analysis units and repeated measurements. Prediction correlations are computed within the groups listed in the last column and averaged within each independent unit.

Study	Independent unit	Repeated measurements and matching
FashionMNIST and controlled ResNet	Training run	Focal examples within each run
Training phase	Paired training run	Focal examples within each checkpoint; paired phases share focal examples, later batches, and evaluation sets
Batch size and peer count	Checkpoint	Focal examples within each batch; nested curves retain focal and peer identities
Continuation horizon, standard baselines, and route partition	Checkpoint	Focal examples within each batch–continuation pair, matched across horizons and partitions

Continued on next page

Table 7: Analysis units and repeated measurements (continued).

Study	Independent unit	Repeated measurements and matching
CIFAR-100 WideResNet	Checkpoint	Focal examples within each continuation sequence
Numerical accuracy and perturbation	Checkpoint	Factual and deletion states for each focal example, horizon, and numerical condition
Training utility	New training run	Policies share the starting model, data, augmentations, and noise realizations

Normalization implementation. The six-horizon and partition studies use shared computations. The WideResNet, batch-size, training-phase, and normalization studies compute moments separately. The ResNet studies use BatchNorm momentum 0.1 and $\epsilon = 10^{-5}$.

Software environment. GPU computations use Python 3.8.20, PyTorch 2.0.1 with CUDA 11.8, and torchvision 0.15.2. Numerical precision and TF32 settings are specified with the corresponding experiments; the supplementary dependency files provide the Python 3.12 environment for recomputing the reported statistics on CPU.

C.1 Prediction metrics and permutation control

MAE averages absolute prediction errors over focal deletions within each run. Kendall τ_b is undefined for constant scores.

For deletion-prediction experiments, we form pointwise 95% percentile intervals from 10,000 bootstrap draws of the independent units in Table 7, preserving paired measurements. The batch-size bootstrap also resamples batches within checkpoints while retaining each batch's focal-example curves.

The permutation control reassigns the $SF + SP$ score increments among focal examples within each matched set, then adds them to the unchanged baseline scores.

C.2 Replay cost

For WideResNet, the runtime ratio divides the time for all 16 coalitions by the mean time of one factual and one ordinary deletion replay. We average these ratios over focal examples and continuation sequences within each checkpoint.

Table 8: Runtime for a complete coalition game and peak allocated GPU memory. Time is seconds per game, reported as the mean [minimum, maximum] of run-level averages.

Design	Seconds/game	Peak memory (GiB)	Replay-time ratio
FashionMNIST trajectory	11.29 $[11.06, 11.52]$	0.12	–
Controlled ResNet-18	3.13 $[2.98, 3.29]$	1.57	–
ResNet-18, epoch 30	2.98 $[2.89, 3.04]$	1.52	–
ResNet-18, epoch 60	3.05 $[3.02, 3.07]$	1.56	–
CIFAR-100 WideResNet	22.27 $[22.14, 22.41]$	13.21	$19.78\times$

D CIFAR-10 Prediction across Six Continuation Horizons

D.1 Configuration

The horizon benchmark uses the full-data ResNet-18 training configuration in Appendix I, with checkpoints saved at epoch 30. The three probe batches each contain 128 examples and are disjoint

within a checkpoint. The first two continuation updates are fixed; later updates follow two independently generated orders. Computation uses deterministic FP32, with TF32 enabled for cuDNN convolutions and disabled for matrix multiplication.

D.2 Prediction at every endpoint

Immediately after the intervention update, adding $SF+SP$ to D lowers MAE in 14 checkpoints and improves ordering in all 16. Both gains are positive in every checkpoint from two updates onward (Table 9).

Table 9: Prediction across six continuation horizons. (a) Mean MAE multiplied by 10^3. (b) Mean Kendall τ_b. Gains compare $D + SF + SP$ with D, with pointwise 95% CIs.

(a) Prediction error

Updates	D	$D + SF + SP$	Full	Zero	MAE gain [95% CI]
0	0.247	0.185	0.075	0.379	$0.062\ [0.033, 0.100]$
2	0.744	0.257	0.196	1.05	$0.487\ [0.420, 0.547]$
10	1.79	0.490	0.482	2.35	$1.30\ [1.01, 1.63]$
34	3.64	1.33	1.33	4.78	$2.31\ [1.81, 2.83]$
130	14.9	5.36	5.36	18.8	$9.49\ [7.47, 11.85]$
393	31.9	10.6	10.6	41.8	$21.3\ [17.7, 25.1]$

(b) Deletion ordering

Updates	D	$D + SF + SP$	Full	τ_b gain [95% CI]
0	0.321	0.630	0.884	$0.308\ [0.252, 0.371]$
2	0.345	0.838	0.874	$0.493\ [0.415, 0.567]$
10	0.298	0.856	0.861	$0.557\ [0.506, 0.615]$
34	0.390	0.817	0.816	$0.427\ [0.356, 0.496]$
130	0.374	0.789	0.789	$0.414\ [0.356, 0.478]$
393	0.391	0.822	0.822	$0.431\ [0.374, 0.497]$

At 393 updates, the peer-statistic contribution remains comparable to loss participation: the mean checkpoint ratio of mean $|\phi_{SP}|$ to mean $|\phi_D|$ is $1.051\ [0.981, 1.120]$.

E Standard Training-Data Attribution Comparisons

E.1 Baseline definitions

TracInCP, last-layer IF, and TRAK use the factual trajectories, focal augmentations, and source evaluation set from Appendix D at 2, 34, and 393 updates.

All-parameter TracInCP. Following Pruthi et al. (2020), we sum learning-rate-weighted gradient inner products over saved checkpoints. Both the focal loss gradient and the mean source CE gradient use evaluation-mode BatchNorm and all trainable parameters, including the BatchNorm affine parameters. For horizon h, the score for deletion benefit is

$$s_{\mathrm{CP}}(q, h) = -\sum_{t \in \mathcal{C}_h} \eta_t \left\langle \nabla_\theta \ell_q(\theta_t), \nabla_\theta J_{E_{\mathrm{src}}}(\theta_t) \right\rangle .$$

The checkpoint sets are $\mathcal{C}_2 = \{0, 2\}$, $\mathcal{C}_{34} = \{0, 2, 10, 34\}$, and $\mathcal{C}_{393} = \{0, 2, 10, 34, 130, 393\}$; zero denotes the state just after the intervention update. The learning rate is 0.0525 at these checkpoints. The TracInCP variant with batch context computes the focal gradient in training mode within its 128-image factual batch; its source gradient remains in evaluation mode.

Damped last-layer influence functions. We apply the local influence approximation of Koh & Liang (2017) to the 5,130-parameter classification head at each factual endpoint, keeping the convolutional representation fixed. Let H be the Hessian of mean CE over all 50,000 training examples,

and let g_q and g_{src} be head gradients. The score is

$$s_{\mathrm{IF}}(q,h) = -\frac{1}{50{,}000}\, g_q^\top (H + 0.0105 I)^{-1} g_{\mathrm{src}}.$$

The diagonal term combines weight regularization 0.0005 and damping 0.01. CNN features use FP32; Hessian products and conjugate gradients use FP64. Conjugate gradients use a 250-iteration cap and relative residual tolerance 10^{-5}.

Regularized single-model TRAK. We use the classifier implementation in `traker` 0.3.2 (Park et al., 2023), with one model per factual endpoint ($M = 1$). Its output is the correct-class logit minus the log-sum-exp of the other logits. All trainable parameters contribute to evaluation-mode gradients. All 50,000 training examples and 5,000 source examples supply projected features; features for focal examples use their original batch augmentations. The Rademacher projection has 20,480 dimensions and a fixed seed. With feature matrix Φ, we set $A = \Phi^\top \Phi + \lambda I$, where λ is 0.001 times the mean Gram diagonal. Writing $Q_i = 1 - p_\theta(y_i \mid x_i)$ and $c = \mathrm{mean}\,|A^{-1}|$, the score for deletion benefit is

$$s_{\mathrm{TRAK}}(q,h) = -Q_q \Phi_q A^{-1} \overline{\Phi}_{\mathrm{src}}^\top / c.$$

Projection uses FP16 gradient inputs and FP32 accumulation. The Gram matrix uses FP64 accumulation and symmetric FP32 storage; the linear system is solved in FP64. The source-CE variant replaces $\overline{\Phi}_{\mathrm{src}}$ by $\mathrm{mean}_{i \in E_{\mathrm{src}}}(Q_i \Phi_i)$, using $\partial \mathrm{CE}/\partial f = -Q$.

E.2 Evaluation and statistical analysis

We evaluate Kendall τ_b and sign agreement before calibration, using ordinary deletion benefit as the target and the paired bootstrap in Appendix C.

Paired tests enumerate all 2^{16} sign flips of model-level gains. Holm correction covers the 18 ordering comparisons between two EGR scores (D and $D + SF + SP$) and three baselines at three horizons. Calibrated MAE uses a separate family of the same 18 comparisons.

Cross-validated CE calibration. At each horizon, the leave-one-model-out calibration uses 720 score–target pairs for fitting and 48 pairs for evaluation. For training-fold score x and target y, the slope is $a = \max\{0, \mathrm{Cov}(x,y)/[\mathrm{Var}(x)(1 + 10^{-6})]\}$ and the intercept is $b = \bar{y} - a\bar{x}$. A zero-variance score uses $a = 0$, giving the training-fold mean. We bootstrap model-level errors with calibration fits fixed.

E.3 Deletion ordering and calibrated error

The $D + SF + SP$ score improves deletion ordering over all three baselines at every horizon. At 393 updates, its gains over TracInCP, IF, and TRAK are 0.826 $[0.758, 0.897]$, 0.799 $[0.741, 0.867]$, and 0.804 $[0.782, 0.827]$, respectively. Across the three baselines and three horizons, the adjusted p-value is 5.49×10^{-4} for each comparison with $D + SF + SP$ and at most 1.282×10^{-3} for comparisons with D.

Table 10: Deletion ordering before calibration, including two baseline variants. Entries are mean Kendall τ_b [95% bootstrap interval].

Method	2 updates	34 updates	393 updates
TracInCP	$0.164\ [0.070, 0.256]$	$-0.005\ [-0.065, 0.058]$	$-0.004\ [-0.072, 0.062]$
Last-layer IF	$0.121\ [0.027, 0.213]$	$-0.010\ [-0.077, 0.048]$	$0.023\ [-0.050, 0.080]$
Single-model TRAK	$0.126\ [0.046, 0.204]$	$0.013\ [-0.037, 0.063]$	$0.019\ [-0.011, 0.047]$
EGR: D	$0.345\ [0.272, 0.421]$	$0.390\ [0.321, 0.461]$	$0.391\ [0.326, 0.450]$
EGR: $D + SF + SP$	$0.838\ [0.796, 0.876]$	$0.817\ [0.784, 0.847]$	$0.822\ [0.801, 0.844]$
Full EGR	$0.874\ [0.829, 0.915]$	$0.816\ [0.783, 0.847]$	$0.822\ [0.801, 0.844]$
Zero	–	–	–
TracInCP with batch context	$0.201\ [0.124, 0.274]$	$-0.028\ [-0.085, 0.024]$	$0.010\ [-0.046, 0.066]$
TRAK with source CE	$0.064\ [-0.004, 0.134]$	$0.017\ [-0.048, 0.078]$	$0.012\ [-0.041, 0.063]$

After calibration, both D and $D+SF+SP$ achieve lower MAE than all three baselines at every horizon (adjusted $p \leq 1.099 \times 10^{-3}$; Table 11). At 393 updates, the MAE intervals are $[0.0100, 0.0112]$ for $D+SF+SP$, $[0.0369, 0.0478]$ for TracInCP and IF, and $[0.0370, 0.0479]$ for TRAK. Full EGR achieves the lowest calibrated error at two updates.

Table 11: Sign agreement before calibration and leave-one-model-out calibrated MAE. Sign agreement is a percentage; MAE is multiplied by 10^3. Entries average model-level metrics.

	Sign agreement ↑			Calibrated MAE ↓		
Method	2	34	393	2	34	393
TracInCP	58.59	54.56	44.92	1.04	4.79	41.9
Last-layer IF	51.56	46.61	49.35	1.05	4.80	41.9
Single-model TRAK	56.51	51.04	49.61	1.05	4.80	41.9
EGR: D	62.50	69.66	73.18	0.745	3.65	32.2
EGR: $D + SF + SP$	89.32	89.45	91.54	0.257	1.34	10.6
Full EGR	91.41	89.45	91.54	0.199	1.34	10.6
Zero / calibrated mean	0.00	0.00	0.00	1.05	4.79	41.9
TracInCP with batch context	59.90	52.08	45.57	0.939	4.79	41.9
TRAK with source CE	51.82	49.48	51.95	1.05	4.80	41.8

F Route Granularity and Interaction Allocation

F.1 Games and allocation metrics

The merged game has players $\{D, BN, PB\}$, with eight coalition values inherited from the original game. The layer-split game has players $\{D, SF, SP_{\text{early}}, SP_{\text{late}}, PB\}$ and 32 coalitions. Switching both peer subroutes together recovers the original 16 values.

The combined statistic contributions are $\phi_{SF} + \phi_{SP}$, ϕ_{BN}, and $\phi_{SF} + \phi_{SP_{\text{early}}} + \phi_{SP_{\text{late}}}$ for the original, merged, and split games. We compute magnitude ratios within each checkpoint and average them across checkpoints. For each partition, the numerator is the mean absolute value of its combined statistic contribution, and the denominator is mean $|\phi_D|$ recomputed for that partition. We normalize the mean absolute change in ϕ_D by the original game's mean $|\phi_D|$. Sign agreement and Kendall correlations compare contributions before and after regrouping.

F.2 WHY REGROUPING CHANGES THE ALLOCATION

The interaction associated with coalition T is

$$\delta(T) = \sum_{U \subseteq T} (-1)^{|T|-|U|} v(U). \tag{18}$$

The Shapley contribution $\phi_g = \sum_{T \ni g} \delta(T)/|T|$ divides each interaction equally among its participating routes. Merging or splitting routes changes the number of participants and hence their shares.

F.3 MAGNITUDE, SIGNS, AND PREDICTION

Regrouping routes changes both the magnitude and sign of their allocations. At 393 updates, the target-set magnitude ratios of statistics to D are 1.225, 1.008, and 1.687 for the original, merged, and split games. On the source set, the combined early- and late-layer peer contribution has a ratio of 1.506 to D.

Merging and splitting change the sign of D in 10.55% and 15.36% of source-set cases, respectively. Kendall rank correlations with the original signed D contributions are 0.788 and 0.698; correlations between their magnitudes are 0.638 and 0.494.

Table 12: Prediction gains under all three partitions. Each entry gives MAE reduction (multiplied by 10^3) / Kendall τ_b gain from adding the sum of statistic-route contributions to that partition's recomputed D.

Updates	Four routes	Merged statistics	Layer-split peers
0	0.0616 / 0.308	0.0614 / 0.299	0.0621 / 0.317
2	0.487 / 0.493	0.487 / 0.503	0.476 / 0.512
10	1.30 / 0.557	1.28 / 0.579	1.30 / 0.554
34	2.31 / 0.427	2.18 / 0.441	2.51 / 0.456
130	9.49 / 0.414	9.13 / 0.424	10.4 / 0.461
393	21.3 / 0.431	20.2 / 0.463	23.0 / 0.492

F.4 SINGLE-ROUTE AND LEAVE-ONE-ROUTE-OUT COMPARISONS

The single-route effect $v(\{g\}) - v(\varnothing)$ holds all other routes at their factual settings; the leave-one-route-out effect $v(\mathcal{P}) - v(\mathcal{P} \setminus \{g\})$ holds them at their deletion settings. For loss participation, both conditional effects remain unchanged across partitions while the Shapley allocation changes (Table 13).

Table 13: Mean absolute D effects at 393 updates on the source set.

Partition	Single-route	Leave-one-route-out	Shapley
Four routes	0.0413	0.0408	0.0252
Merged statistics	0.0413	0.0408	0.0291
Layer-split peers	0.0413	0.0408	0.0197

G CIFAR-100 WIDERESNET REPLICATION

G.1 DESIGN

We train WideResNet-28-10 on the complete CIFAR-100 training split. The model has 36.5 million parameters and 25 BatchNorm layers. Training uses random crops, horizontal flips, batches of 128,

SGD with Nesterov momentum 0.9, weight decay 5×10^{-4}, and a cosine schedule from learning rate 0.1 with $T_{\max} = 200$. We save eight epoch-100 checkpoints and use probe learning rate 0.05.

Each checkpoint supplies eight distinct focal examples sampled from one 128-example update. Each deletion is replayed along two independent sequences of two 32-example updates.

G.2 Statistic contributions improve prediction

Adding SF and SP reduces MAE by 0.00112 $[7.1 \times 10^{-4}, 0.00151]$ and increases Kendall τ_b by 0.531 $[0.357, 0.696]$ (Table 14).

Table 14: CIFAR-100 WideResNet prediction error and deletion ordering at each checkpoint.

(a) Prediction error (MAE)

Run	Test accuracy	D	$D + SF + SP$	Full
1	64.43%	0.00315	0.00108	0.00122
2	64.77%	0.00178	8.37×10^{-4}	7.73×10^{-4}
3	67.98%	0.00120	0.00118	0.00106
4	66.09%	0.00243	9.83×10^{-4}	8.59×10^{-4}
5	65.94%	0.00304	0.00208	0.00213
6	65.81%	0.00294	0.00144	0.00135
7	65.91%	0.00392	0.00323	0.00305
8	67.29%	0.00302	0.00166	0.00144

(b) Deletion ordering (Kendall τ_b)

Run	D	$D + SF + SP$	Full
1	0.571	0.893	0.893
2	0.036	0.750	0.857
3	0.429	0.536	0.643
4	0.071	0.929	0.857
5	0.036	0.750	0.643
6	0.286	0.571	0.500
7	0.143	0.821	0.750
8	0.179	0.750	0.821

H Batch Size and the Number of Affected Peers

H.1 Configuration

Varying batch size tests how averaging changes the response of a fixed group of peers. Varying the affected-peer count at a fixed batch size then tests how those responses accumulate in the gradient and subsequent training. We use the four epoch-30 checkpoints from Appendix I. At each checkpoint, we sample three new 128-example probe batches, excluding the exploratory batch, and pair each with two 32-example continuation batches. The resulting sets of 192 examples are disjoint within each checkpoint. Each probe supplies nested batches with $B \in \{8, 16, 32, 64, 128\}$ and the same eight focal positions.

The peer-count intervention uses $K \in \{0, 1, 3, 7, 15, 31, 63, 127\}$ at $B = 128$, with all other routes factual and $K = 0$ as the reference.

H.2 Batch-size scaling

Loss-participation and peer-statistic contributions shrink at similar rates with batch size, with fitted log–log slopes of -1.746 and -1.782. Their magnitude ratio has slope -0.036 $[-0.301, 0.235]$ (Table 15). Increasing batch size attenuates both contributions while preserving a peer-statistic effect of comparable magnitude to loss participation. The fixed-peer loss curve isolates the per-peer response; the route magnitudes also include effects propagated through later updates.

H.3 Propagation to target loss

The target-loss difference from $K = 0$ grows with peer count, with a fitted log–log slope of 0.420 $[0.341, 0.534]$ (Table 16). The positive peer-count slopes for gradient distance, parameter distance, and target-loss change connect the normalization intervention to the final prediction task: exposing more peers to the changed statistics produces a larger effect after continued training.

Table 15: Batch-size and peer-count effects. Slopes are fitted on log–log scales. Gradient and parameter differences are ℓ_2 distances from the $K = 0$ branch, measured at the intervention update and after two continuation updates, respectively. Ratios compare these distances at $K = 127$ and $K = 1$ with $B = 128$.

Quantity	Estimate	95% CI	Direction across batches or examples
Loss-change slope for seven fixed peers vs B	-1.055	$[-1.230, -0.836]$	negative in 12/12 batches
Summed absolute loss-change slope vs K	1.143	$[0.894, 1.475]$	positive in 12/12 batches
Gradient distance slope vs K	0.753	$[0.530, 1.069]$	positive in 12/12 batches
Parameter distance slope vs K after two continuation updates	0.345	$[0.303, 0.401]$	positive in 12/12 batches
Gradient distance ratio	37.31	$[13.23, 211.84]$	$K = 127 > K = 1$ in 96/96 focal examples
Parameter distance ratio after two continuation updates	4.97	$[4.21, 6.25]$	$K = 127 > K = 1$ in 96/96 focal examples
Mean $\lvert\phi_D\rvert$ slope vs B	-1.746	$[-1.979, -1.517]$	–
Mean $\lvert\phi_{SP}\rvert$ slope vs B	-1.782	$[-1.994, -1.558]$	–
Mean $\lvert\phi_{SP}\rvert$/mean $\lvert\phi_D\rvert$ slope vs B	-0.036	$[-0.301, 0.235]$	–
Mean $\lvert\phi_{SP}\rvert$/mean $\lvert\phi_D\rvert$ at $B = 128$	0.971	$[0.609, 1.742]$	–

Table 16: Pooled peer-count curves at $B = 128$ over 96 focal deletions. Gradient and parameter differences use the $K = 0$ branch as the reference; parameters are measured after two continuation updates.

K	Summed absolute loss change	Gradient difference ℓ_2	Parameter difference ℓ_2	Target loss $\lvert\Delta J_{K-0}\rvert$
1	0.00287	0.00298	0.00397	8.53×10^{-5}
3	0.00937	0.00787	0.00475	1.51×10^{-4}
7	0.0348	0.0203	0.00664	1.87×10^{-4}
15	0.0782	0.0333	0.00839	2.12×10^{-4}
31	0.181	0.0561	0.0117	3.42×10^{-4}
63	0.339	0.0790	0.0151	4.91×10^{-4}
127	0.669	0.111	0.0198	7.10×10^{-4}

H.4 Deletion prediction across batch sizes

Adding statistics reduces MAE at every batch size, by 92.4% at $B = 8$ and 49.7% at $B = 128$; all 95% intervals for the reductions exclude zero. The resulting Kendall τ_b ranges from 0.780 to 0.887 (Table 17).

I How Route Contributions Change during Training

I.1 Configuration

We train the CIFAR-style ResNet-18 in Appendix C on all 50,000 CIFAR-10 training images for 60 epochs. Training uses batches of 128, random crops with four-pixel padding, horizontal flips, SGD with momentum 0.9 and weight decay 5×10^{-4}, and a cosine learning-rate schedule from 0.1 to 0.005 over 60 epochs. We compare epoch-30 and epoch-60 checkpoints from the same four runs, with two unaugmented continuation batches per probe. Matching examples and continuations

Table 17: Batch-size effects on (a) mean absolute loss changes among seven fixed peers and relative route magnitude, and (b) deletion prediction. MAE reduction compares $D + SF + SP$ with D.

(a) Peer loss changes and route contributions

B	Fixed-peer loss change	Mean $\lvert\phi_{SP}\rvert$/mean $\lvert\phi_D\rvert$
8	0.0916	1.064
16	0.0477	0.830
32	0.0189	0.867
64	0.0109	0.776
128	0.00496	0.971

(b) Deletion prediction

	MAE			Kendall τ_b	
B	D	$D + SF + SP$	Reduction	D	$D + SF + SP$
8	0.110	0.00843	92.4%	0.714	0.857
16	0.0175	0.00368	79.0%	0.536	0.845
32	0.00452	0.00141	68.8%	0.411	0.887
64	0.00146	5.98×10^{-4}	59.2%	0.381	0.815
128	6.85×10^{-4}	3.44×10^{-4}	49.7%	0.440	0.780

across phases lets us distinguish a change in the relative contribution of each route from the decline in absolute deletion effects as training progresses.

Each checkpoint supplies one fixed 32-example probe update and 16 focal deletions. At epochs 30 and 60, the probe learning rates are 0.0525 and 0.005, and target-set accuracies are 85.20–87.32% and 94.30–94.86%.

I.2 Peer and buffer contributions decay more slowly

Peer and buffer contributions decrease less than loss participation between epochs 30 and 60 (Table 19). The ratio of mean $\lvert\phi_{PB}\rvert$ to mean $\lvert\phi_D\rvert$ rises from 0.12 to 3.56, and the combined share of SP and PB in mean absolute allocation rises from 48% to 89% (Figure 4). Prediction gains become less consistent: adding $SF + SP$ to D improves ordering in all four runs at epoch 30 and in two at epoch 60 (Table 18).

Table 18: Full-data ResNet at epoch 60: per-run prediction and ranking.

(a) Prediction error (MAE)

Run	D	$D + SF + SP$	Full
1	3.39×10^{-4}	4.38×10^{-4}	4.20×10^{-4}
2	4.66×10^{-4}	2.58×10^{-4}	1.96×10^{-4}
3	1.15×10^{-4}	1.23×10^{-4}	6.04×10^{-5}
4	1.84×10^{-4}	2.70×10^{-4}	2.43×10^{-4}

(b) Deletion ordering (Kendall τ_b)

Run	D	$D + SF + SP$	Full
1	0.033	-0.167	-0.217
2	-0.033	0.700	0.783
3	-0.267	-0.250	0.683
4	0.367	0.100	0.217

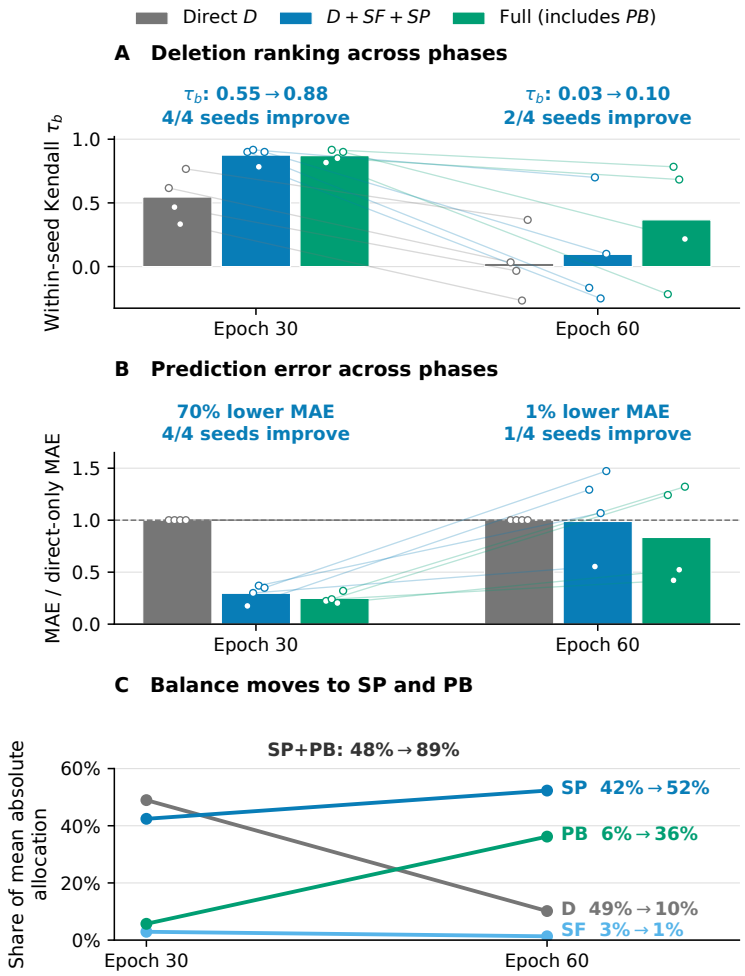


Figure 4: Changes between epochs 30 and 60. (A–B) Prediction gains from adding $SF + SP$ to D. MAE is normalized by the corresponding D-score MAE: pooled values for bars and run-level values for dots. Lines connect matched runs. (C) Each route's share of mean absolute allocation, pooled across runs at each epoch; the annotation gives the combined $SP + PB$ share.

Table 19: Full-data ResNet: mean absolute route allocations across training phases.

Route	Epoch 30	Epoch 60	Fold decrease
Loss participation	0.00500	3.32×10^{-5}	150.7×
Focal statistics	2.98×10^{-4}	4.36×10^{-6}	68.3×
Peer statistics	0.00434	1.71×10^{-4}	25.4×
Running buffer	5.83×10^{-4}	1.18×10^{-4}	4.93×

J Trajectory Deletion and the GroupNorm Control

J.1 Configuration

Deleting before batch construction changes both the focal update and the batches that follow it. This study tests the resulting later-membership contribution, while GroupNorm provides a control in which normalization acts within each example. The MLP has two hidden layers of widths 64

and 32, with BatchNorm after each hidden layer. It trains on 1,024 examples with batch size 16, SGD learning rate 0.05, no optimizer momentum, and BatchNorm momentum 0.1. After three warm-up epochs, the trajectory replay follows three fixed epoch orders for each run. We select 32 class-balanced focal examples per run, each occurring before the final batch of every replay epoch.

The source set contains 5,000 class-balanced training examples outside the 1,024-example training subset; the target is the complete 10,000-example test set. We compare BatchNorm with matched GroupNorm models.

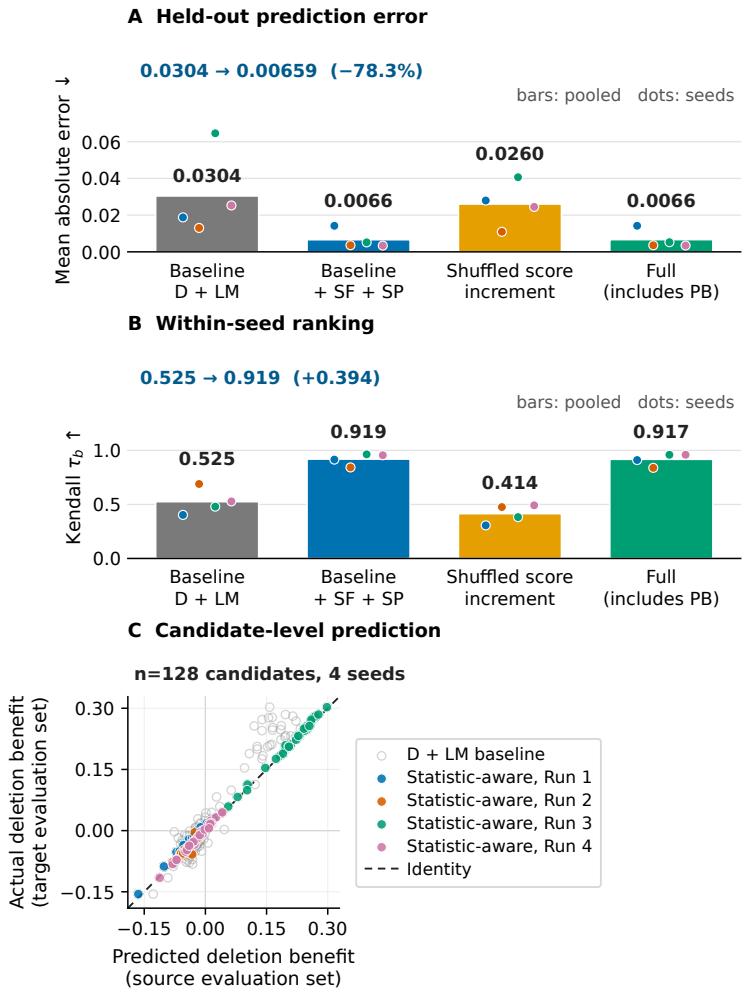


Figure 5: FashionMNIST trajectory deletion. (A–B) Prediction error and ordering across four runs. (C) Predicted versus observed deletion benefit; the dashed line marks equality. Bars average runs; colored dots identify runs in A–B and focal deletions in C.

J.2 Prediction results

With BatchNorm, adding $SF + SP$ to $D + LM$ lowers MAE and increases Kendall τ_b in all four runs (Figure 5). In the matched GroupNorm models, the SF, SP, and PB contributions are exactly zero for every focal deletion. Loss participation and later membership remain active, separating the effect of changing the training sequence from the effect of sharing normalization statistics.

K Controlled ResNet and Frozen-BatchNorm Comparisons

K.1 Configuration

The ResNet-18 in Appendix C trains on a class-balanced subset of 5,000 examples for 20 warm-up epochs. Training uses batch size 32, SGD learning rate 0.05, momentum 0.9, weight decay 0.0005, and BatchNorm momentum 0.1. Images receive fixed per-channel normalization and no data augmentation.

Each checkpoint supplies one 32-example probe batch, 16 focal deletions, and two continuation batches. We compare BatchNorm with frozen BatchNorm to distinguish the influence of current-batch statistics and running-buffer updates from the loss-driven parameter update. Frozen BatchNorm uses the saved running statistics throughout replay.

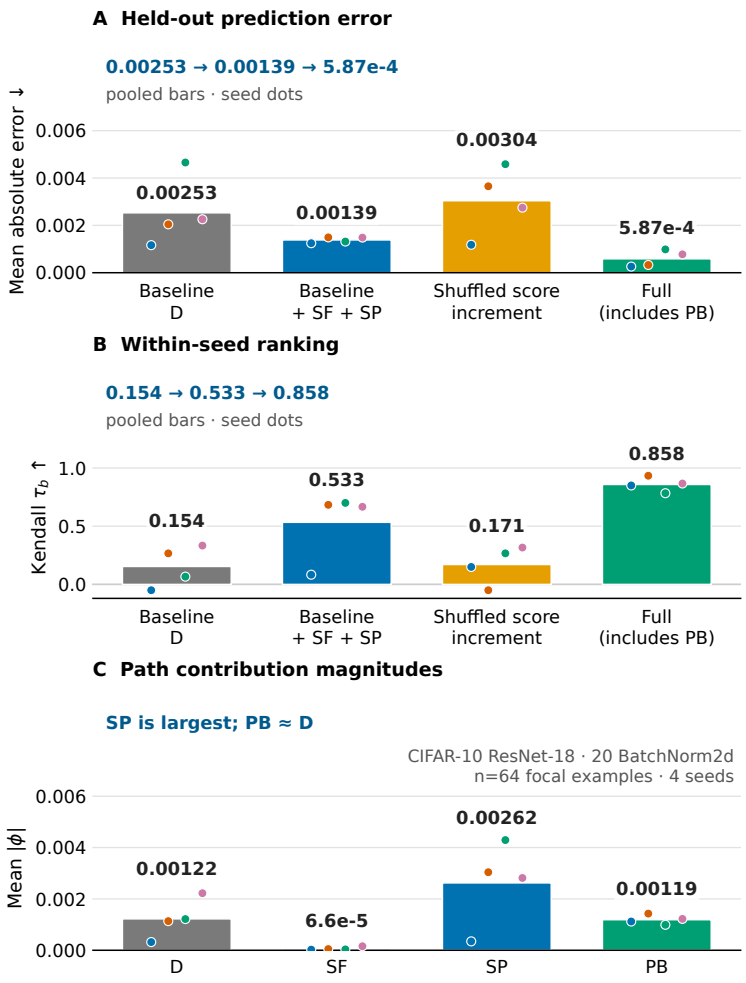


Figure 6: Controlled CIFAR-10 ResNet-18. Panels A–B compare prediction error and ordering; panel C shows mean absolute route allocations. Bars average runs; dots show run-level values.

K.2 Buffers improve prediction beyond forward statistics

Adding $SF + SP$ to D increases Kendall τ_b in all four runs and lowers MAE in three. Adding PB lowers MAE further in every run (Figure 6). The additional MAE reduction shows that the running buffers carry useful information about endpoint evaluation loss beyond the forward-statistic contributions. With frozen BatchNorm, all coalitions with the same D setting have identical losses, and the SF, SP, and PB contributions are exactly zero for every focal deletion.

L TRAINING UTILITY UNDER NOISE AND EXPOSURE

L.1 TASK AND DATA

The official training set is partitioned into 40,000 pretraining images, 8,000 continuation images, and 2,000 candidates for source evaluation. For each model, we select a class-balanced source set of 200 clean images; its mean CE supplies the action-selection objective. The 8,000 target images in Section 5 exclude the two 1,000-image test subsets used in development.

The ResNet-18 in Appendix C is pretrained for 30 epochs with batch size 128, SGD learning rate 0.1, momentum 0.9, weight decay 5×10^{-4}, and a 60-epoch cosine schedule ending at 0.005. Inputs use random crops with four-pixel padding and random horizontal flips.

For each model, all policies start from the same pretrained parameters and BatchNorm buffers, with a fresh SGD optimizer at learning rate 0.001 and the pretraining momentum and weight decay. Replays and continuations use double precision.

The noise schedule uses shuffled blocks of 32 updates, each containing 16 noisy-image and 16 clean-image updates. Noise is independent $\mathcal{N}(0, 0.25^2)$ per pixel, followed by clipping to $[0, 1]$ and normalization; the seven peer images remain unchanged.

The exposure shift applies $x \mapsto \mathrm{clip}(0.65x + 0.30, 0, 1)$ before normalization. Secondary outcomes measure accuracy and CE on untransformed images at update 64 and on both image sets after 32 ordinary updates without added noise.

L.2 ACTION SELECTION

For each focal image, let $L(S)$ be the mean CE on the clean source set after one update using coalition $S \subseteq \{D, SF, SP, PB\}$. We compute the Shapley contributions from the 16 coalition losses, normalize benefits as $b_g = -\phi_g / \max(L(\varnothing), 10^{-12})$, and use threshold $\epsilon_{\mathrm{act}} = 10^{-6}$. The proposed route set includes D when $b_D > \epsilon_{\mathrm{act}}$. It includes SF, SP, PB when $b_{SF} + b_{SP} > \epsilon_{\mathrm{act}}$; otherwise, it includes PB when $b_{PB} > \epsilon_{\mathrm{act}}$.

An action in Table 20 is eligible when its route set is contained in the proposal. The policy applies the eligible action with the lowest $L(S)$ when $L(\varnothing) - L(S) > \epsilon_{\mathrm{act}} \max(L(\varnothing), 10^{-12})$; otherwise it keeps the example. Ties follow the table order.

Loss masking averages over the seven retained losses while retaining the focal image as a statistic source. Full deletion uses seven receivers and their statistics. Buffer exclusion uses running-buffer updates from a separate forward pass that excludes the focal image and precedes the parameter update.

Table 20: Training actions and their corresponding route choices.

Action	Routes set to deletion
Keep	$\varnothing$
Mask focal loss	$\{D\}$
Exclude focal image from running-statistic updates	$\{PB\}$
Mask loss and exclude focal image from running-statistic updates	$\{D, PB\}$
Exclude focal image from batch and running statistics	$\{SF, SP, PB\}$
Delete focal image	$\{D, SF, SP, PB\}$

L.3 COMPARISON METHODS

The D-only (direct-route) policy selects loss masking using the D benefit from the full Shapley allocation. Total-effect deletion selects between deleting and keeping the example by comparing their source losses.

Direct search minimizes source loss over all six actions in Table 20 or over the first four D/PB actions. The restricted EGR policy applies the D and PB proposal tests to those four actions.

Fixed policies apply the same operation at every update, including holding the BatchNorm buffers constant. The variance-based baselines mask or delete the focal image when its mean within-channel spatial variance in normalized coordinates is below 10^{-12}. The roughness-based baselines act when the mean squared difference between horizontally or vertically adjacent pixels, in $[0, 1]$ coordinates, exceeds 0.03125.

Margin variants replace source CE with the mean of $\mathrm{sigmoid}[-(z_y - \max_{k \neq y} z_k)]$, where z_k is the logit for class k and y is the true class.

Table 21: All 22 training policies. Accuracy and CE are measured under the exposure shift after 64 updates; runtime covers all 96 updates. Entries are means across models.

Method	Accuracy (%)	CE	Runtime (s)
EGR-guided policy	84.98	0.4803	418.28
EGR-guided policy (margin)	84.89	0.4833	418.69
Ordinary training	84.42	0.4968	6.63
Always mask focal loss	84.23	0.5009	6.71
High-loss filtering	84.15	0.5024	9.04
D-only policy	84.45	0.4930	417.48
Total-effect deletion	84.85	0.4790	57.87
Direct search (six actions)	85.00	0.4780	123.70
Direct search (four actions)	85.01	0.4761	87.06
EGR policy (D, PB actions)	85.09	0.4739	418.72
Always exclude focal statistics	85.05	0.4843	6.94
Always delete focal example	84.92	0.4797	6.44
Always hold BN buffers	81.78	0.5666	6.63
Variance-triggered loss mask	84.42	0.4968	5.68
Variance-triggered deletion	84.42	0.4968	5.68
D-only policy (margin)	84.43	0.4952	417.90
Total-effect deletion (margin)	84.88	0.4799	57.94
Direct search (six actions, margin)	84.93	0.4830	123.84
Direct search (four actions, margin)	84.93	0.4794	87.18
EGR policy (D, PB actions, margin)	84.89	0.4811	419.03
Roughness-triggered loss mask	84.27	0.4989	5.74
Roughness-triggered deletion	84.98	0.4753	5.60

L.4 Statistical analysis

Each model contributes one equally weighted paired observation. The two primary accuracy comparisons share significance level $\alpha_1 = 0.025$ from the sequential schedule $\alpha_j = 0.05/2^j$. Each interval is $\bar{d} \pm t_{15,1-0.025/4} s_d/\sqrt{16}$, where d is the model-level accuracy difference in percentage points. Secondary comparisons against 21 methods use $t_{15,1-0.025/42}$.

L.5 Accuracy and computational cost

Four-action direct search reaches 85.01% mean accuracy with a runtime of 87.06 seconds per model, compared with 84.98% and 418.28 seconds for the full EGR policy (Table 21). Runtime covers action selection, all 96 training updates, and amortized setup and execution overhead; shared pretraining and target evaluation are excluded.

L.6 Actions selected during training

The action frequencies in Table 23 show how each policy uses the available interventions. Full EGR selects all six actions: loss masking is the most frequent at 23.4%, and full deletion accounts for 11.8%. The restricted EGR policy excludes the focal image from running-statistic updates in 55.4% of decisions, either alone or together with loss masking; four-action direct search does so in

Table 22: Primary and secondary utility endpoints. (a) EGR accuracy and its gains over ordinary training and high-loss filtering, in percentage points. (b) Mean target CE for the same three policies; lower is better. Ordinary continuation consists of 32 updates after the intervention period.

(a) Accuracy

Endpoint	View	EGR (%)	Δ ordinary	Δ high-loss
After 64 updates	Exposure	84.98	+0.56	+0.84
After 64 updates	Clean	89.03	+0.08	+0.05
After ordinary continuation	Exposure	85.37	+0.16	+0.11
After ordinary continuation	Clean	89.11	+0.07	−0.07

(b) Cross-entropy

Endpoint	View	EGR	Ordinary	High-loss
After 64 updates	Exposure	0.4803	0.4968	0.5024
After 64 updates	Clean	0.3442	0.3500	0.3470
After ordinary continuation	Exposure	0.4646	0.4681	0.4634
After ordinary continuation	Clean	0.3391	0.3402	0.3380

55.0%. Both policies therefore use buffer exclusion frequently while retaining the image as a source of forward-pass batch statistics.

Table 23: Selected actions over 64 intervention updates in each of 16 models, giving 1,024 decisions per policy. Entries are percentages. Restricted EGR and direct search use the four D, PB actions; dashes mark unavailable actions.

Action	Full EGR	Restricted EGR	Direct search
Keep	18.5	21.2	18.6
Mask focal loss	23.4	23.4	26.5
Exclude focal image from running-statistic updates	13.3	27.1	23.3
Mask loss and exclude focal image from running-statistic updates	18.6	28.3	31.6
Exclude focal image from batch and running statistics	14.5	–	–
Delete focal image	11.8	–	–

M Extended Related Work

Normalization statistics. Normalization methods differ in which examples and training states supply their statistics. Reference batches, population corrections, and historical estimates alter this source (Salimans et al., 2016; Ioffe, 2017; Kou et al., 2020; Wu & Johnson, 2021; Yao et al., 2021); layer, instance, group, filter-response, and proxy normalization avoid pooling across the current batch (Ba et al., 2016; Ulyanov et al., 2017; Wu & He, 2018; Singh & Krishnan, 2020; Labatie et al., 2021). Within BatchNorm, corrections address backward-pass statistics, an example's influence on its own normalization, and estimation noise (Yan et al., 2020; Kosson et al., 2024; Singh & Shrivastava, 2019). Shuffling addresses information leakage across examples (Garov et al., 2024; He et al., 2020), and data-stream composition affects statistical bias (Pham et al., 2022; Lyu et al., 2023). Analyses of BatchNorm objectives and atypical examples further connect these dependencies to learning (Lian & Liu, 2019; Doan et al., 2026). EGR measures how an example's participation in forward statistics and running-buffer updates contributes to its endpoint deletion effect. The GroupNorm and frozen-BatchNorm controls remove these dependencies and test the corresponding zero contributions.

Causal paths and coalition structure. Route attribution requires both an intervention on a computation and a rule for allocating interactions. Flow and causal-path methods define contributions through computational structure (Singal et al., 2021; Ter-Minassian et al., 2023); the choice of coalition game determines the resulting allocation (Sundararajan & Najmi, 2020). Sequential counterfactual decomposition follows effects along decision trajectories (Triantafyllou et al., 2025), and SMDA separates feature-activation and output-probability contributions to symbolic-policy updates through a first-order expansion (Habibi et al., 2026). EGR defines its interventions through replayed training operations and allocates their endpoint loss changes using Shapley values. The layer-partition comparison in Appendix F examines how grouping these operations changes the allocation while preserving the total deletion effect.

Influence approximations and training dynamics. Influence estimates depend on the response being predicted and the training dynamics represented by the approximation. Influence-function accuracy varies with architecture and regularization (Basu et al., 2021); the proximal Bregman response defines an alternative target (Bae et al., 2022), and VIF handles non-decomposable losses (Deng et al., 2025). Simfluence predicts losses under alternative training sequences (Guu et al., 2023), while MAGIC predicts data-addition and deletion effects through metadifferentiation (Ilyas & Engstrom, 2025). Analyses by individual update, model component, and training stage further resolve how predictions change during learning (Ren & Sutherland, 2025; Chen et al., 2026; Lee et al., 2026). EGR supplies a replay-based decomposition of the endpoint response. Its horizon comparisons measure numerical agreement with ordinary deletion separately from the transfer of deletion scores between evaluation sets.

Subset interventions and dynamic valuation. Data valuation can vary the unit of intervention and the treatment of training randomness. Randomized training-set interventions measure memorization and prediction changes (Feldman & Zhang, 2020; Lin et al., 2022), while subset selection captures group effects beyond individual scores (Hu et al., 2024). Shapley methods value federated contributors (Wang et al., 2020) and update valuations after data additions or deletions (Xia et al., 2025). Methods that represent training dynamics allocate contributions within one run (Wang et al., 2025a), separate instantaneous and propagated effects across federated updates (Zeng et al., 2026), or average over training randomness (Nguyen et al., 2023; Mlodozeniec et al., 2025). EGR takes one example's deletion as the intervention and uses its training routes as players. Matching the starting state and random choices across coalitions makes their differences attributable to the selected operations; replication across checkpoints and continuation sequences then measures variation in these effects.

Unlearning and replay state. Retraining after deletion provides a reference for machine unlearning (Cao & Yang, 2015; Deng et al., 2026). Provenance, cached gradients, sliced retraining, and saved updates reduce recomputation (Wu et al., 2020b;a; Bourtoule et al., 2021; Graves et al., 2021); checkpoint methods rewind training or estimate state corrections (Mu & Klabjan, 2025; Rezaei et al., 2026). The reference also depends on training order and restored state: unrolling identifies order-dependent errors (Thudi et al., 2022a), while replay definitions and deterministic checkpointing specify the state required for reproducible continuation (Thudi et al., 2022b; Xu et al., 2022). EGR uses paired factual and deletion replays to measure contributions to this endpoint difference. Including optimizer state, running buffers, and batch membership in the replay captures the routes by which a deletion changes subsequent training and evaluation.