KNOWLEDGE NETWORKS ENABLE STABLE AND EFFECTIVE ATTRIBUTION: DATA SHAPLEY VALUATION IN ONE TRAINING RUN

Anonymous authors
Paper under double-blind review

ABSTRACT

In-Run Data Shapley makes principled data valuation feasible for modern foundation models by estimating sample contributions along one observed training trajectory. However, existing in-run estimators mainly treat examples as independent contributors, making attribution scores unstable under noisy gradient signals and prone to favoring semantically redundant samples. A direct second-order treatment could model these interactions, but Hessian-dependent computation is costly and difficult to scale. To address these challenges, we propose Graph-Aware In-Run Data Shapley, a simple but effective approach that introduces a sample-level knowledge network into one-run valuation. The network connects semantically related examples and is used to smooth structurally supported contributions while penalizing redundant neighbors during selection. This design stabilizes attribution, improves diversity-aware data valuation, and replaces expensive Hessian-based interaction modeling with sparse neighborhood aggregation. Extensive experiments on image and language learning tasks show that the proposed graph-aware estimator improves attribution stability, subset selection, and computational efficiency over state-of-the-art approaches. Code is publicly available at `https://anonymous.4open.science/r/Shapley_code-7172/`.

1 INTRODUCTION

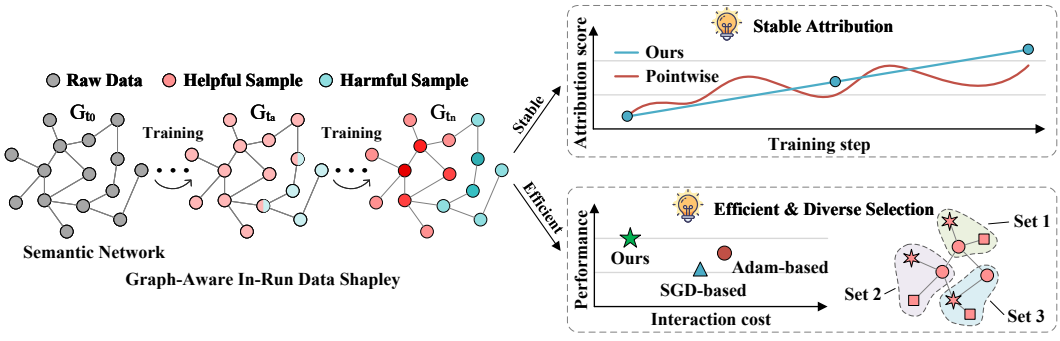

Figure 1: We propose a knowledge-network-based solution for Data Shapley valuation in one training run, i.e., Graph-Aware In-Run Data Shapley, which enables stable sample attribution and efficient diverse data selection compared to pointwise and interaction-heavy baseline methods.

Modern learning systems are increasingly limited not only by model design, but also by the quality and organization of their training data Paul et al. (2021); Sorscher et al. (2022); Xie et al. (2023); Lin et al. (2024). Large corpora often contain mislabeled, duplicated, noisy, or low-utility samples, and training on all available data can waste computation or even harm generalization Koh & Liang (2017); Toneva et al. (2019); Gao et al. (2020); Raffel et al. (2020); Abbas et al. (2023); Hoffmann et al. (2022). Data valuation therefore asks a central question in data-centric learning: *which training examples are truly useful for a given model and objective*?

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.

Existing data valuation methods have evolved along a clear computational trajectory. Early retraining-based methods, such as Data Shapley and datamodels, estimate sample utility by training models on different data subsets, providing principled attribution at substantial computational cost Ghorbani & Zou (2019); Ilyas et al. (2022). To avoid this repeated training, trained-model-based methods, including influence functions, TRAK, and recent LLM-scale attribution systems, infer sample effects from a fixed model through parameter, gradient, or prediction-level approximations Koh & Liang (2017); Park et al. (2023); Choe et al. (2025); Chang et al. (2025). More recently, In-Run Data Shapley methods push valuation inside the training process itself, using one observed optimization trajectory to accumulate sample contributions Wang et al. (2025); Ding et al. (2026). This makes Shapley-style attribution much more practical, but it also introduces a key limitation: *each sample is still mainly valued as an isolated point*.

This pointwise view causes two problems. First, the contribution of a sample can fluctuate across optimization steps, especially when gradients are noisy or the optimizer state changes Toneva et al. (2019); Paul et al. (2021). Second, valuation alone does not guarantee diversity: many near-duplicate or semantically similar samples may all receive high scores and dominate the selected subset Abbas et al. (2023); Zhang et al. (2025). Modeling such interactions with second-order information is possible in principle, but Hessian construction or repeated Hessian-vector computation is expensive Koh & Liang (2017); Choe et al. (2025); Zhang et al. (2025). Thus, an effective in-run valuation method should stabilize sample attribution and control redundancy without losing the efficiency of one-run estimation.

We propose *Graph-Aware In-Run Data Shapley* to address these problems. The main idea is to build a sample-level knowledge network from semantic similarity and use it as a structural prior for Data Shapley valuation, leveraging semantic structure to stabilize attribution and facilitate diverse data selection. Starting from optimizer-aware in-run contributions, the proposed method diffuses evidence among related samples to make supported scores more stable, and introduces a neighborhood-aware penalty to reduce redundant selections. Because the graph correction only requires sparse neighborhood aggregation, it provides a lightweight alternative to expensive Hessian-dependent interaction modeling. We evaluate the method on data attribution and data selection tasks in both visual and language learning settings. The results show that knowledge-network correction improves attribution stability, selected-subset quality, and computational efficiency compared with state-of-the-art baselines. These findings suggest that sample relations are not only useful side information, but an effective structure for making one-run Data Shapley valuation more stable and more actionable.

Our contributions are summarized as follows:

- We propose a graph-aware valuation framework that uses a sample-level knowledge network to stabilize attribution and improve diversity-aware data selection.
- We replace costly Hessian-dependent interaction modeling with sparse neighborhood aggregation, substantially reducing the complexity of sample-diversity correction.
- We validate the method on complementary attribution and selection tasks, showing significant improvements over state-of-the-art approaches.

2 Problem Formulation

Preliminary on Data Shapley. Given a training set $\mathcal{D} = \{z_i\}_{i=1}^n$, where $z_i = (x_i, y_i)$, a learning algorithm $\mathcal{A}$ maps a sample subset to model parameters. A validation set $\mathcal{D}_{val}$ defines the coalition utility $U(\mathcal{S}) \equiv U(\mathcal{A}(\mathcal{S}); \mathcal{D}_{val})$, where a larger value indicates better performance on the target objective. Data Shapley treats training examples as players in a cooperative game and assigns z_i the expected marginal contribution Ghorbani & Zou (2019):

$$\phi_i(U) = \mathbb{E}_\pi[U(S_i^\pi \cup \{z_i\}) - U(S_i^\pi)], \tag{1}$$

where π is sampled uniformly from the set Π of all permutations, and S_i^π contains the samples preceding z_i in π. Although Eq. (1) is principled, evaluating these coalitional utilities requires repeated model training.

Valuation Process in One Training Run. In-Run Data Shapley replaces repeated training with local valuation along an observed optimization trajectory Wang et al. (2025). At step t, a mini-batch

$\mathcal{B}_t \subseteq \mathcal{D}$ updates the model from θ_t to θ_{t+1} according to:

$$\theta_{t+1} = \theta_t - \eta_t \Delta_t(\mathcal{B}_t), \tag{2}$$

where η_t is the learning rate and $\Delta_t(\cdot)$ the optimizer-dependent update direction. For a subset $\mathcal{S} \subseteq \mathcal{B}_t$, its local utility is:

$$U^{(t)}(\mathcal{S}) = U\big(\theta_t - \eta_t \Delta_t(\mathcal{S})\big) - U(\theta_t), \tag{3}$$

let $c_i^{(t)} = \phi_i(U^{(t)})$ denote the local contribution of z_i, where $c_i^{(t)} = 0$ if $z_i \notin \mathcal{B}_t$. By the additivity of the Shapley value, its contribution over the complete training run is:

$$v_i = \sum_{t=0}^{T-1} c_i^{(t)}. \tag{4}$$

This formulation attributes a particular model trajectory rather than an abstract learning algorithm, but it still accumulates each sample independently.

Graph-aware Data Valuation. Training examples often exhibit semantic relations that are invisible to pointwise accumulation. We abstract these relations as a sample-level knowledge network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where node $v_i \in \mathcal{V}$ represents z_i and an edge connects semantically related samples. Given a representation $\mathbf{h}_i$, its adjacency matrix is defined as:

$$\mathbf{A}_{ij} = \begin{cases} 1, & z_j \in \mathrm{TopK}_{l \neq i} \, \mathrm{sim}(\mathbf{h}_i, \mathbf{h}_l), \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

$\mathbf{W}_{ij}$ records the normalized affinity on each retained edge. Given the local contributions $\{c_i^{(t)}\}$, the knowledge network $\mathcal{G}$, and a selection budget b, we formulate graph-aware in-run valuation as:

$$(\hat{\mathbf{v}}, \hat{\mathcal{S}}) = F\big(\{c_i^{(t)}\}_{i,t}, \mathcal{G}, b\big), \quad |\hat{\mathcal{S}}| = b, \tag{6}$$

where $\hat{\mathbf{v}}$ denotes structurally corrected attribution scores and $\hat{\mathcal{S}}$ is a high-value subset with controlled semantic redundancy.

Pointwise vs. Graph-aware In-Run Approaches. The core difference lies in how local evidence is accumulated. Pointwise approaches apply an independent temporal sum $v_i = \sum_t c_i^{(t)}$, so the estimate of z_i depends only on its own observed updates. In contrast, a graph-aware approach conditions each update on the evidence carried by its neighborhood:

$$\hat{v}_i = \sum_{t=0}^{T-1} \Gamma\big(c_i^{(t)}, \mathcal{H}_i^{(t)}\big), \quad \mathcal{H}_i^{(t)} = \{c_j^{(\tau)} : z_j \in \mathcal{N}(i), \tau \leq t\}, \tag{7}$$

where $\Gamma(\cdot)$ denotes a graph correction function. This structural distinction allows related samples to support reliable attribution and exposes redundancy among simultaneously valuable examples.

Main Challenges of Graph-aware In-Run Data Shapley:

(1) **Stable Attribution from Noisy Training Signals.** In-run contributions vary with model states, gradient scales, and optimizer moments. Neighboring samples provide corroborating evidence, but excessive smoothing can obscure genuine differences. The first challenge is to improve stability while preserving sample-specific utility.

(2) **Efficient Modeling of Sample Relations.** Sample relations are essential for interaction-aware valuation and diverse selection. Yet Hessian construction and repeated Hessian-vector products are difficult to scale. The second challenge is to capture useful interactions and suppress redundancy through sparse aggregation.

3 METHODOLOGY

We propose Graph-Aware In-Run Data Shapley, a framework explicitly designed to address the challenges above. As shown in Fig. 2, we introduce a **Knowledge Network-guided Attribution Stabilization** module to address Challenge (1), and design a **Diversity-aware Graph Valuation** module for Challenge (2).

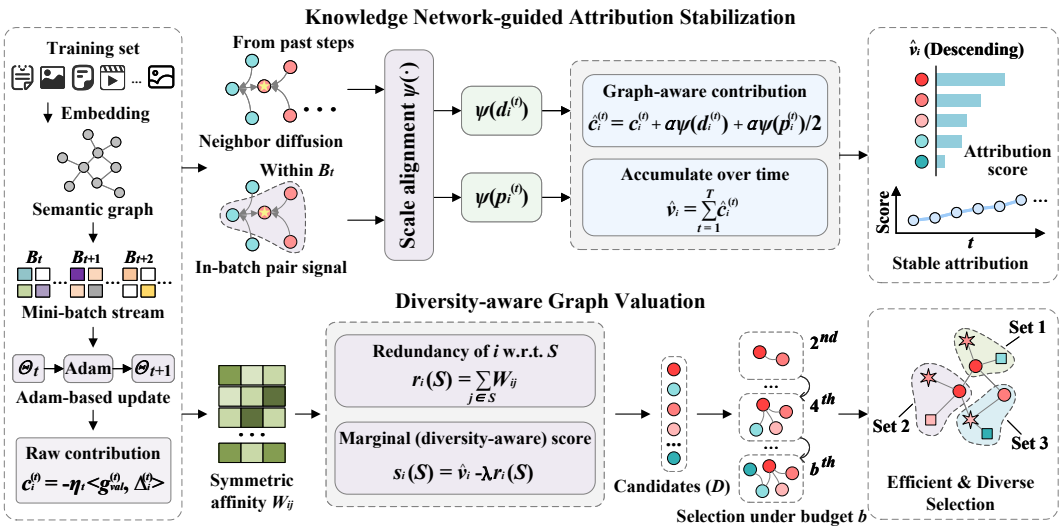

Figure 2: Framework of the proposed Graph-Aware In-Run Data Shapley. The method constructs a sample-level knowledge network from semantic relations and combines historical neighbor diffusion with in-batch pair signals to stabilize sample contributions along one observed Adam training trajectory. The same network provides redundancy-aware marginal scores for subset selection, yielding stable attribution scores and an efficient, diverse subset.

3.1 Knowledge Network-guided Attribution Stabilization

Throughout training, each mini-batch provides a local observation of how its samples alter validation utility. A pointwise approximation can mismatch the optimizer state, while even optimizer-aware accumulation ignores semantic support. To obtain stable attribution, we first derive optimizer-aware contributions and then propagate their evidence over the sample-level knowledge network.

Optimizer-aware In-Run Contribution. For Adam-based training, let $\mathbf{g}_i^{(t)} = \nabla_\theta \ell(\theta_t; z_i)$ denote the gradient of sample z_i. Given the first- and second-moment states $\mathbf{m}_{t-1}$ and $\mathbf{q}_{t-1}$, its sample-conditioned moments are calculated as Ding et al. (2026):

$$\begin{aligned} \mathbf{m}_i^{(t)} &= \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_i^{(t)}, \\ \mathbf{q}_i^{(t)} &= \beta_2 \mathbf{q}_{t-1} + (1-\beta_2)\big(\mathbf{g}_i^{(t)}\big)^2, \end{aligned} \tag{8}$$

where β_1 and β_2 are momentum coefficients. The bias-corrected Adam-aware update direction is:

$$\Delta_i^{(t)} = \frac{\hat{\mathbf{m}}_i^{(t)}}{\sqrt{\hat{\mathbf{q}}_i^{(t)} + \epsilon}}, \tag{9}$$

where ϵ is a numerical stabilizer. Let $\mathbf{g}_{val}^{(t)} = \nabla_\theta U(\theta_t; \mathcal{D}_{val})$ denote the gradient of validation utility. Using the first-order Taylor expansion of Eq. (3), the raw contribution of z_i is formulated as:

$$c_i^{(t)} = -\eta_t \left\langle \mathbf{g}_{val}^{(t)}, \Delta_i^{(t)} \right\rangle, \quad z_i \in \mathcal{B}_t, \tag{10}$$

where $\mathcal{B}_t$ is the mini-batch at step t. This optimizer-aware score measures update-utility alignment but remains sensitive to mini-batch and optimizer-state fluctuations.

Sample-level Knowledge Network Construction. We represent each sample by a semantic embedding $\mathbf{h}_i$, connect its k nearest neighbors using Eq. (5), and normalize nonnegative affinities:

$$\mathbf{W}_{ij} = \frac{\max(\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j), 0)}{\sum_{z_l \in \mathcal{N}(i)} \max(\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_l), 0)}, \quad z_j \in \mathcal{N}(i). \tag{11}$$

where $\mathcal{N}(i)$ is the retained neighborhood of z_i. Each normalized row defines a local evidence distribution, restricting propagation to semantically related samples.

Neighborhood Evidence Diffusion. For batch t, we first aggregate historical neighboring values:

$$d_i^{(t)} = \sum_{z_j \in \mathcal{N}(i)} \mathbf{W}_{ij} v_j^{(t-1)}, \tag{12}$$

where $v_j^{(t-1)} = \sum_{\tau<t} c_j^{(\tau)}$ accumulates raw values before the current update. This term provides structural support from related samples observed earlier, extending beyond the local observation.

Graph Interaction Surrogate. The historical term stabilizes attribution but does not describe interactions among related samples within the same update. A direct second-order expansion would introduce gradient-Hessian-gradient products. We instead define a batch-level pair signal:

$$p_i^{(t)} = \sum_{z_j \in \mathcal{N}(i) \cap \mathcal{B}_t} \mathbf{W}_{ij} c_j^{(t)}. \tag{13}$$

where $\mathcal{N}(i) \cap \mathcal{B}_t$ contains the neighbors that co-occur with z_i at step t. This term encodes a pairwise signal analogous to the interaction component of second-order valuation, serving as a lightweight sample-space surrogate for parameter-space curvature.

Scale-aligned Graph Correction. The graph terms and raw contributions can exhibit different numerical ranges. Direct addition may overwhelm the trajectory-specific signal. Given the vector $\mathbf{c}^{(t)}$ of raw contributions in the current mini-batch, we define the scale-matching operator:

$$\psi(\mathbf{x}) = \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x}) + \epsilon} \sigma(\mathbf{c}^{(t)}), \tag{14}$$

where μ and σ denote the mean and standard deviation, and ϵ prevents division by zero. The operator preserves relative graph evidence while matching its variation to raw scores.

Finally, the graph-aware contribution at step t is calculated as:

$$\hat{c}_i^{(t)} = c_i^{(t)} + \alpha \psi(d_i^{(t)}) + \frac{\alpha}{2} \psi(p_i^{(t)}), \tag{15}$$

α controls the strength of graph correction. The final graph-aware attribution score is obtained by:

$$\hat{v}_i = \sum_{t=0}^{T-1} \hat{c}_i^{(t)}. \tag{16}$$

Through Eq. (15), a locally noisy observation receives support only from related samples, while the pair signal supplies an interaction-aware correction without explicit curvature computation. The sparse graph prevents indiscriminate global smoothing and preserves the contribution carried by each sample's own optimization trajectory.

3.2 Diversity-aware Graph Valuation

Stable attribution identifies valuable examples, but value ranking alone does not determine an effective subset. Multiple near-duplicate samples may receive similar scores because they induce similar updates. Selecting them repeatedly spends the data budget on the same semantic region. We therefore reuse the knowledge network to balance trajectory-derived value against the information already covered by the selected subset.

Symmetric Neighborhood Affinity. Since k-nearest-neighbor membership need not be reciprocal, we use $\bar{\mathbf{W}}_{ij} = (\mathbf{W}_{ij} + \mathbf{W}_{ji})/2$, assigning zero to absent edges. This preserves sparsity and gives each related pair a symmetric redundancy cost.

Value-Diversity Objective. Given a budget b, an effective subset should balance graph-aware value against repeated neighborhood coverage. We formulate this objective as:

$$\max_{\mathcal{S} \subseteq \mathcal{D}, |\mathcal{S}|=b} \sum_{z_i \in \mathcal{S}} \hat{v}_i - \lambda \sum_{\substack{z_i, z_j \in \mathcal{S} \\ i<j}} \bar{\mathbf{W}}_{ij}, \tag{17}$$

where λ controls the trade-off between sample value and semantic redundancy. The first term rewards examples that contribute to the observed training trajectory, whereas the second term discourages dense selections within a single semantic region.

Neighborhood-aware Marginal Value. For a candidate z_i and the current subset $\mathcal{S}$, we define its covered-neighborhood mass as:

$$r_i(\mathcal{S}) = \sum_{z_j \in \mathcal{S}} \bar{\mathbf{W}}_{ij}. \tag{18}$$

A large $r_i(\mathcal{S})$ indicates information already represented by neighbors, yielding the marginal score:

$$s_i(\mathcal{S}) = \hat{v}_i - \lambda r_i(\mathcal{S}), \tag{19}$$

which equals the gain of adding z_i to Eq. (17), directly integrating deduplication into selection.

Iterative Graph Selection. Starting from $\mathcal{S} = \varnothing$, the selected subset is generated by:

$$z^* = \underset{z_i \in \mathcal{D} \setminus \mathcal{S}}{\operatorname{argmax}} s_i(\mathcal{S}), \quad \mathcal{S} \leftarrow \mathcal{S} \cup \{z^*\}. \tag{20}$$

After selecting z^*, the redundancy of each remaining candidate can be updated incrementally:

$$r_i(\mathcal{S} \cup \{z^*\}) = r_i(\mathcal{S}) + \bar{\mathbf{W}}_{i*}. \tag{21}$$

Only candidates connected to z^* receive the update. The knowledge network therefore guides the subset toward distinct semantic neighborhoods without evaluating all pairwise interactions.

Coupling Attribution and Selection. The shared network corroborates uncertain attribution signals and identifies information already covered during selection, avoiding an unrelated post-hoc diversity heuristic. Selection remains traceable: each penalized candidate is linked to the selected neighbors responsible for its redundancy cost.

3.3 Training and Optimization

Graph-Aware In-Run Data Shapley runs alongside the original optimizer without additional model training. At step t, the model minimizes $\mathcal{L}_{tr}^{(t)} = |\mathcal{B}_t|^{-1} \sum_{z_i \in \mathcal{B}_t} \ell(\theta_t; z_i)$ using the standard update; the knowledge network alters neither the objective nor the training trajectory. Before this update, we compute the validation gradient and sample-conditioned directions in Eq. (9), evaluate the raw and graph-aware contributions through Eqs. (10) and (15), and accumulate:

$$\hat{v}_i^{(t+1)} = \hat{v}_i^{(t)} + \mathbb{I}[z_i \in \mathcal{B}_t]\hat{c}_i^{(t)}, \quad \hat{v}_i^{(0)} = 0, \tag{22}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Repeating the procedure for T steps yields $\hat{\mathbf{v}}$ from the same training run. Afterward, Eq. (20) selects b samples, with λ controlling semantic diversity.

Graph-guided Data Selection. Attribution visits only k neighbors per mini-batch sample, incurring $\mathcal{O}(|\mathcal{B}_t|k)$ graph cost per step, while Eq. (21) updates only edges incident to each newly selected sample. The procedure thus avoids Hessian matrices, Hessian-vector products, and retraining while producing stable scores and a diverse high-value subset.

4 Experiments

Experimental Protocol. We evaluate data valuation in both language and vision settings. For language, we sample 100,000 training examples and 10,000 disjoint validation examples from the Pile dataset Gao et al. (2020), with each example truncated to 128 tokens. We use GPT-2 for attribution and fine-tune all model parameters when evaluating the selected subsets. For vision, we use CIFAR-10 with the ResNet-18 model, sampling 1,000 training images per run and using held-out images for valuation and final evaluation. We compare with Random, BM25 Robertson & Zaragoza (2009), and KNN-Shapley Jia et al. (2019); retraining-based Data Shapley, Datamodels, and Empirical Influence Ghorbani & Zou (2019); Ilyas et al. (2022); Wang et al. (2025); trained-model-based Influence Functions, TRAK, and LESS Koh & Liang (2017); Park et al. (2023); Xia et al. (2024); and first- and second-order In-Run Data Shapley and their Adam-aware variants Wang et al. (2025); Ding et al. (2026). Unless otherwise specified, we construct a sparse sample-level knowledge network in task-aligned representation spaces by retaining the ten strongest semantic neighbors of each sample and normalizing their cosine affinities. We derive these representations from pooled GPT-2 hidden states for Pile and penultimate-layer ResNet-18 features for CIFAR-10, and use the resulting network as a shared structural prior for attribution and diversity-aware selection.

4.1 How Effective Is Graph-Aware In-Run Data Valuation?

We evaluate the methods from two complementary perspectives. Noise AUROC measures how well attribution scores distinguish corrupted training examples from clean ones, where 0.5 indicates random performance. Val. Accuracy evaluates next-token prediction after retraining GPT-2 on the selected 20%-80% subsets across four data budgets. Avg. Rank summarizes overall performance over the five reported metrics, providing a compact measure of cross-objective consistency.

Table 1: Overall data valuation performance on Pile. Values are the mean and standard deviation over three runs. Avg. Rank is computed over the five reported metric means. Higher is better except for Avg. Rank. The best and second-best results are marked in **bold** and underlined, respectively.

	Attribution Quality	Val. Accuracy (%)				Overall
Method	Noise AUROC ↑	20% ↑	40% ↑	60% ↑	80% ↑	Avg. Rank ↓
Random	0.499 (0.001)	38.852 (0.026)	39.182 (0.014)	41.295 (0.018)	41.345 (0.024)	9.40
BM25	0.460 (0.001)	38.085 (0.011)	39.471 (0.029)	40.740 (0.021)	42.930 (0.052)	9.80
KNN-Shapley	**0.623** (0.000)	38.643 (0.063)	39.005 (0.013)	40.285 (0.011)	41.629 (0.043)	8.60
Influence Functions	0.506 (0.002)	39.581 (0.061)	39.989 (0.018)	40.166 (0.041)	41.265 (0.033)	9.00
TRAK (1 ckpt)	0.516 (0.005)	39.526 (0.039)	39.978 (0.026)	41.163 (0.023)	42.287 (0.010)	8.00
LESS	0.516 (0.005)	39.576 (0.040)	39.978 (0.026)	40.163 (0.023)	41.287 (0.010)	9.20
In-Run Data Shapley (1st)	0.497 (0.004)	40.763 (0.022)	40.859 (0.034)	42.282 (0.020)	43.346 (0.009)	6.80
In-Run Data Shapley (2nd)	0.498 (0.004)	40.793 (0.016)	41.143 (0.011)	42.304 (0.009)	42.365 (0.019)	6.20
Adam-aware In-Run (1st)	0.560 (0.002)	41.055 (0.005)	41.180 (0.011)	42.304 (0.007)	43.366 (0.007)	3.50
Adam-aware In-Run (2nd)	0.559 (0.002)	40.803 (0.021)	41.143 (0.016)	43.261 (0.010)	43.351 (0.019)	4.10
Graph-aware In-Run (1st)	0.563 (0.002)	41.799 (0.020)	**42.132** (0.009)	**43.308** (0.021)	44.345 (0.015)	**1.60**
Graph-aware In-Run (2nd)	0.562 (0.001)	**41.825** (0.011)	42.126 (0.025)	43.275 (0.004)	**44.350** (0.021)	1.80

As shown in Table 1, the Graph-aware In-Run approach achieves the highest validation accuracy at every selection budget (Val. Accuracy). Compared with the corresponding Adam-aware estimator, the first-order variant improves accuracy by 0.744-1.004 percentage points across the four budgets, while the second-order variant obtains gains of up to 1.022 points. The Graph-aware first- and second-order methods also attain the two best average ranks, demonstrating strong performance across the remaining evaluation metrics. These results highlight the value of incorporating sample relations into in-run attribution, yielding subsets that better support downstream model training across data budgets. Graph correction further improves both Adam-aware estimators on noise detection, while preserving the gains observed in subset training. Although KNN-Shapley remains strongest on Noise AUROC, reflecting its specialization in neighborhood-based detection, the proposed method performs consistently well across both attribution quality and downstream utility. Overall, the results demonstrate that Graph-aware In-Run provides the strongest cross-metric performance, with its clearest advantage appearing in high-value subset selection.

4.2 Robustness and Attribution Stability

We evaluate robustness with 10% corrupted texts on Pile and 10% mislabeled images on CIFAR-10 over three random seeds. AUROC measures how well attribution scores distinguish corrupted from clean samples, while contamination measures the corrupted fraction among the top-10% valued samples. Score change and promotion rate further indicate whether corrupted samples lose or gain value after graph correction. As shown in Table 2, graph correction increases CIFAR-10 AUROC by 0.127-0.132 and reduces contamination by 6 percentage points, with only 1.0-1.3% of corrupted images gaining value. We also assess whether different seeds produce similar sample rankings (Spearman and Kendall), select the same top-10% samples (overlap), and assign consistent attribution scores (dispersion). All four stability measures improve after graph correction. These diagnostics use cached scores and the frozen Pile trajectory-replay approximation.

4.3 High-Value and Diverse Data Selection

We compare the complete Graph method with same-order In-Run Data Shapley using Top-k selection under four low-data budgets. Validation loss and token accuracy measure the utility of the selected subset after retraining, while graph affinity measures redundancy among selected samples.

Table 2: Robustness and attribution stability. Contam.: corrupted fraction among the top-10% valued samples; Δ_{noise}/Promo.: mean standardized score change/promotion rate of corrupted nodes.

Corruption Robustness					
Dataset	Method	AUROC ↑	Contam. ↓	Δ_{noise} ↓	Promo. ↓
Pile	Adam-aware In-Run (1st)	0.559 (0.002)	0.093 (0.002)	–	–
	Adam-aware In-Run (2nd)	0.559 (0.002)	**0.093** (0.002)	–	–
	Graph-aware In-Run (1st)	**0.563** (0.002)	**0.092** (0.002)	−0.010	0.406
	Graph-aware In-Run (2nd)	**0.562** (0.001)	**0.093** (0.002)	−0.010	0.408
CIFAR-10	Adam-aware In-Run (1st)	0.500 (0.019)	0.130 (0.024)	–	–
	Adam-aware In-Run (2nd)	0.491 (0.009)	0.127 (0.009)	–	–
	Graph-aware In-Run (1st)	**0.627** (0.015)	**0.070** (0.008)	−0.416	0.013
	Graph-aware In-Run (2nd)	**0.623** (0.007)	**0.067** (0.012)	−0.401	0.010

Cross-Seed Stability on Pile				
Method	Spearman ↑	Kendall τ ↑	Overlap ↑	Dispersion ↓
Adam-aware In-Run (1st)	0.164	0.110	0.164	0.623
Adam-aware In-Run (2nd)	0.165	0.111	0.166	0.622
Graph-aware In-Run (1st)	**0.183**	**0.123**	**0.175**	**0.615**
Graph-aware In-Run (2nd)	**0.185**	**0.124**	**0.177**	**0.614**

One-hop coverage is the fraction of samples covered by the selected nodes and their neighbors, and domain entropy measures the balance across data domains. As shown in Table 3, Graph improves validation loss and token accuracy in all eight order-budget settings, with the largest gains at the 1% and 2% budgets. It also consistently reduces graph affinity and increases one-hop coverage, indicating less redundant and broader semantic selection. Domain entropy improves in six settings and changes only marginally at 10%, where domain coverage approaches saturation. These results show that graph-guided selection improves both subset utility and semantic diversity.

Table 3: Subset utility and diversity on Pile. In-Run denotes the In-Run Data Shapley with Top-k selection, while Graph denotes graph-aware attribution.

		1%		2%		5%		10%	
Metric	Order	In-Run	Graph	In-Run	Graph	In-Run	Graph	In-Run	Graph
Val. loss ↓	1st	3.4080	**3.2837**	3.3639	**3.2475**	3.2636	**3.1917**	3.2251	**3.1604**
	2nd	3.4106	**3.2869**	3.3592	**3.2457**	3.2503	**3.1953**	3.2146	**3.1632**
Token accuracy (%) ↑	1st	40.7012	**41.9672**	41.2997	**42.4882**	42.0751	**43.0699**	42.8677	**43.4701**
	2nd	40.6274	**41.9214**	41.2974	**42.4846**	42.0856	**43.0658**	42.9921	**43.4266**
Graph affinity/item ↓	1st	0.0156	**0.0095**	0.0220	**0.0159**	0.0475	**0.0339**	0.0720	**0.0609**
	2nd	0.0154	**0.0097**	0.0218	**0.0161**	0.0472	**0.0345**	0.0719	**0.0611**
One-hop coverage ↑	1st	0.1246	**0.1375**	0.2336	**0.2536**	0.4840	**0.5054**	0.6807	**0.7471**
	2nd	0.1248	**0.1371**	0.2340	**0.2529**	0.4844	**0.5042**	0.6910	**0.7456**
Domain entropy ↑	1st	2.0186	**2.0745**	2.0658	**2.1018**	2.0983	**2.1027**	**2.1010**	2.0998
	2nd	2.0205	**2.0742**	2.0660	**2.0993**	2.0999	**2.1044**	**2.1009**	2.0999

4.4 Semantic Retrieval and Domain Contribution Dynamics

Table 4 evaluates source retrieval under textual variation. Graph In-Run ties for first on partial-exact and paraphrased queries and leads under significant paraphrasing with a rank of 1.3. For similar-topic queries, it approaches Influence Function (26.7 vs. 26.0) and outperforms BM25 (231.3). These results concern the complete fusion of semantic and lexical retrieval with graph-regularized scores, rather than graph propagation alone. We further track domain contributions with 1,817 training documents from 17 domains and 16 mathematical validation documents over 500 GPT-2 updates. In Fig. 3, PubMed Abstracts falls from 48.4% at step 16 to 4.0% at step 500, while DM Mathematics finishes highest at 25.4%. The highest- and lowest-scoring documents are a math-

Table 4: Source ranks under textual variation. Graph rows evaluate the retrieval fusion system. Best and second-best distinct ranks are bold and underlined; shaded rows denote our methods.

Method	Partial-exact	Paraphrase	Strong paraphrase	Similar topic
In-Run Data Shapley (1st)	<u>141.7</u>	144.0	136.0	146.7
In-Run Data Shapley (2nd)	<u>141.7</u>	144.0	136.0	146.7
Influence Function	**1.0**	**1.0**	5.7	**26.0**
BM25	**1.0**	<u>2.7</u>	<u>2.3</u>	231.3
Graph-aware In-Run (1st)	**1.0**	**1.0**	**1.3**	<u>26.7</u>
Graph-aware In-Run (2nd)	**1.0**	**1.0**	**1.3**	<u>26.7</u>

ematical sequence example (7.02×10^{-5}) and a GitHub configuration example (-8.62×10^{-5}), respectively, illustrating objective-specific and stage-dependent data value.

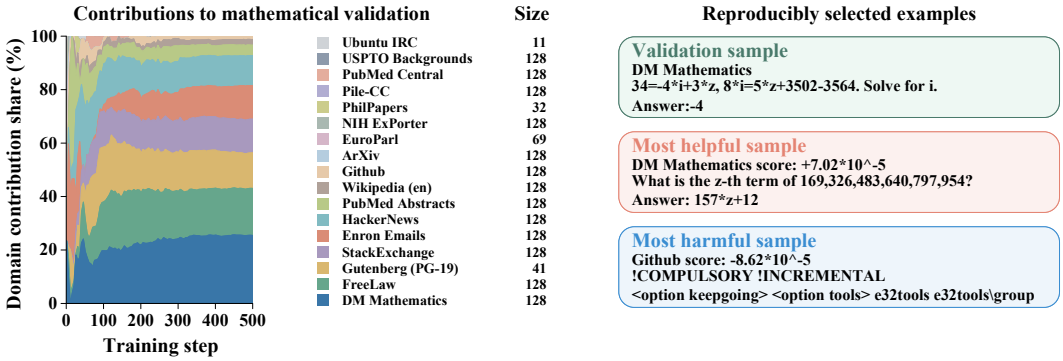

Figure 3: Domain contribution dynamics toward mathematical validation utility. Areas show normalized domain contribution shares; Size indicates the number of training documents per domain.

4.5 Online Computational Efficiency

We compare HVP-based (Hessian–Vector Product) attribution with the proposed HVP-free Graph implementation. Experiments use GPT-2 on 1,000 Pile documents of 128 tokens. Table 5 reports clean-setting results. Relative to second-order Adam-aware attribution, Graph reduces interaction time from 2.690 to 0.037 seconds, total runtime by 5.6%, and peak online memory by 47.0%. The smaller end-to-end gain reflects the shared cost of gradient computation and training.

Table 5: Online efficiency on Pile. Times are seconds, reported as mean (sample standard deviation) over three seeds. Attribution and total time include graph construction where applicable; attribution excludes optimizer updates. Memory denotes peak online GPU usage.

Method	Interaction	Attribution	Total	Memory (MiB)
In-Run (2nd)	<u>2.671</u> (0.033)	<u>12.310</u> (0.128)	<u>14.767</u> (0.167)	<u>3316.1</u>
Adam-aware (2nd)	2.690 (0.014)	12.339 (0.098)	14.770 (0.125)	3320.1
HVP-free Graph	**0.037** (0.001)	**11.508** (0.085)	**13.939** (0.098)	**1756.6**

5 Conclusion

In this paper, we propose Graph-Aware In-Run Data Shapley, which uses a sample-level knowledge network to stabilize attribution and guide diverse subset selection. By combining optimizer-aware contributions with sparse graph interactions, the framework incorporates semantic relations into training-data valuation. Experiments on language and vision datasets support improvements in attribution stability, corruption detection, and subset quality, while an HVP-free implementation reduces computational and memory costs. These findings highlight the value of modeling sample relations for effective data valuation along a single training trajectory.

AI USE STATEMENT

Generative AI tools were used only for language polishing. All research ideas, methodology, experiments, analyses, and conclusions were developed and verified by the authors.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or other sensitive information, and we identify no specific ethical concerns arising from this study.

REFERENCES

Amro Kamal Mohamed Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. In *ICLR 2023 Workshops: MRL*, 2023.

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*, 2025.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff G. Schneider, Eduard H. Hovy, Roger Baker Grosse, and Eric P. Xing. What is your data worth to GPT? LLM-scale data valuation with influence functions. In *Advances in Neural Information Processing Systems*, 2025.

Meng Ding, Zeqing Zhang, Di Wang, and Lijie Hu. In-run data shapley for adam optimizer. *Third Workshop on Data-Centric Foundation Models at ICLR*, 2026.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

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, 2019.

Amirata Ghorbani, Michael P. Kim, and James Zou. A distributional framework for data valuation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3535–3544, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, pp. 30016–30030, 2022.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J. Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11):1610–1623, 2019.

Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. LAVA: Data valuation without pre-specified learning algorithms. In *International Conference on Learning Representations*, 2023.

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

Yongchan Kwon and James Zou. Beta shapley: A unified and noise-reduced data valuation framework for machine learning. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802, 2022.

Yongchan Kwon and James Zou. Data-OOB: Out-of-bag estimate as a simple and efficient data value. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18135–18152, 2023.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. DataInf: Efficiently estimating data influence in LoRA-tuned LLMs and diffusion models. In *International Conference on Learning Representations*, 2024.

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, 2022.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In *Advances in Neural Information Processing Systems*, 2024.

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

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, 2021.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 4(1-2):1–174, 2009.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Jiachen T. Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6388–6421, 2023.

Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *International Conference on Learning Representations*, 2025.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, 2023.

Chih-Kuan Yeh, Joon Sik Kim, Ian En-Hsu Yen, and Pradeep K. Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Jiantao Qiu, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. Harnessing diversity for important data selection in pretraining large language models. In *International Conference on Learning Representations*, 2025.

A Related works

A.1 Model-Behavior-Based Valuation

Model-behavior-based valuation estimates the usefulness of training examples from observable model signals, such as predictions, gradients, and training dynamics. Early representative approaches include influence functions, which approximate the effect of up-weighting a training example through gradients and Hessian-vector products Koh & Liang (2017), and representer-point methods, which decompose predictions into weighted similarities with training examples Yeh et al. (2018). TracIn avoids explicit Hessian inversion by accumulating gradient alignment along saved checkpoints Pruthi et al. (2020), while datamodels learn counterfactual prediction models from many trained subsets Ilyas et al. (2022). TRAK further improves the efficiency-accuracy tradeoff by combining random projections with kernel-inspired attribution Park et al. (2023). With the rise of large language models, recent work scales this behavior-based view with low-rank gradient projection, optimizer-aware influence, LoRA-specific approximations, and fact-tracing systems such as LOGRA/LOGIX, LESS, DataInf, and TrackStar Choe et al. (2025); Xia et al. (2024); Kwon et al. (2024); Chang et al. (2025). These methods make training data valuation increasingly practical, but they usually operate after or around a trained model and often require gradient stores, checkpoints, Hessian approximations, or additional projection systems.

Another branch uses model behavior as a selection or curation signal rather than as an attribution score for a specific prediction. Forgetting events and EL2N identify examples that are repeatedly forgotten or remain informative early in training Toneva et al. (2019); Paul et al. (2021). Data pruning and scaling-law studies suggest that carefully selected examples can reduce the cost of scaling models Sorscher et al. (2022); Hoffmann et al. (2022). For language-model pretraining, DoReMi optimizes domain mixtures, RHO-1 emphasizes useful tokens, and recent diversity-aware methods such as Quad show that importance and diversity should be considered jointly Xie et al. (2023); Lin et al. (2024); Zhang et al. (2025). Semantic deduplication methods further show that embedding-space relations can remove redundant corpus regions before training Abbas et al. (2023). These studies reveal that data value is not only pointwise utility of an isolated example, but also depends on training stage, model state, data distribution, and semantic relations among examples, which motivates valuation methods to move beyond independent scoring.

A.2 Shapley-Based Data Attribution

Shapley-based data attribution provides a principled way to assign each training example a marginal contribution to model utility. Data Shapley formalizes this idea by viewing training examples as players in a cooperative game Ghorbani & Zou (2019). Follow-up studies improve its statistical efficiency or adapt it to specific settings, including distributional Shapley, KNN-Shapley, Beta Shapley, Data-OOB, Data Banzhaf, randomized-experiment estimators, and learning-agnostic valuation frameworks Ghorbani et al. (2020); Jia et al. (2019); Kwon & Zou (2022; 2023); Wang & Jia (2023); Lin et al. (2022); Just et al. (2023). These methods are effective evaluation standards for identifying helpful or harmful samples, but many of them still depend on repeated subset training, predefined utility functions, or proxy distances between training and validation distributions.

In-Run Data Shapley recently shifts Shapley-style valuation from repeated retraining to a single observed optimization trajectory Wang et al. (2025). Instead of asking how performance changes

over many independently trained subsets, it estimates per-sample marginal contributions during the actual training process, making Shapley-style attribution more compatible with foundation-model pretraining. The Adam-optimizer extension further adapts the in-run formulation to modern optimizer states Ding et al. (2026). However, existing in-run estimators mainly accumulate contributions for each sample independently, so noisy gradient signals may lead to unstable scores and high-value near-duplicates may dominate subset selection.

In summary, behavior-based methods improve scalability by extracting value signals from trained models or training dynamics, while Shapley-based methods offer a clearer valuation principle but often pay for it with retraining or pointwise estimation. Existing methods also tend to separate attribution stability from diversity-aware selection: the former is handled through optimizer or gradient approximations, and the latter through separate deduplication or data-selection heuristics. Our work bridges these two threads by introducing a sample-level knowledge network into In-Run Data Shapley. The network diffuses structurally supported attribution evidence and penalizes redundant neighbors, providing stable and diverse data valuation in one training run without expensive Hessian-dependent interaction modeling.

B Algorithmic Procedure

Algorithm 1 combines online attribution with the greedy selection in Eqs. (20) and (21). Separate buffers $\mathbf{a}$ and $\hat{\mathbf{v}}$ store raw and corrected scores. We use larger-is-better utility, with $U = -\mathcal{L}_{val}$ for loss-based evaluation.

Algorithm 1 Graph-Aware In-Run Data Shapley and subset selection

Require: Training data $\mathcal{D}$, validation data $\mathcal{D}_{val}$, model θ_0, optimizer, T steps, neighborhood size k, correction strength α, penalty $\lambda \geq 0$, budget $b \leq n$
Ensure: Corrected scores $\hat{\mathbf{v}}$ and selected subset $\mathcal{S}$

Construct semantic k-neighbor graph $\mathbf{W}$
Initialize raw and corrected scores $\mathbf{a} = \hat{\mathbf{v}} = \mathbf{0}$
for $t = 0, \ldots, T-1$ **do**
  Obtain $\mathcal{B}_t$ and snapshot the pre-update Adam state
  Compute the validation-utility gradient $\mathbf{g}_{val}^{(t)}$
  Compute all $\Delta_i^{(t)}$ from the saved Adam state, without higher-order differentiation
  $c_i^{(t)} \leftarrow -\eta_t \langle \mathbf{g}_{val}^{(t)}, \Delta_i^{(t)} \rangle$, for all $z_i \in \mathcal{B}_t$
  $d_i^{(t)} \leftarrow \sum_{z_j \in \mathcal{N}(i)} \mathbf{W}_{ij} a_j$, for all $z_i \in \mathcal{B}_t$
  $p_i^{(t)} \leftarrow \sum_{z_j \in \mathcal{N}(i) \cap \mathcal{B}_t} \mathbf{W}_{ij} c_j^{(t)}$, using a batch-index map
  $\hat{\mathbf{c}}^{(t)} \leftarrow \mathbf{c}^{(t)} + \alpha \psi(\mathbf{d}^{(t)}) + (\alpha/2)\psi(\mathbf{p}^{(t)})$
  Accumulate $a_i \leftarrow a_i + c_i^{(t)}$ and $\hat{v}_i \leftarrow \hat{v}_i + \hat{c}_i^{(t)}$ for all $z_i \in \mathcal{B}_t$
  Apply the original optimizer update using the training loss
end for
Set $\bar{\mathbf{W}} \leftarrow (\mathbf{W} + \mathbf{W}^\top)/2$, $\mathcal{S} \leftarrow \varnothing$, and $\mathbf{r} \leftarrow \mathbf{0}$
for $q = 1, \ldots, b$ **do**
  $i^* \leftarrow \arg\max_{i : z_i \notin \mathcal{S}} (\hat{v}_i - \lambda r_i)$, breaking ties by sample index
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{z_{i^*}\}$
  Update $r_i \leftarrow r_i + \bar{\mathbf{W}}_{ii^*}$ for all unselected neighbors of z_{i^*}
end for
return $\hat{\mathbf{v}}, \mathcal{S}$

Implementation conventions. The graph is fixed before attribution, with zero diagonal; rows without positive affinities remain zero. Historical diffusion reads raw scores accumulated before the current step, while the pair term uses current-batch contributions. The operator ψ acts batchwise and returns zero for degenerate scales. Attribution computations leave optimizer moments unchanged and preserve the random state used by training, including dropout. Loss-change scores are negated before graph correction to follow the larger-is-better utility convention.

C Full Derivations and Complexity

C.1 Optimizer-aware Local Contributions

Conditional Adam directions. At step t, fix the observed parameters and optimizer state. Let s_t be the number of completed optimizer updates, and define $F_t(\mathbf{g})$ by:

$$F_t(\mathbf{g}) = \frac{\big[\beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}\big]/(1-\beta_1^{s_t+1})}{\sqrt{\big[\beta_2 \mathbf{q}_{t-1} + (1-\beta_2)\mathbf{g}^{\odot 2}\big]/(1-\beta_2^{s_t+1})} + \epsilon}, \tag{23}$$

$$\Delta_i^{(t)} = F_t(\mathbf{g}_i^{(t)}). \tag{24}$$

Operations are elementwise and use fixed pre-update moments. The proxy excludes decoupled weight decay; including AdamW decay requires shifting the utility reference by its coalition-independent displacement.

Taylor expansion. Let $\mathbf{H}_U^{(t)} = \nabla_\theta^2 U(\theta_t; \mathcal{D}_{val})$. If U is twice differentiable with locally Lipschitz Hessian, a displacement $\mathbf{s}$ satisfies:

$$U(\theta_t + \mathbf{s}) - U(\theta_t) = \langle \mathbf{g}_{val}^{(t)}, \mathbf{s} \rangle + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_U^{(t)} \mathbf{s} + \mathcal{O}(\|\mathbf{s}\|^3). \tag{25}$$

Setting $\mathbf{s} = -\eta_t \Delta_i^{(t)}$ gives Eq. (10). With $U = -\mathcal{L}_{val}$, positive scores predict loss reduction; loss-change scores have the opposite sign.

An additive direction surrogate. For fixed $\mathbf{s}_i = -\eta_t \Delta_i^{(t)}$ and additive coalition displacements $\mathbf{s}_\mathcal{S} = \sum_{z_i \in \mathcal{S}} \mathbf{s}_i$, define:

$$\widetilde{U}_t(\mathcal{S}) = \sum_{z_i \in \mathcal{S}} \mathbf{g}_{val}^{(t)\top} \mathbf{s}_i + \frac{1}{2} \sum_{z_i, z_j \in \mathcal{S}} \mathbf{s}_i^\top \mathbf{H}_U^{(t)} \mathbf{s}_j. \tag{26}$$

For a predecessor set $\mathcal{S}$ excluding z_i, its marginal gain is:

$$\widetilde{U}_t(\mathcal{S} \cup \{z_i\}) - \widetilde{U}_t(\mathcal{S}) = \mathbf{g}_{val}^{(t)\top} \mathbf{s}_i + \frac{1}{2}\mathbf{s}_i^\top \mathbf{H}_U^{(t)} \mathbf{s}_i + \sum_{z_j \in \mathcal{S}} \mathbf{s}_i^\top \mathbf{H}_U^{(t)} \mathbf{s}_j. \tag{27}$$

In a uniformly random permutation, each $z_j \neq z_i$ precedes z_i with probability $1/2$. Averaging Eq. (27) therefore gives:

$$\begin{aligned} \widetilde{c}_{i,2}^{(t)} &= c_i^{(t)} + \frac{1}{2}\mathbf{s}_i^\top \mathbf{H}_U^{(t)} \sum_{z_j \in \mathcal{B}_t} \mathbf{s}_j \\ &= c_i^{(t)} + \frac{\eta_t^2}{2} \Delta_i^{(t)\top} \mathbf{H}_U^{(t)} \sum_{z_j \in \mathcal{B}_t} \Delta_j^{(t)}. \end{aligned} \tag{28}$$

This exact allocation of the quadratic surrogate sums to $\widetilde{U}_t(\mathcal{B}_t)$ and requires only one HVP (Hessian–Vector Product) with the total batch displacement.

Approximation boundary. Adam is nonlinear: $F_t(\sum_i \mathbf{g}_i) \neq \sum_i F_t(\mathbf{g}_i)$ in general, and coalition means depend on subset size. Eq. (28) therefore approximates the actual Adam game through both Taylor truncation and additive directions. Any fixed batch-normalization factor must enter $\mathbf{s}_i$ consistently in both orders.

For curvature $h_i = \widetilde{c}_{i,2}^{(t)} - c_i^{(t)}$, the second-order implementation applies a scale cap $\rho \geq 0$:

$$\gamma_t = \min\left\{1, \frac{\rho\, \sigma(\mathbf{c}^{(t)})}{\sigma(\mathbf{h}^{(t)}) + \epsilon}\right\}, \qquad c_{i,\text{clip}}^{(t)} = c_i^{(t)} + \gamma_t h_i. \tag{29}$$

Degenerate curvature scales or a disabled cap leave h_i unchanged. Capping modifies the quadratic allocation; the HVP-free graph branch uses neither curvature nor second-order caches.

C.2 Graph Evidence and Scale Alignment

Historical and current evidence. Let $\mathbf{R}_t \in \{0,1\}^{B_t \times n}$ select the current batch, where $B_t = |\mathcal{B}_t|$, and let $\mathbf{a}_t = \sum_{\tau<t} \mathbf{R}_\tau^\top \mathbf{c}^{(\tau)}$ store accumulated raw scores. The two graph terms are:

$$\mathbf{d}^{(t)} = \mathbf{R}_t \mathbf{W} \mathbf{a}_t, \qquad \mathbf{p}^{(t)} = \mathbf{R}_t \mathbf{W} \mathbf{R}_t^\top \mathbf{c}^{(t)}. \tag{30}$$

These are Eqs. (12) and (13) in matrix form: historical evidence and current batch evidence, respectively. Semantic affinity does not equal $\Delta_i^\top \mathbf{H}_U \Delta_j$, whose sign and magnitude depend on curvature; the pair term is a structural surrogate.

Scale and total-score preservation. Using the population standard deviation over the current batch, Eq. (14) has:

$$\mathbf{1}^\top \psi(\mathbf{x}) = 0, \qquad \sigma(\psi(\mathbf{x})) = \sigma(\mathbf{c}^{(t)}) \frac{\sigma(\mathbf{x})}{\sigma(\mathbf{x}) + \epsilon} \le \sigma(\mathbf{c}^{(t)}). \tag{31}$$

Thus centered graph evidence has bounded variation, with zero correction for degenerate scales. Consequently:

$$\sum_{z_i \in \mathcal{B}_t} \hat{c}_i^{(t)} = \sum_{z_i \in \mathcal{B}_t} c_i^{(t)}, \qquad \sum_{i=1}^{n} \hat{v}_i = \sum_{i=1}^{n} v_i. \tag{32}$$

Without subsequent clipping or reweighting, the correction preserves total raw proxy value up to numerical error. This does not establish exact utility conservation, all Shapley axioms, or guaranteed variance reduction; reliability still depends on neighboring evidence.

C.3 Diversity Objective and Greedy Updates

Let J denote Eq. (17), with symmetric nonnegative $\bar{\mathbf{W}}$ and zero diagonal. Adding $z_i \notin \mathcal{S}$ gives:

$$J(\mathcal{S} \cup \{z_i\}) - J(\mathcal{S}) = \hat{v}_i - \lambda \sum_{z_j \in \mathcal{S}} \bar{\mathbf{W}}_{ij} = \hat{v}_i - \lambda r_i(\mathcal{S}). \tag{33}$$

After adding z_{i^*}, only its incident edges introduce new redundancy:

$$r_i(\mathcal{S} \cup \{z_{i^*}\}) - r_i(\mathcal{S}) = \bar{\mathbf{W}}_{ii^*}. \tag{34}$$

The marginal update is exact. For $\mathcal{S} \subseteq \mathcal{T}$ and $\lambda \ge 0$:

$$\hat{v}_i - \lambda r_i(\mathcal{S}) \ge \hat{v}_i - \lambda r_i(\mathcal{T}), \qquad z_i \notin \mathcal{T}. \tag{35}$$

Thus J has diminishing returns but may have negative marginals. Fixed-budget greedy selection has no claimed global-optimality or monotone-submodular guarantee; $\lambda = 0$ recovers Top-b selection.

C.4 Time and Space Complexity

Let p be the number of attributed parameters, d the embedding dimension, $B = \max_t B_t$, and m the number of stored directed edges. A k-neighbor graph satisfies $m \le nk$. Let C_{emb} denote representation extraction cost and C_t^{base} the cost of validation gradients, per-sample gradients, Adam-aware directions, and the actual optimizer update at step t.

Graph construction. Exact similarities require $\mathcal{O}(n^2 d)$ time and $\mathcal{O}(n^2)$ storage. Chunking q rows reduces temporary storage to $\mathcal{O}(qn)$, plus $\mathcal{O}(nd + nk)$ for embeddings and edges, without reducing similarity computation. Top-k partitioning adds $\mathcal{O}(n^2)$ work, versus $\mathcal{O}(n^2 \log n)$ for full sorting. ANN costs depend on the index, dimension, and recall; no universal subquadratic bound is assumed.

Online attribution. With a $\mathcal{O}(B_t)$ batch-index map and expected constant-time lookup, graph aggregation costs $\mathcal{O}(B_t k)$; scale matching and accumulation cost $\mathcal{O}(B_t)$. Hence:

$$C_{\text{online}} = C_{\text{emb}} + C_{\text{build}} + \sum_{t=0}^{T-1} \left[C_t^{\text{base}} + \mathcal{O}(B_t k) \right]. \tag{36}$$

Table 6: Additional costs beyond common model differentiation and optimization. $m_s \leq 2nk$ is the symmetrized edge count; HVP memory also depends on retained activations.

Component	Time	Storage
Dense similarity computation	$\mathcal{O}(n^2 d)$	$\mathcal{O}(n^2 + nd)$
Chunked exact similarities	$\mathcal{O}(n^2 d)$	$\mathcal{O}(qn + nd + nk)$
Sparse graph correction / step	$\mathcal{O}(B_t k)$	$\mathcal{O}(nk + n + Bk)$
HVP curvature / step	$C_{\mathrm{HVP},t} + \mathcal{O}(B_t p)$	step vectors $\mathcal{O}(B_t p)$
Greedy selection, full scans	$\mathcal{O}(bn + m_s)$	$\mathcal{O}(n + m_s)$
Greedy selection, indexed heap	$\mathcal{O}(n + (b + m_s)\log n)$	$\mathcal{O}(n + m_s)$

Here C_{build} excludes embedding extraction. Graph arithmetic alone costs $\mathcal{O}(B_t k)$, versus $\mathcal{O}(B_t n)$ for dense historical aggregation. Storage is $\mathcal{O}(nk+n+Bk)$, plus $\mathcal{O}(nd)$ if embeddings are retained, excluding model parameters, optimizer states, and activations.

HVP (Hessian–Vector Product) comparison. One HVP $\mathbf{H}_U^{(t)} \sum_i \mathbf{s}_i$ and B_t inner products cost $C_{\mathrm{HVP},t} + \mathcal{O}(B_t p)$. Matrix-free HVPs avoid $\mathcal{O}(p^2)$ Hessian storage but retain higher-order graphs; storing all step vectors adds $\mathcal{O}(B_t p)$ memory. Graph replaces this curvature stage with $\mathcal{O}(B_t k)$ arithmetic while retaining common first-order and optimizer costs.

Greedy selection. Symmetrization retains at most $2nk$ entries, although individual degrees may reach $\mathcal{O}(n)$. Updating all unselected neighbors costs $\mathcal{O}(nk)$ across selection. Full candidate scans give total time $\mathcal{O}(bn + nk)$ and storage $\mathcal{O}(n + nk)$. An optional indexed max-heap reduces maximum lookup and key updates to $\mathcal{O}(\log n)$ each, yielding $\mathcal{O}(n + (b + nk)\log n)$ time with linear initialization; this optimization is not assumed in the reported timings.

D Ablations and Sensitivity

Protocol. We conducted CIFAR-10 mechanism diagnostics with a frozen ImageNet-pretrained ResNet-18 and a trainable linear head. Three data seeds each sampled 1,000 training images with exactly 100 incorrect labels. A fixed, disjoint 1,000-image validation set was drawn from the official training split, with 128 images used for attribution; testing used 2,000 official test images. The classifier used normalized 512-dimensional features extracted at 128×128 input resolution, with 5,130 parameters including biases. Attribution followed five epochs of Adam training with learning rate 10^{-3}, batch size 64, weight decay 10^{-4}, and label smoothing 0.1. Each graph configuration replayed the recorded first-order Adam-aware batch scores, accumulating history across epochs without HVPs or second-order caches. The optional gate used gradient agreement at classifier initialization rather than future gradients or dynamic gate updates; the ungated variant matches the semantic weighting in the main text.

Each selected 200-image subset was retrained for 50 head-training epochs, reducing the learning rate by a factor of ten every ten epochs. Identical subsets within a seed pair reused one fit. Across 43 graph configurations and a random reference, the 396 cells involved 311 distinct subset fits; nine full-data fits provided additional references. Random and full-data references were included as additional controls without changing the parameter grid. Table 7 reports means and sample standard deviations across the seed pairs. Contamination is the selected fraction of mislabeled images; coverage measures the fraction of nodes reached by the selected set in the reference graph.

D.1 Parameter and Module Sensitivity

We evaluated the complete grid $k \in \{5, 10, 20\}$, $\alpha \in \{0, 0.2, 0.4\}$, and $f_\lambda \in \{0, 0.05, 0.1\}$, where f_λ denotes the relative diversity-penalty setting. The default $(10, 0.2, 0.05)$ was fixed before test evaluation. Table 7B summarizes one-factor changes. Increasing α from 0 to 0.4 improves AUROC from 0.571 to 0.708 and reduces contamination from 0.073 to 0.023, while yielding the highest test accuracy of 16.739%. Increasing k to 20 further improves accuracy and AUROC over the default. Varying f_λ mainly affects the diversity-coverage trade-off while leaving AUROC nearly unchanged. Overall, these results show that graph correction strength and neighborhood size effectively control attribution quality and downstream subset selection.

Table 7: CIFAR-10 sensitivity at a 20% budget. Entries are means with sample standard deviations across nine crossed data-training seed pairs. Contam. is selected corruption; coverage uses a fixed reference graph. Panel B varies one parameter from $(k, \alpha, f_\lambda) = (10, 0.2, 0.05)$. Bold/underline mark the best/second-best results within each panel; shading marks the default gated extension.

Configuration	Acc. (%) ↑	AUROC ↑	Contam. ↓	Coverage ↑	Recall@10 ↑
A. Modules and graph construction					
Graph + gate (default)	<u>15.122</u> (3.310)	**0.647** (0.009)	**0.052** (0.012)	0.813 (0.017)	–
Without gate	14.956 (3.285)	0.558 (0.014)	0.071 (0.018)	0.812 (0.019)	–
Without history	15.011 (3.440)	0.584 (0.009)	0.070 (0.018)	**0.836** (0.021)	–
Without pair	14.989 (3.087)	0.636 (0.009)	0.055 (0.014)	0.818 (0.011)	–
ResNet (32×32)	14.717 (3.669)	0.618 (0.008)	0.059 (0.014)	0.833 (0.021)	–
Pooled pixels (8×8)	14.061 (2.905)	0.598 (0.008)	0.066 (0.014)	<u>0.835</u> (0.017)	–
Mean-symmetric	**15.444** (3.440)	<u>0.639</u> (0.009)	<u>0.054</u> (0.013)	0.831 (0.017)	–
Mutual-neighbor	14.356 (3.496)	0.625 (0.009)	0.057 (0.016)	<u>0.835</u> (0.020)	–
B. One-factor parameter changes					
$k = 5$	15.133 (3.322)	0.639 (0.009)	0.055 (0.015)	<u>0.825</u> (0.014)	–
$k = 20$	<u>15.306</u> (2.409)	<u>0.652</u> (0.009)	<u>0.051</u> (0.011)	0.812 (0.019)	–
$\alpha = 0$	15.167 (3.550)	0.571 (0.009)	0.073 (0.019)	**0.840** (0.025)	–
$\alpha = 0.4$	**16.739** (2.622)	**0.708** (0.010)	**0.023** (0.007)	0.798 (0.026)	–
$f_\lambda = 0.1$	14.828 (3.486)	0.647 (0.009)	0.053 (0.013)	0.817 (0.016)	–
C. ANN search budget					
HNSW 2	15.022 (3.420)	0.635 (0.008)	<u>0.054</u> (0.014)	**0.821** (0.018)	0.534 (0.009)
HNSW 8	14.422 (2.845)	<u>0.644</u> (0.008)	<u>0.054</u> (0.013)	<u>0.820</u> (0.010)	0.812 (0.013)
HNSW 32	<u>15.467</u> (3.199)	**0.647** (0.008)	**0.052** (0.012)	0.815 (0.017)	<u>0.945</u> (0.002)
HNSW 128	**15.717** (3.244)	**0.647** (0.009)	**0.052** (0.012)	0.813 (0.017)	**0.989** (0.001)

The module ablations in Table 7A confirm the importance of the graph components. Removing the gate or history reduces AUROC from 0.647 to 0.558 and 0.584, respectively, and increases contamination, while removing the pair term causes a smaller drop. These results show that gradient gating and historical evidence provide gains, with the pair term offering complementary information. Graph variants show that stronger pretrained features improve attribution quality. ResNet features at 128×128 achieve higher AUROC than 32×32 features or pooled pixels. Mean-symmetric propagation achieves the highest test accuracy in Panel A, while the default directed formulation gives the best AUROC and contamination, indicating that graph constructions favor different objectives.

D.2 Approximate Neighbors

Single-threaded HNSW fixed $M = 4$ and the construction budget at 20, while varying the query budget. Increasing the query budget from 2 to 128 raises Recall@10 from 0.534 to 0.989, while AUROC improves from 0.635 to 0.647 and approaches the exact-graph result. At larger budgets, attribution quality remains stable and accuracy improves, with HNSW 128 reaching 15.717%. These results show that approximate neighbor search closely recovers exact-graph behavior without exhaustive retrieval. At very small budgets, missing neighbors can alter graph degree, so the lowest-budget setting approximates both neighbor identity and connectivity.

D.3 Separating Seed Effects

For HNSW with a query search budget of 8, we fully crossed three data, training, and graph seeds. Graph seeds changed insertion order only. Fig. 4 regroups the same 27 conditions by each factor. Fixing the other two factors, the mean accuracy ranges are 4.850, 4.678, and 2.272 percentage points for data, training, and graph seeds, respectively, while the corresponding AUROC ranges are 0.015, 0.017, and only 0.002. Thus, graph-seed variation is substantially smaller than variation from data sampling or model training, indicating that the approximate graph construction is highly stable with respect to insertion order. Training seeds leave ANN recall unchanged because the features are frozen, and exact graphs produce identical scores and selections across nominal graph seeds. Ranking comparisons are restricted to identical data IDs across training seeds.

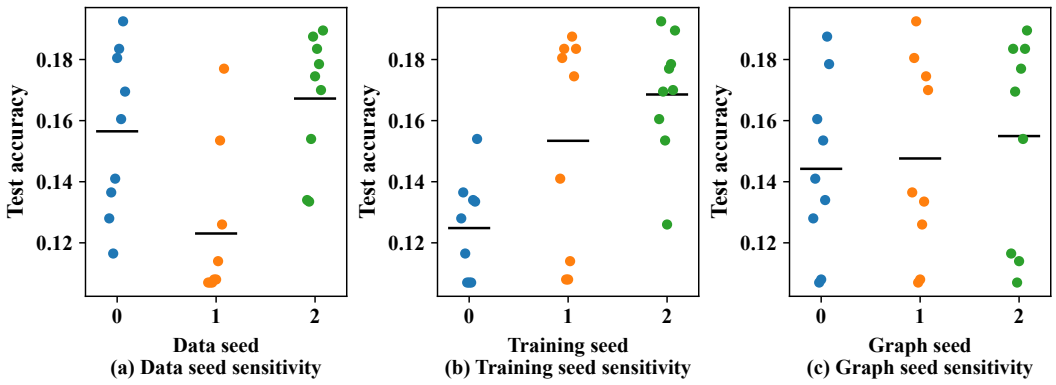

Figure 4: Seed sensitivity on CIFAR-10 with an HNSW query search budget of 8. Panels (a)-(c) show the same 27 seed combinations grouped by data, training, and graph seed, respectively. Each seed level contains nine points for the other two factors; black lines indicate the corresponding mean test accuracy. Horizontal offsets separate points only.

E Stability and Noise Diagnostics

Protocols. Pile diagnostics use the fixed 1,000-document GPT-2 online experiment with three seeds, 128-token inputs, and final-LayerNorm training. Graph uses first-order Adam-aware contributions and semantic interactions without HVPs or gradient gating. CIFAR-10 diagnostics reuse the nine frozen-ResNet-18 head-training trajectories from three data seeds; its Graph extension includes initialization-gradient gating. Both settings contain 10% corrupted training samples. We retain the recorded scores and trajectories without retraining or tuning.

Table 8: Ranking stability and noise detection. For Pile, clean/noisy columns denote training scenarios, with Spearman mean (SD) over three overlapping seed pairs. For CIFAR-10, the noisy column averages within-dataset training-seed correlations, with SD over three data seeds. AUROC SD uses three Pile seeds or nine crossed CIFAR-10 conditions. Oracle rows use known corruption indicators; dashes indicate unreported correlations.

Method	Spearman: clean ↑	Spearman: noisy ↑	Noise AUROC ↑
Pile: three training seeds			
In-Run (2nd)	0.106 (0.020)	0.101 (0.027)	0.494 (0.033)
Adam-aware (2nd)	0.138 (0.020)	0.112 (0.038)	0.507 (0.009)
Adam-aware (1st)	0.137 (0.019)	0.112 (0.038)	0.508 (0.008)
Graph (HVP-free)	0.142 (0.018)	0.115 (0.034)	0.509 (0.012)
Graph + diversity	–	–	0.509 (0.012)
Blocked noise sources (oracle)	–	–	0.536 (0.010)
CIFAR-10: three data seeds crossed with three training seeds			
Adam-aware (1st)	–	0.700 (0.016)	0.571 (0.009)
Graph + gate	–	0.733 (0.014)	0.647 (0.009)
Without gate	–	0.721 (0.015)	0.558 (0.014)
Without history	–	0.703 (0.016)	0.584 (0.009)
Without pair	–	0.727 (0.014)	0.636 (0.009)
Graph + diversity	–	–	0.647 (0.009)
Blocked noise sources (oracle)	–	–	0.647 (0.009)

E.1 Ranking Stability and Score Distributions

Table 8 shows that graph-aware correction consistently improves ranking stability over the corresponding Adam-aware scores. On Pile, Graph achieves the highest Spearman correlation among the evaluated trajectory-based variants, increasing from 0.137 to 0.142 on clean trajectories and from 0.112 to 0.115 under corruption. These consistent gains indicate that neighborhood evidence pro-

vides a complementary structural signal, making sample rankings more stable across independent training runs and different perturbation settings.

The improvement is more pronounced on CIFAR-10, where gated Graph raises the correlation from 0.700 to 0.733. The ablation results further support the contribution of the proposed components. Removing historical neighborhood evidence reduces the correlation to 0.703, whereas removing the gate or pair term yields 0.721 and 0.727, respectively. In particular, the substantial drop without history indicates that accumulated neighborhood evidence plays an important role in stabilizing trajectory-based attribution across repeated training runs.

Fig. 5 further illustrates the effect on noisy samples. Across the nine CIFAR-10 trajectories, Graph consistently shifts mislabeled samples toward lower relative scores while preserving higher rankings for clean samples. Since scores are standardized independently within each trajectory, the comparison reflects systematic changes in relative attribution rather than differences in raw score scale.

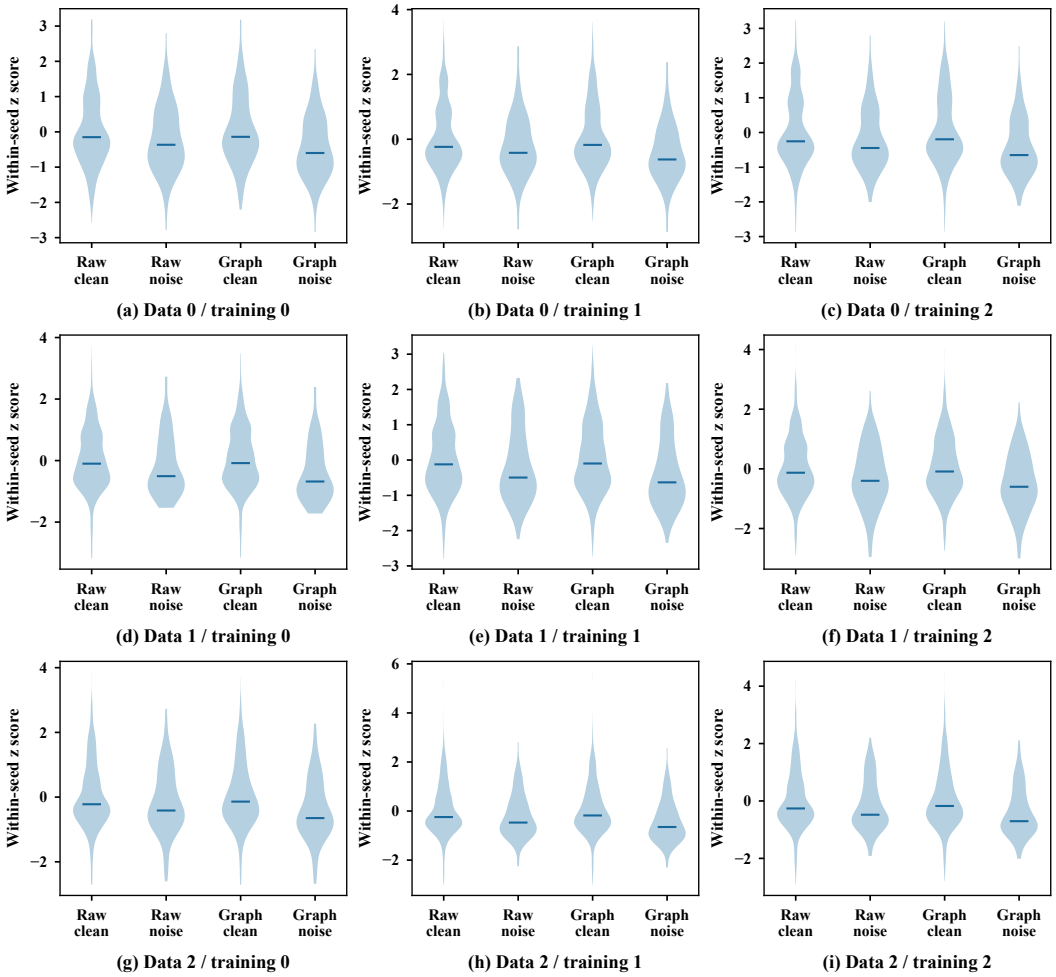

Figure 5: CIFAR-10 score distributions across all nine data-training seed combinations. Each panel compares first-order Adam-aware (Raw) and gated Graph scores for 900 correctly labeled and 100 mislabeled samples. Scores are standardized separately within each method and trajectory; horizontal marks indicate medians. Clean/noise denotes sample status within the same noisy trajectory, not separate training scenarios.

E.2 Noise Selection and Propagation

Graph-aware attribution also improves noise identification and subset quality on CIFAR-10. As shown in Table 8, the noise-detection AUROC increases from 0.571 for first-order Adam-aware

attribution to 0.647 with gated Graph. The benefit is also reflected in the selected subsets. At the 20% retention budget, for example, the corruption fraction decreases from 0.073 to 0.050, and Graph maintains lower contamination than the raw Adam-aware baseline across most retention budgets in Table 9. Diversity-aware selection preserves this overall behavior while additionally encouraging broader coverage of the selected data.

Table 9: Selected corruption fractions at all six retention budgets (lower is better). Parentheses are sample SDs across three Pile seeds or nine crossed CIFAR-10 conditions. The input corruption rate is 0.100. Graph rows without diversity use score ranking; oracle source blocking is diagnostic only.

Method	1%	2%	5%	10%	20%	40%
Pile						
In-Run (2nd)	0.200 (0.173)	0.117 (0.104)	0.127 (0.050)	0.123 (0.021)	0.128 (0.023)	0.106 (0.012)
Adam-aware (2nd)	0.100 (0.000)	0.067 (0.029)	0.120 (0.020)	0.110 (0.036)	0.102 (0.016)	0.093 (0.005)
Adam-aware (1st)	0.133 (0.058)	0.067 (0.029)	0.120 (0.020)	0.113 (0.032)	0.102 (0.016)	0.097 (0.006)
Graph (HVP-free)	0.100 (0.000)	0.050 (0.000)	0.120 (0.020)	0.117 (0.031)	0.098 (0.010)	0.097 (0.007)
Graph + diversity	0.100 (0.000)	0.050 (0.000)	0.120 (0.020)	0.117 (0.031)	0.098 (0.010)	0.098 (0.008)
Blocked noise sources	0.033 (0.058)	0.050 (0.000)	0.093 (0.023)	0.093 (0.032)	0.088 (0.010)	0.092 (0.006)
CIFAR-10						
Adam-aware (1st)	0.056 (0.073)	0.067 (0.043)	0.040 (0.020)	0.052 (0.015)	0.073 (0.020)	0.090 (0.006)
Graph + gate	0.000 (0.000)	0.044 (0.030)	0.031 (0.015)	0.031 (0.011)	0.050 (0.011)	0.071 (0.005)
Without gate	0.033 (0.050)	0.056 (0.039)	0.031 (0.015)	0.047 (0.014)	0.071 (0.017)	0.092 (0.006)
Without history	0.056 (0.073)	0.061 (0.042)	0.038 (0.019)	0.050 (0.017)	0.068 (0.017)	0.088 (0.004)
Without pair	0.022 (0.044)	0.044 (0.030)	0.031 (0.015)	0.033 (0.011)	0.055 (0.014)	0.074 (0.005)
Graph + diversity	0.000 (0.000)	0.044 (0.030)	0.031 (0.015)	0.032 (0.011)	0.052 (0.012)	0.071 (0.005)
Blocked noise sources	0.000 (0.000)	0.044 (0.030)	0.031 (0.015)	0.031 (0.011)	0.050 (0.010)	0.069 (0.005)

One limitation appears on Pile, where noise detection improves only marginally. The propagation statistics in Table 10 provide a plausible explanation. Corrupted recipients obtain 0.930 of their incoming semantic weight from corrupted sources, indicating that corrupted samples can form locally coherent neighborhoods and reinforce one another through graph propagation. In contrast, gradient gating substantially suppresses such propagation on CIFAR-10. Mislabeled samples receive only 0.043 total gated incoming mass, compared with 0.467 for clean samples. This contrast suggests that structural correction is particularly effective when unreliable neighborhood evidence can be identified and attenuated before it affects sample valuation.

Table 10: Propagation diagnostics, averaged over runs. Noise share is the fraction of incoming weight from corrupted sources before/after gating; mass is total gated incoming weight. Promotion is the fraction whose rank improves under Graph; Δq is mean percentile lift (positive is promotion), and Δq_{block} is the change from actual Graph after blocking noisy sources. Pile has no gate. Values rounded to zero need not be exactly zero.

Recipients	Noise share	Gated share	Gated mass	Promotion	Δq	Δq_{block}
Pile / clean	0.018	–	–	0.482	0.000	0.003
Pile / noisy	0.930	–	–	0.463	-0.001	-0.024
CIFAR-10 / clean	0.109	0.010	0.467	0.566	0.008	0.000
CIFAR-10 / noisy	0.105	0.152	0.043	0.021	-0.068	0.000

The source-blocking diagnostic further supports this interpretation. Blocking messages emitted by known corrupted Pile nodes increases AUROC to 0.536 while leaving CIFAR-10 nearly unchanged, providing additional evidence that controlling corrupted-to-corrupted propagation can improve graph-based noise discrimination. This result further highlights the importance of reliable neighborhood evidence during graph correction. The diagnostic uses oracle corruption indicators only for analysis and is not part of the proposed method.

E.3 Neighborhood Structure and Cases

Table 11 provides additional evidence that the graph captures informative local structure. On CIFAR-10, the corruption fraction decreases sharply from 0.460 in the lowest label-agreement bin

to only 0.002 in the highest, accompanied by a rank-percentile change from -0.070 to 0.062. Gradient agreement exhibits nearly the same pattern. These consistent trends show that neighborhoods with stronger semantic and optimization agreement are closely associated with clean samples and positive graph corrections, supporting the relational evidence for refining trajectory-based valuation.

The degree and density statistics reveal a complementary effect: samples in denser neighborhoods generally receive larger positive rank adjustments, consistent with the proposed use of neighborhood evidence to stabilize isolated trajectory-based estimates. Together with the ablation results, these diagnostics indicate that the benefit of Graph arises not merely from smoothing scores, but from exploiting structured agreement among related samples.

Table 11: Lowest- and highest-quantile neighborhood diagnostics, averaged equally over all runs. Columns report within-bin corruption fraction and mean rank-percentile lift. Quantiles are defined within each run; tied boundaries are merged. These are descriptive associations, not causal effects. All intermediate bins remain in the source results.

Neighborhood property	Low: noise	Low: Δq	High: noise	High: Δq
Pile / Domain agreement	0.287	-0.001	0.000	0.001
Pile / Gradient agreement	0.045	0.001	0.330	-0.004
Pile / Degree	0.059	0.000	0.092	0.001
Pile / Density	0.058	-0.006	0.205	0.003
CIFAR / Label agreement	0.460	-0.070	0.002	0.062
CIFAR / Gradient agreement	0.458	-0.070	0.002	0.063
CIFAR / Degree	0.097	-0.011	0.118	0.013
CIFAR / Density	0.092	-0.022	0.113	0.026

The qualitative cases in Figs. 6 and 7 further illustrate how graph correction changes individual rankings. On CIFAR-10, the most strongly demoted mislabeled example moves from rank 458 to 638, whereas the most promoted mislabeled example changes only from 883 to 882. This asymmetry is consistent with the aggregate noise-detection results, suggesting that gated graph correction primarily suppresses unreliable samples rather than indiscriminately amplifying their neighbors. On Pile, where gradient gating is not applied, corrupted samples can receive stronger support from semantically similar corrupted neighbors, explaining the less selective rank corrections observed in the qualitative cases. Overall, the visualized neighborhoods illustrate both the structural evidence exploited by graph correction and the role of gating in controlling unreliable propagation.

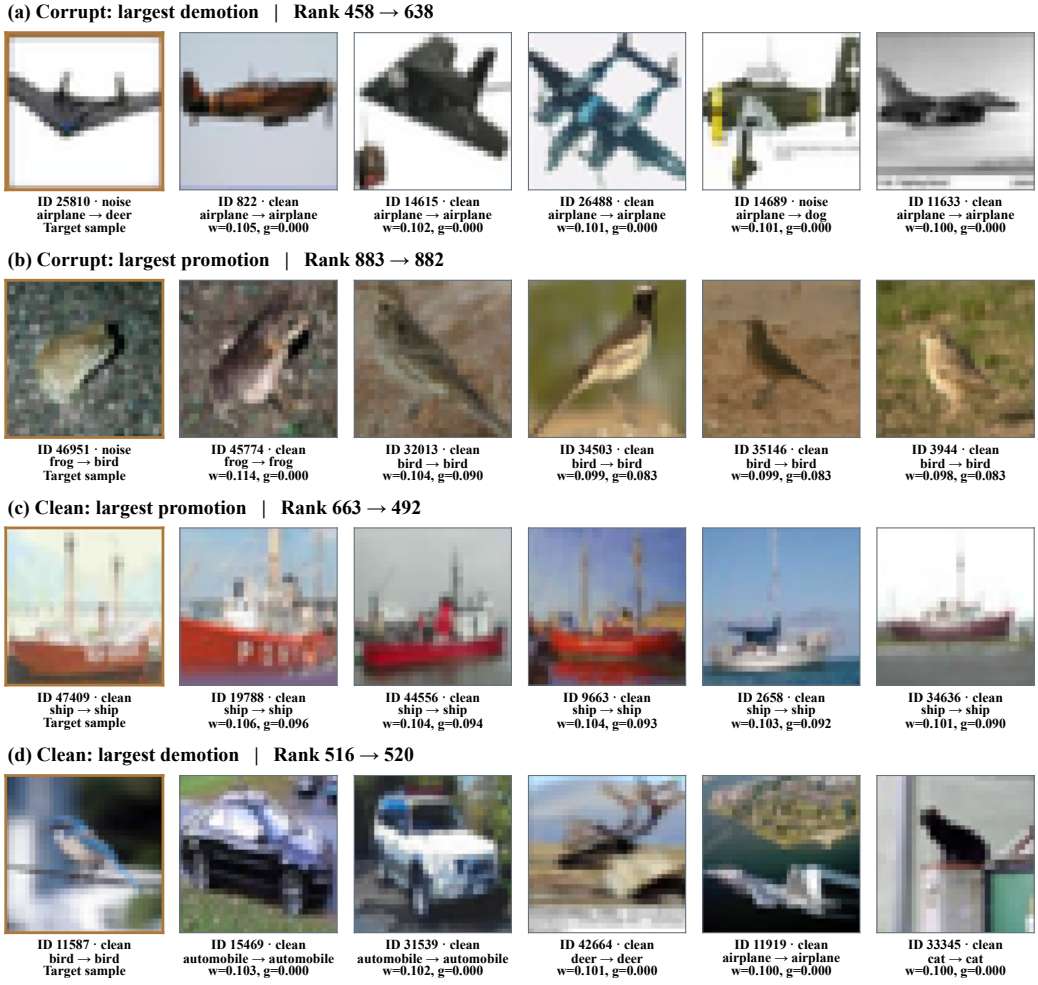

Figure 6: CIFAR-10 extreme neighborhood cases at data/training seed 0. Each row shows one target followed by its five highest-weight semantic neighbors. Labels read true class → observed class; w is semantic weight and g is effective gated weight. Lower ranks indicate higher value. Images are enlarged directly from the original 32×32 arrays without synthesized detail.



Figure 7: Pile rank-change extrema at seed 0. Cards show actual-input excerpts, clean counterparts for corrupted inputs or the highest-weight neighbor for clean inputs, and the closest validation query. Ellipses mark omitted text, including the non-English prefix in panel (c). All excerpts come directly from recorded inputs of at most 128 tokens. Domain labels refer to the original data sources, and lower ranks indicate higher estimated value.