WHICH INFLUENCE ARE WE ESTIMATING? THE ROLE OF COUNTERFACTUAL SPECIFICATIONS IN DATA ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating the influence of training examples on model behavior is essential for data debugging, valuation, and attribution. Existing influence estimators often produce incompatible rankings, which are commonly ascribed to approximation error. We argue that a more fundamental source of disagreement is specification mismatch: influence depends on the behavior being attributed, the intervention applied to each training example, and the counterfactual training process that maps the intervention to a model response. These choices are especially important when the target behavior requires a tractable surrogate, such as query loss, a logit, or a margin. We formalize influence as a counterfactual estimand, distinguish specification mismatch across estimands from approximation error in estimating a fixed estimand, and organize representative estimators by their implied specifications. We further derive a local decomposition that exposes how behavior signals, training signals, and counterfactual parameter responses interact. Controlled experiments show that exact estimands under different specifications can induce different rankings, whereas approximation error grows as perturbations move farther from their linearization points. Experiments on noisy label detection and LLM attribution show that specification choices significantly affect attribution quality, especially for the choice of behavior surrogate. Behavior-aligned specifications can identify target-specific training examples obscured by default loss-based or similarity-based specifications. These results establish specification analysis as a necessary first step for interpreting and comparing data influence estimators.

1 INTRODUCTION

Understanding how training examples affect model behavior is a fundamental problem in modern machine learning, with applications in model debugging (Koh & Liang, 2017; Yeh et al., 2018; Pruthi et al., 2020), data selection (Xia et al., 2024; Zhao et al., 2025; Dai et al., 2025), and safety analysis (Lin et al., 2024; Coalson et al., 2025; Li et al., 2026b). A wide range of approaches for estimating the influence of training data (i.e., influence estimators) has been proposed, from retraining-based methods such as leave-one-out evaluation (Molinaro et al., 2005) and data Shapley values (Ghorbani & Zou, 2019; Kwon & Zou, 2021) to more scalable methods based on feature similarity (Yeh et al., 2018; Hanawa et al., 2021; Pezeshkpour et al., 2021; Wang et al., 2024; Yang et al., 2025b; Li et al., 2026b) or gradient information (Hampel, 1974; Ling, 1984; Koh & Liang, 2017; Kwon et al., 2024; Park et al., 2023; Lin et al., 2024). Despite their widespread use, these methods often produce incompatible influence rankings on the same model and dataset (Bae et al., 2022; Basu et al., 2021; Li et al., 2025; Aberger et al., 2025; Li et al., 2026a), as exemplified in Figure 1. This disagreement is particularly difficult to interpret for large language models (LLMs), where retraining-based references are often prohibitively expensive.

These discrepancies are commonly attributed to approximation error since many practical methods replace retraining with local linearizations or cheaper similarity-based scores. However, influence estimators may in fact target different specifications of influence, including the behavior being attributed, the intervention applied, and the counterfactual training process. Changing any of these choices changes the counterfactual question and can alter the influence ranking even when the corresponding quantity is computed exactly. Among these choices, behavior is especially important

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.

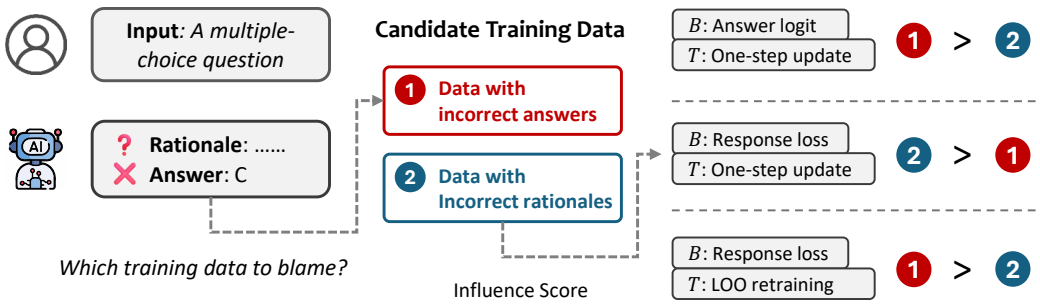


Figure 1: Influence rankings depend on the counterfactual specification. Different behavior surrogates and counterfactual training processes can induce different rankings for the same query.

when the target behavior is non-differentiable. Practical methods therefore rely on tractable but potentially problematic surrogates, such as query loss or a logit. Query loss remains a common default, but whether it captures the target behavior is often left unexamined. A mismatched surrogate can lead to poor attribution even when the estimator accurately estimates its specified quantity.

To clarify these ambiguities, we introduce an influence specification framework that disentangles specification mismatch from approximation error. *Specification mismatch* occurs when estimators target different counterfactual quantities, while *approximation error* occurs when estimators target the same specified quantity but approximate it with different levels of accuracy. Within our framework, the intervention defines what is changed in the training data or objective, whereas the counterfactual training process maps this change to a counterfactual model and determines the resulting parameter response through which the behavior change is measured. We then organize representative influence estimators according to their specifications, and extend a local decomposition of learning dynamics (Ren & Sutherland, 2025) to counterfactual training processes and expose how behavior signals, training signals, and counterfactual parameter responses interact.

We evaluate these distinctions in controlled experiments and practical attribution tasks. Controlled experiments show that methods with different specifications can induce different influence rankings, while approximation error under a fixed specification grows as local approximations move farther from their reference points. Experiments on noisy label detection and LLM attribution further show that behavior specification affects attribution quality and downstream data decisions. In particular, behavior-aligned surrogates can reveal target-specific training examples obscured by default loss-based or similarity-based specifications. These results establish specification analysis as a necessary first step for interpreting and comparing data influence estimators.

Our main contributions are as follows:

- We formalize data influence as a counterfactual estimand and distinguish specification mismatch from approximation error as different sources of ranking disagreement.
- We organize representative influence estimators according to their implied specifications and extend a local decomposition of learning dynamics to counterfactual training processes.
- We show through controlled experiments that changing the specification can alter exact influence rankings, whereas approximation error under a fixed specification distorts rankings as local approximations move away from their reference points.
- We demonstrate on noisy label detection and LLM attribution that behavior-aligned surrogates can improve attribution quality and downstream data decisions, while default loss-based or similarity-based specifications can obscure target-specific training examples.

2 Influence as a Counterfactual Estimand

In this section, we formalize data influence as a counterfactual estimand, which measures how a specified model behavior would change under an intervention associated with a training example. Let $\mathcal{D} = \{z_i\}_{i=1}^n$ denote the training set, q a query of interest, and $\theta_{\mathcal{D}}$ the factual model obtained by

training on $\mathcal{D}$. We use $B(q;\theta) \in \mathbb{R}$ to denote the operationalized behavior whose change we seek to attribute. This behavior may be the target behavior itself or a tractable surrogate such as a loss.

Influence Specification. Let $\mathcal{P}_k$ denote an intervention on z_k, such as upweighting z_k or modifying its training objective. Let $\mathcal{T}$ denote a specified counterfactual training process, such as retraining or a local update. Under $\mathcal{P}_k$, the process $\mathcal{T}$ produces a counterfactual model $\theta_k^{\mathrm{cf}} \sim \mathcal{T}(\mathcal{D}, \theta_{\mathcal{D}}, \mathcal{P}_k)$. To characterize existing approaches on data influence estimation, we define an influence specification as $\mathcal{S} = (B, \mathcal{P}, \mathcal{T})$ where $\mathcal{P} = \{\mathcal{P}_k\}_{k=1}^n$ denotes the family of example-specific interventions. The influence of z_k on the behavior measured at q is the counterfactual estimand:

$$I_{\mathcal{S}}(q,k) = \mathbb{E}\left[B(q;\theta_k^{\mathrm{cf}}) - B(q;\theta_{\mathcal{D}})\right]. \tag{1}$$

The expectation accounts for randomness in training or evaluation when present. Equation 1 makes explicit that influence is defined relative to $\mathcal{S}$. Each choice of $\mathcal{S}$ determines the answer of a counterfactual question. Changing $\mathcal{S}$ can alter the exact influence ranking even when the corresponding estimand is computed exactly. We adopt the convention that positive influence means that the specified intervention increases B. For existing methods, we may flip the sign to keep the definition of influence consistent. For example, under data removal, a negative influence means that removing z_k decreases B, indicating that z_k supports the measured behavior in the factual model.

Estimands and Estimators. Given an estimand $I_{\mathcal{S}}(q,k)$ defined above, an influence estimator $\widehat{I}_{\mathcal{S}}(q,k)$ is a method for approximating it (since computing the estimand exactly is often computationally infeasible). Under a fixed specification, any discrepancy between the estimator and the estimand constitutes approximation error. For a candidate index set $\mathcal{K} \subseteq \{1,\dots,n\}$, let

$$\pi_{\mathcal{S}}(q) = \mathrm{argsort}_{k\in\mathcal{K}}\, I_{\mathcal{S}}(q,k), \qquad \widehat{\pi}_{\mathcal{S}}(q) = \mathrm{argsort}_{k\in\mathcal{K}}\, \widehat{I}_{\mathcal{S}}(q,k) \tag{2}$$

denote the exact and estimated influence rankings in descending order. When $\mathcal{S}$ is fixed, disagreement between $\pi_{\mathcal{S}}(q)$ and $\widehat{\pi}_{\mathcal{S}}(q)$ measures how approximation error affects the ranking. In contrast, exact rankings under different $\mathcal{S}$ may disagree as they answer different counterfactual questions.

When two methods share the same specification, disagreement between their estimated rankings implies approximation error in at least one method, but does not identify the more accurate estimator. Influence estimators should therefore be compared against references defined by the same specification before ranking disagreement is interpreted as estimation failure.

3 Counterfactual Specifications of Influence Estimators

In this section, we characterize existing popular influence estimators through the specification $\mathcal{S} = (B, \mathcal{P}, \mathcal{T})$. We first show that similarity-based estimators admit an exact counterfactual interpretation under a standardized local update. We then derive a local decomposition of learning-based estimators, in which behavior and training signals interact through the parameter response induced by the counterfactual process. Table 1 summarizes the specifications of representative estimators.

Similarity-Based Estimators. Similarity-based estimators measure the affinity between a query and a training example in a chosen feature space. Given a feature map ϕ, a common score is $\langle \phi(q), \phi(x_k) \rangle$, as used in representation similarity methods (Hanawa et al., 2021; Pezeshkpour et al., 2021; Yang et al., 2025b). For a scalar model output, choosing $\phi(x) = \nabla_\theta f_\theta(x)|_{\theta=\theta_{\mathcal{D}}}$ yields the scalar neural tangent kernel (Jacot et al., 2018; Chen et al., 2021; Nguyen et al., 2021).

Although similarity scores do not explicitly simulate a training process, they admit a counterfactual interpretation under a constructed local specification. Consider a frozen-feature linear readout $f_w(x) = w^\top \phi(x)$ with query behavior $B(q;w) = f_w(q)$. Define $\mathcal{P}_k^{\mathrm{sim}}$ through the label-agnostic auxiliary loss $\ell_k^{\mathrm{sim}}(w) = -f_w(x_k)$, and let $\mathcal{T}_{\mathrm{sim}}$ apply one gradient step from the factual readout $w_{\mathcal{D}}$, yielding an update in the direction $\phi(x_k)$ as

$$w_k^{\mathrm{cf}} = w_{\mathcal{D}} - \eta \cdot \nabla_w \ell_k^{\mathrm{sim}}(w_{\mathcal{D}}) = w_{\mathcal{D}} + \eta \cdot \phi(x_k), \tag{3}$$

where $\eta > 0$ is shared across training examples. The resulting change in query behavior is

$$I_{\mathcal{S}_{\mathrm{sim}}}(q,k) = B(q;w_k^{\mathrm{cf}}) - B(q;w_{\mathcal{D}}) = \eta \cdot \langle \phi(q), \phi(x_k) \rangle. \tag{4}$$

Thus, up to the common positive scale η, feature similarity is the exact influence under the constructed specification $\mathcal{S}_{\mathrm{sim}} = (B, \mathcal{P}^{\mathrm{sim}}, \mathcal{T}_{\mathrm{sim}})$. This construction isolates feature geometry through

Table 1: Representative influence estimators and their implied counterfactual specifications.

Estimator	**Behavior** B	**Intervention** $\mathcal{P}$	**Training process** $\mathcal{T}$
Feature similarity	Query readout	Add auxiliary loss	One-step update
Gradient similarity	Query loss	Upweight z_k	One-step update
Trajectory-based	Query loss	Upweight z_k	Trajectory aggregation
Influence functions	Query loss	Infinitesimally upweight z_k	Local re-optimization
Leave-one-out	Query behavior	Remove z_k	Retraining

a readout update determined solely by $\phi(x_k)$, making explicit the behavior and intervention associated with the similarity score. A positive inner product increases the query readout under this standardized update, whereas a negative inner product decreases it.

Learning-Based Estimators. Unlike similarity-based estimators, learning-based estimators use the training objective to determine each example's update direction and combine it with a query behavior signal (Pruthi et al., 2020; Park et al., 2023; Lin et al., 2024; Xia et al., 2024; Wang et al., 2026).

We first consider a one-step gradient counterfactual. Let $\ell(z_k;\theta)$ be the per-example training loss. Let $\mathcal{P}_k^+$ denote an upweighting intervention that adds one occurrence of z_k to an auxiliary update objective. Let $\mathcal{T}_{\text{grad}}$ apply one-step gradient descent $\theta^{\text{cf}}_{k,\text{grad}} = \theta_{\mathcal{D}} - \eta \cdot \nabla_\theta \ell(z_k;\theta_{\mathcal{D}})$, where any objective normalization is absorbed into $\eta > 0$. For a differentiable query behavior $B(q;\theta)$, a first-order Taylor expansion around $\theta_{\mathcal{D}}$ gives

$$I_{\mathcal{S}_{\text{grad}}}(q,k) = B\big(q;\theta^{\text{cf}}_{k,\text{grad}}\big) - B(q;\theta_{\mathcal{D}}) = -\eta \cdot \langle \nabla_\theta B(q;\theta_{\mathcal{D}}), \nabla_\theta \ell(z_k;\theta_{\mathcal{D}})\rangle + R_k, \tag{5}$$

where R_k is the Taylor remainder. Dropping R_k yields the corresponding first-order estimator. Compared with Equation 4, the update direction here is determined by the training objective and therefore depends on the training target and the current model prediction.

Following the local decomposition of learning dynamics (Ren & Sutherland, 2025), we factor parameter gradients through the model outputs. Let $f_\theta(x) \in \mathbb{R}^m$ denote the model output and $J_x = \nabla_\theta f_\theta(x) \in \mathbb{R}^{m\times d}$ its parameter Jacobian. By the chain rule, it gives $\nabla_\theta \ell(z_k;\theta_{\mathcal{D}}) = J_{x_k}^\top s_\ell(z_k)$ and $\nabla_\theta B(q;\theta_{\mathcal{D}}) = J_q^\top s_B(q)$ where $s_\ell(z_k) = \nabla_{f_\theta(x_k)}\ell(z_k;\theta_{\mathcal{D}})$ and $s_B(q) = \nabla_{f_\theta(q)} B(q;\theta_{\mathcal{D}})$ denote the example-specific learning and behavior signal. Substituting these expressions into Equation 5 gives the first-order estimator

$$\widehat{I}_{\mathcal{S}_{\text{grad}}}(q,k) = -\eta \cdot s_B(q)^\top K_{\theta_{\mathcal{D}}}(q,x_k)\, s_\ell(z_k), \qquad K_{\theta_{\mathcal{D}}}(q,x_k) = J_q J_{x_k}^\top. \tag{6}$$

Equation 6 separates the behavior signal, Jacobian interaction, and training signal. For scalar outputs (or can be reduced to a scalar), $K_{\theta_{\mathcal{D}}}(q,x_k)$ is the inner product of the corresponding Jacobian features, linking this expression to Equation 4. Thus, learning-based scores can be viewed as Jacobian similarities conditioned on the query behavior and training objective.

The same perspective extends to general counterfactual processes through their induced parameter paths. Let $\mathcal{P}_k(\alpha)$ interpolate between the factual setting and the full intervention, and let $\theta_k(\alpha)$ denote the corresponding model path, with $\theta_k(0) = \theta_{\mathcal{D}}$ and $\theta_k(1) = \theta_k^{\text{cf}}$. For a differentiable path and behavior, the endpoint difference in Equation 1 can be written as

$$I_{\mathcal{S}}(q,k) = \mathbb{E}\left[\int_0^1 \nabla_\theta B(q;\theta_k(\alpha))^\top \frac{d\theta_k(\alpha)}{d\alpha}\, d\alpha\right]. \tag{7}$$

Equation 7 accumulates local interactions between the behavior gradient and the parameter response along the counterfactual path. For example, consider an upweighting path $L_{\mathcal{P}_k(\alpha)}(\theta) = L_{\mathcal{D}}(\theta) + \alpha \cdot \ell(z_k;\theta)$, the derivative of the training gradient with respect to α at $\alpha = 0$ is $\nabla_\theta \ell(z_k;\theta_{\mathcal{D}})$. If the local parameter response is linear in this perturbation, we have

$$\left.\frac{d\theta_k(\alpha)}{d\alpha}\right|_{\alpha=0} = -\mathcal{R}_{\mathcal{T}} \cdot \nabla_\theta \ell(z_k;\theta_{\mathcal{D}}). \tag{8}$$

Here $\mathcal{R}_{\mathcal{T}}$ maps the training gradient perturbation to the negative first-order parameter response under $\mathcal{T}$. A one-step update has $\mathcal{R}_{\mathcal{T}} = \eta \cdot I$ where I is the identity operator. Local re-optimization along

a differentiable stationary point branch with invertible Hessian $H_{\mathcal{D}}$ has $\mathcal{R}_{\mathcal{T}} = H_{\mathcal{D}}^{-1}$, recovering the classical influence function response (Hampel, 1974; Ling, 1984; Koh & Liang, 2017). Trajectory-based methods extend this structure across checkpoints by accumulating gradient interactions or propagating perturbations through unrolled update Jacobians (Pruthi et al., 2020; Bae et al., 2024). Substituting Equation 8 into the integrand of Equation 7 at the factual model gives

$$\widehat{I}_{\mathcal{S}}(q,k) = -s_B(q)^\top K_{\mathcal{T}}(q,x_k)\, s_\ell(z_k), \qquad K_{\mathcal{T}}(q,x_k) = J_q \mathcal{R}_{\mathcal{T}} J_{x_k}^\top, \tag{9}$$

where $K_{\mathcal{T}}(q,x_k)$ captures the transition-conditioned interaction between the query and training example. Equation 9 generalizes the one-step decomposition in Equation 6 to processes admitting the local response in Equation 8, showing that the training process determines how behavior and training signals interact. Equation 7 is the exact path-level representation, whereas Equation 9 is its first-order approximation obtained by linearizing at the factual model. Taken together, influence estimators share a common local structure. Representation similarity drops both signals, and restoring them recovers much of the gap to learning-based estimators as shown in Appendix B.2.

4 Controlled Experiments on Influence Specification

In this section, we study how influence specifications affect influence rankings. We first compare exact counterfactual quantities under different specifications, then fix $\mathcal{S}$ to measure approximation error relative to specification-matched references.

Controlled Setting. We use FashionMNIST (Xiao et al., 2017) with an ℓ_2-regularized multinomial logistic regression model, whose smooth strongly convex objective has a unique global optimum, enabling accurate computation of Hessian and finite-perturbation references. Starting from this optimum, we vary one component of $\mathcal{S} = (B, \mathcal{P}, \mathcal{T})$ at a time. We consider four behavior measures (query loss, logit, soft margin, and hard margin), finite upweighting interventions $\mathcal{P}_k(\alpha)$, and counterfactual processes including one-step updates, unrolled trajectories, inverse-Hessian responses, and Newton re-optimization. We measure ranking agreement using Kendall's τ (Kendall, 1938). We additionally perform a non-convex check on CIFAR-10 (Krizhevsky et al., 2009) with a simple CNN. Detailed configurations are provided in Appendix A.1.

Finding 1: Changing the specification changes the question.

Different choices of B, $\mathcal{P}$, or $\mathcal{T}$ can produce different influence rankings since they answer different counterfactual questions.

Figure 2(a) isolates specification mismatch by varying one component of $\mathcal{S}$ at a time. Along the behavior axis, we fix the intervention and one-step update, and compare exact behavior changes under soft margin, hard margin, and query logit with those under query loss. Soft margin produces the same ranking as query loss[1], whereas hard margin and query logit produce significantly different rankings. Along the intervention axis, we fix the query loss and counterfactual process, and compare exact leave-one-out removal with finite upweighting at different perturbation scales. The rankings diverge as the intervention changes, showing that removing and reweighting a training example generally define different influence quantities, even when both counterfactual models are computed exactly. Along the training process axis, five-step unrolled responses remain close to one-step responses, whereas the infinitesimal re-optimization response represented by the inverse Hessian is nearly uncorrelated with the one-step estimand. Thus, changing any component of the specification can significantly alter the exact influence ranking. Influence methods should be compared only after their intended reference specifications are aligned.

Finding 2: Fixing the specification isolates approximation error.

When an approximate score and its reference share the same specification, ranking disagreement reflects approximation error and grows as the counterfactual moves farther from the linearization point.

[1]The soft margin and query loss agree exactly because they are related by a strictly monotonic transformation. See Proposition 1 for details.

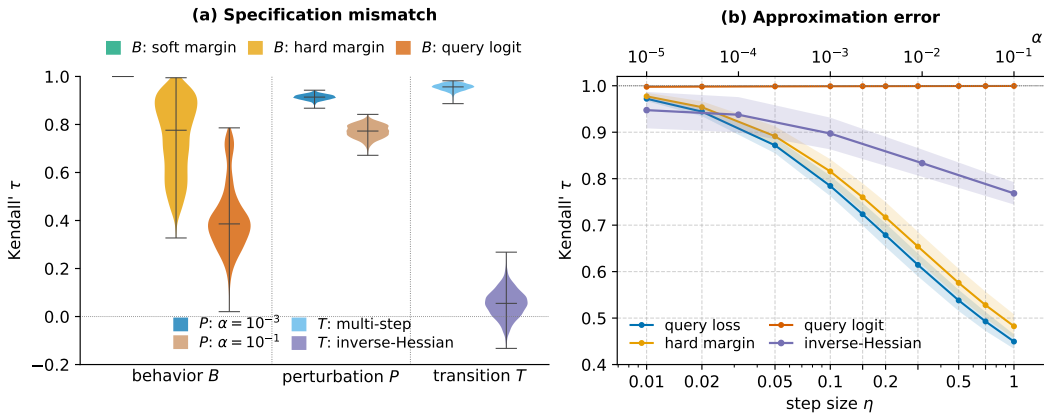


Figure 2: Controlled experiments disentangle specification mismatch from approximation error. (a) Exact rankings vary across B, $\mathcal{P}$, and $\mathcal{T}$ specifications. (b) With the specification fixed, ranking agreement decreases with step size for nonlinear behaviors and perturbation scale for inverse-Hessian responses. Results are averaged over multiple queries and three random seeds.

Figure 2(b) illustrates approximation error by comparing each approximate score with its specification-matched reference. For one-step updates, the reference evaluates the behavior after a parameter update of size η, whereas the approximate score uses the first-order Taylor expansion in Equation 5. Ranking agreement decreases with η for nonlinear behaviors such as query loss and hard margin, but remains high for query logit, which is linear in the parameter update. For re-optimization, the reference evaluates the behavior after Newton re-optimization under a finite upweighting of scale α, whereas the inverse-Hessian score linearizes the response around the unperturbed optimum. Ranking agreement decreases with α as the perturbed optimum moves farther from this linearization point. These results show that approximation error depends on both behavior curvature and the distance between the counterfactual state and the linearization point.

In summary, ranking disagreement is informative about approximation quality only when the underlying specifications are aligned. When a computable reference is available, fixing $\mathcal{S}$ isolates approximation error. When direct counterfactual evaluation is infeasible, specifying the behavior, intervention, and counterfactual process remains essential for interpreting influence rankings. We observe the same qualitative patterns in a supplementary non-convex check on CIFAR-10 with a CNN. Full experimental results and analysis are provided in Appendix B.1.

5 Specification Effects in Practical Attribution

In this section, we study how influence specifications affect attribution in practical tasks where accurate per-example counterfactual references are computationally infeasible. We evaluate specification choices via noisy label detection and LLM attribution, measuring both the identification of problematic examples and model behavior after removing selected data. As attribution utility depends on the target task, we evaluate practical specification choices across several attribution settings.

5.1 Noisy Label Detection

Noisy label detection identifies training examples with incorrect labels, which may alter predictions on correctly labeled data. A trusted set with verified labels therefore provides a reference for identifying training examples estimated to undermine correct predictions. We rank candidates by their estimated harmful effect on the trusted behavior and inspect or remove the highest-ranked examples.

Setup. We train ResNet-18 on 45,000 CIFAR-10 examples with symmetric label noise across rates 0.05 to 0.4. A disjoint clean set of 5,000 trusted examples, with 500 per class, defines the trusted behavior. We evaluate GradSim, TracIn (Pruthi et al., 2020), and mini-batch LiSSA (Koh & Liang, 2017), representing one-step, trajectory-based, and inverse-Hessian responses, respectively. Each

Table 2: Noisy label detection on CIFAR-10 with label noise at $\rho = 0.2$. Results are averaged over three training seeds with a fixed corruption mask. Blue shading marks the preferred behavior surrogate for each method and metric, and bold indicates the best result in each column.

Method	Trusted behavior	AUPRC	AUROC	P@1%	P@5%	P@10%
RepSim	/	0.216 ± 0.008	0.532 ± 0.013	0.246 ± 0.026	0.223 ± 0.013	0.219 ± 0.016
NTK	/	0.197 ± 0.004	0.475 ± 0.011	0.258 ± 0.024	0.221 ± 0.022	0.207 ± 0.012
Random	/	0.198 ± 0.001	0.497 ± 0.001	0.194 ± 0.011	0.191 ± 0.005	0.197 ± 0.004
GradSim	Trusted loss	0.401 ± 0.017	0.653 ± 0.020	0.843 ± 0.042	0.628 ± 0.026	0.511 ± 0.019
	Target logit	0.345 ± 0.022	0.598 ± 0.014	**0.995 ± 0.001**	0.527 ± 0.051	0.389 ± 0.028
	Hard margin	0.402 ± 0.068	0.638 ± 0.027	0.899 ± 0.101	0.646 ± 0.145	0.514 ± 0.100
TracIn	Trusted loss	0.349 ± 0.027	0.501 ± 0.025	0.810 ± 0.046	0.690 ± 0.077	0.537 ± 0.037
	Target logit	**0.602 ± 0.031**	0.716 ± 0.016	0.970 ± 0.018	0.920 ± 0.030	**0.831 ± 0.024**
	Hard margin	0.468 ± 0.038	0.604 ± 0.042	0.902 ± 0.023	0.826 ± 0.036	0.704 ± 0.059
LiSSA	Trusted loss	0.535 ± 0.025	**0.759 ± 0.011**	0.964 ± 0.016	0.798 ± 0.042	0.651 ± 0.034
	Target logit	0.421 ± 0.013	0.642 ± 0.008	**0.995 ± 0.005**	0.708 ± 0.033	0.492 ± 0.017
	Hard margin	0.587 ± 0.020	0.742 ± 0.010	0.992 ± 0.006	**0.940 ± 0.028**	0.781 ± 0.022

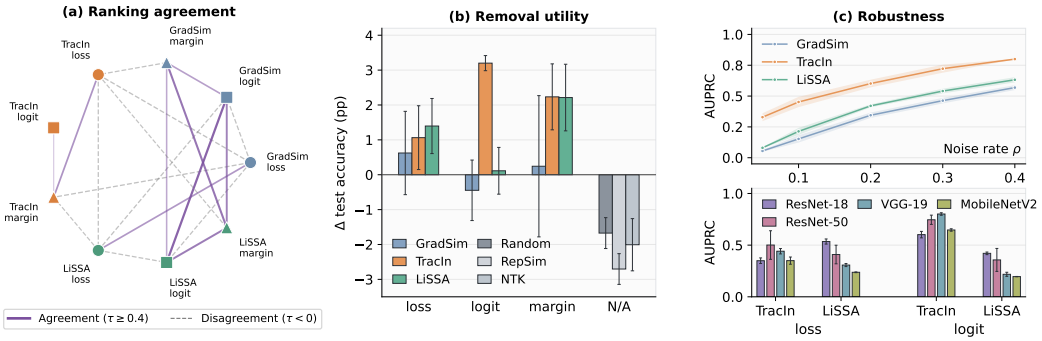


Figure 3: Influence specification affects noisy label detection. (a) Ranking agreement across methods with different behavior specifications. (b) Test accuracy after removing the top 10% of ranked training examples and retraining. (c) Robustness across noise rates and model architectures.

method is paired with trusted loss, target logit, and hard margin as behavior surrogates. Random selection, representation similarity, and NTK similarity serve as baselines. We report AUPRC, AUROC, and top-k precision for corruption detection, together with test accuracy changes after removal and retraining. To avoid label leakage, training examples are scored using their observed labels, and corruption indicators are used only for evaluation. Results are averaged over three training seeds with a fixed corruption mask. Implementation details are provided in Appendix A.2.

Results. Table 2 reports detection performance at noise rate $\rho = 0.2$. Representation and NTK similarity perform close to random selection as they do not explicitly leverage the behavior and training signals provided by the labels. In contrast, learning-based estimators generally achieve higher detection performance. However, we observe that the choice of trusted behavior largely affects the ranking produced by the same method, and the preferred behavior surrogate depends on the estimator itself. For TracIn, replacing trusted loss with target logit increases AUPRC from 0.349 to 0.602 and P@10% from 0.537 to 0.831. For LiSSA, the same replacement reduces AUPRC from 0.535 to 0.421, while hard margin achieves its highest AUPRC of 0.587 and P@10% of 0.781. Although loss change is commonly adopted as the default surrogate in previous studies (Koh & Liang, 2017; Ren et al., 2018; Pruthi et al., 2020; Evans et al., 2024), it often falls short of delivering the strongest attribution performance and can obscure the effectiveness of an estimator under other behavior specifications. Different estimators may therefore suit different behavior surrogates across tasks, and no single surrogate, including the default loss, is universally optimal.

Figure 3(a) shows that two specifications yield more similar rankings when they share more components, whereas changing either component can lead to significant disagreement. These differences translate into downstream removal utility. As shown in Figure 3(b), removing the top 10% of exam-

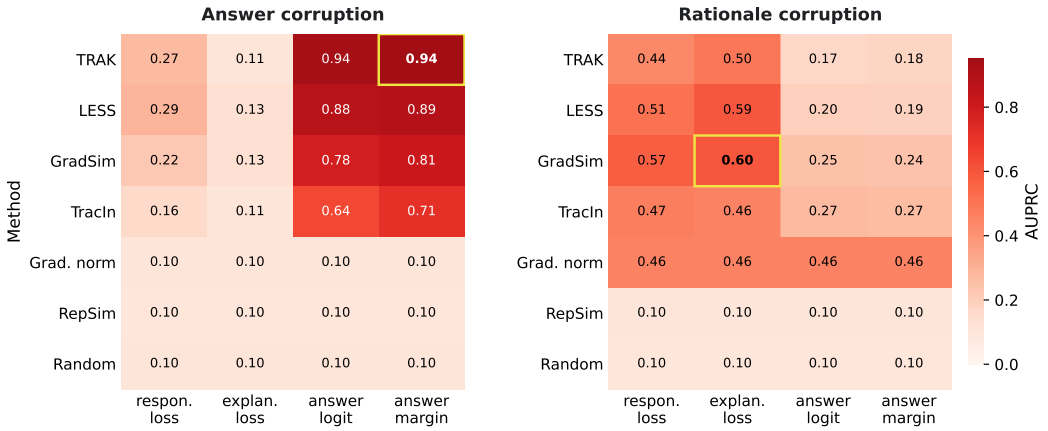


Figure 4: Response-corruption attribution with Qwen3-8B on ScienceQA. Cells report mean AUPRC over three training seeds for detecting answer-corrupted and rationale-corrupted examples. Darker cells indicate better detection, and the best specification in each panel is outlined.

ples selected by TracIn improves test accuracy by 3.20 percentage points with target logit, compared with 1.06 points with trusted loss. LiSSA with hard margin improves accuracy by 2.21 points, while random selection and similarity baselines reduce accuracy. Figure 3(c) shows that TracIn consistently benefits from target logit, while LiSSA generally favors trusted loss or hard margin across noise rates and model architectures. These results confirm that the best surrogate for each estimator also delivers the largest downstream gain, making surrogate choice a practical concern. We provide full experimental results in Appendix B.2.

5.2 LLM Attribution

Identifying the training examples responsible for LLM behaviors is increasingly important, since these behaviors are diverse and context-dependent and the choice of behavior surrogate is therefore critical for attribution. We accordingly study how behavior specification affects attribution in two practical settings, namely response corruption and conditional backdoor attribution.

Setup. We fine-tune Qwen3-8B (Yang et al., 2025a) on text-only ScienceQA (Lu et al., 2022), which provides questions, answer choices, and rationales. In response-corruption attribution, the training set contains clean examples, examples with incorrect answers, and examples with mismatched rationales. We evaluate whether attribution can separately identify the two corrupted groups. In conditional backdoor attribution, poisoned examples associate a trigger in an activating context with an incorrect target answer. Benign-trigger examples contain the same triggers in nonactivating contexts and retain correct answers. Attribution should identify harmful poison while excluding benign-trigger and clean examples. We evaluate GradSim, TracIn, TRAK (Park et al., 2023), and LESS (Xia et al., 2024), each paired with four behavior surrogates. Response loss averages token losses over the full response, and explanation loss restricts this average to rationale tokens. Answer logit measures the specified answer token, and answer margin measures its advantage over the strongest competing choice. Gradient norm, representation similarity, and random selection serve as baselines. We report AUPRC separately for each corruption type, treating all other examples as negatives. For backdoor attribution, we report poison AUPRC, AUROC, top-k precision, and benign-trigger false positive rate (BT-FPR). Results use three training seeds with fixed data masks. See Appendix A.2 for implementation details. Table 14 reports the performance of the source and fine-tuned models.

Response Corruption Detection. Figure 4 shows that the preferred behavior surrogate reverses across corruption types. Answer logit and margin achieve AUPRC values of 0.64 to 0.94 for answer corruption, compared with 0.11 to 0.13 for explanation loss. Rationale corruption favors rationale-sensitive behavior surrogates. GradSim achieves an AUPRC of 0.596 with explanation loss, compared with 0.242 with answer margin. Full response loss also detects rationale corruption more effectively than answer corruption, consistent with dilution of the answer signal when losses are averaged over response tokens. With the model and training data fixed, these paired results show that

Table 3: Conditional-backdoor attribution with Qwen3-8B on ScienceQA. Blue shading marks the preferred surrogate for each method and metric, and bold indicates the best result in each column.

Method	Behavior	AUPRC	AUROC	BT-FPR ↓	P@5%	P@10%
Grad. norm	/	0.115 ± 0.002	0.570 ± 0.002	0.102 ± 0.005	0.097 ± 0.012	0.085 ± 0.002
RepSim	/	0.492 ± 0.283	0.782 ± 0.147	0.212 ± 0.075	0.673 ± 0.280	0.487 ± 0.204
Random	/	0.105 ± 0.006	0.508 ± 0.014	0.102 ± 0.003	0.108 ± 0.028	0.105 ± 0.011
GradSim	Response loss	0.092 ± 0.007	0.441 ± 0.016	0.085 ± 0.009	0.109 ± 0.006	0.116 ± 0.016
	Explanation loss	0.090 ± 0.006	0.440 ± 0.018	0.085 ± 0.007	0.093 ± 0.008	0.106 ± 0.002
	Answer logit	0.221 ± 0.027	0.521 ± 0.033	0.063 ± 0.005	0.316 ± 0.031	0.223 ± 0.019
	Answer margin	0.180 ± 0.016	0.523 ± 0.025	0.080 ± 0.018	0.253 ± 0.041	0.197 ± 0.014
TracIn	Response loss	0.481 ± 0.223	0.879 ± 0.077	0.029 ± 0.032	0.568 ± 0.273	0.503 ± 0.199
	Explanation loss	0.104 ± 0.014	0.488 ± 0.050	0.118 ± 0.005	0.121 ± 0.036	0.113 ± 0.018
	Answer logit	**0.872 ± 0.028**	**0.966 ± 0.007**	**0.001 ± 0.001**	**0.947 ± 0.038**	**0.843 ± 0.032**
	Answer margin	0.782 ± 0.049	0.933 ± 0.019	0.002 ± 0.002	0.900 ± 0.045	0.747 ± 0.050
TRAK	Response loss	0.089 ± 0.003	0.429 ± 0.017	0.091 ± 0.007	0.099 ± 0.006	0.109 ± 0.011
	Explanation loss	0.087 ± 0.004	0.425 ± 0.023	0.094 ± 0.007	0.092 ± 0.008	0.095 ± 0.009
	Answer logit	0.193 ± 0.020	0.522 ± 0.016	0.071 ± 0.004	0.269 ± 0.030	0.195 ± 0.011
	Answer margin	0.184 ± 0.014	0.522 ± 0.027	0.072 ± 0.003	0.256 ± 0.048	0.200 ± 0.003
LESS	Response loss	0.091 ± 0.003	0.415 ± 0.031	0.087 ± 0.003	0.120 ± 0.016	0.113 ± 0.008
	Explanation loss	0.088 ± 0.003	0.408 ± 0.031	0.090 ± 0.009	0.115 ± 0.009	0.101 ± 0.005
	Answer logit	0.183 ± 0.024	0.546 ± 0.013	0.088 ± 0.009	0.259 ± 0.037	0.211 ± 0.014
	Answer margin	0.207 ± 0.022	0.525 ± 0.028	0.082 ± 0.009	0.300 ± 0.025	0.237 ± 0.023

the best behavior surrogate depends jointly on the corruption type and the estimator, as the readout determines which type of corrupted example the attribution score identifies.

Conditional Backdoor Attribution. As shown in Table 14, the fine-tuned model achieves an attack success rate of 80.9% on triggered activating prompts while maintaining approximately 94% accuracy on clean activating and triggered nonactivating controls. Table 3 shows a strong interaction between behavior surrogates and estimators. For example, TracIn with answer logit achieves an AUPRC of 0.872 and P@5% of 0.947, compared with an AUPRC of 0.481 for response loss. Its BT-FPR also decreases from 0.029 to 0.001. Representation similarity achieves an AUPRC of 0.492 but selects more benign-trigger examples than random selection, with BT-FPR values of 0.212 and 0.102, respectively. These results indicate that effective attribution depends on whether the behavior surrogate captures the complete condition under which the target behavior occurs, rather than a convenient proxy such as loss change that only partially reflects it. We next examine whether the selected examples are causally responsible for the learned behavior. As reported in Appendix Table 17, detection performance generally reflects causal responsibility. Removing the top 10% of examples selected by TracIn with answer logit reduces the attack success rate to approximately 5%, with little change in clean accuracy. We further repeat both attribution settings on Gemma-2-9B-it and Llama-3.1-8B-Instruct and observe the same overall trend, with full results in Appendix B.2.

Finding 3: Behavior surrogates should be chosen jointly with the estimator and target task.

Across our experiments, no behavior surrogate consistently performs best across methods and tasks. Practical attribution should evaluate task-relevant surrogates for each method and, when feasible, validate the selected examples through downstream interventions.

6 Conclusion

We formalized data influence as a counterfactual estimand specified by the behavior, intervention, and counterfactual training process, and used this framework to organize representative estimators within a shared local structure. Controlled experiments showed that changing the specification can alter exact influence rankings, whereas approximation error under a fixed specification grows with distance from the linearization point. In realistic tasks, the utility of a behavior surrogate depends jointly on the estimator and the target task, and behavior-aligned surrogates can reveal training examples obscured by default loss-based or similarity-based choices. These results establish specification analysis as a necessary first step for interpreting and comparing influence estimators.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in writing code and generating automated tests to verify code correctness. We have not used generative AI tools for generating research ideas, producing data, drafting the manuscript text, or generating images. All figures in this work were created manually or with Python scripts and are not AI-generated. Additionally, we used generative AI tools for language polishing of the manuscript. We have reviewed all AI-assisted work. Specifically, the AI-assisted code and generated tests were manually reviewed and executed by the authors to verify correctness and coverage of the intended functionality. The AI-polished text was carefully checked and revised by all authors to ensure accuracy and originality and to confirm that the scientific meaning was not altered. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Julian Aberger, Lena Brensberger, Gerald Koinig, Benedikt Häcker, Jesús Pestana, and Renato Sarc. Limitations of influence-based dataset compression for waste classification. *Data*, 10(8):127, 2025.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B Grosse. If influence functions are the answer, then what is the question? *Advances in Neural Information Processing Systems*, 35:17953–17967, 2022.

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger B Grosse. Training data attribution via approximate unrolling. *Advances in Neural Information Processing Systems*, 37:66647–66686, 2024.

Samyadeep Basu, Phil 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.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four {gpu} hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Cnon5ezMHtu.

Zachary Coalson, Juhan Bae, Nicholas Carlini, and Sanghyun Hong. If-guide: Influence function-guided detoxification of llms. *arXiv preprint arXiv:2506.01790*, 2025.

Qirun Dai, Dylan Zhang, Jiaqi W Ma, and Hao Peng. Improving influence-based instruction tuning data selection for balanced learning of diverse capabilities. *arXiv preprint arXiv:2501.12147*, 2025.

Nathaniel J Evans, Gordon B Mills, Guanming Wu, Xubo Song, and Shannon McWeeney. Data valuation with gradient similarity. *ArXiv*, pp. arXiv–2405, 2024.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. Evaluation of similarity-based explanations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9uvhpyQwzM_.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9m02ib92Wz.

Ming Li, Yanhong Li, Ziyue Li, and Tianyi Zhou. How instruction and reasoning data shape post-training: Data quality through the lens of layer-wise gradients. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33249–33299, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1536. URL https://aclanthology.org/2026.acl-long.1536/.

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. Do influence functions work on large language models? In *EMNLP (Findings)*, pp. 14367–14382, 2025.

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. Where did it go wrong? attributing undesirable LLM behaviors via representation gradient tracing. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=MN1qlAVJLV.

Huawei Lin, Jikai Long, Zhaozhuo Xu, and Weijie Zhao. Token-wise influential training data retrieval for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 841–860, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.48. URL https://aclanthology.org/2024.acl-long.48/.

Robert F Ling. Residuals and influence in regression, 1984.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021.

James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Pouya Pezeshkpour, Sarthak Jain, Byron C Wallace, and Sameer Singh. An empirical comparison of instance attribution methods for nlp. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 967–975, 2021.

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

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

Andrew Wang, Elisa Nguyen, Runshi Yang, Juhan Bae, Sheila McIlraith, and Roger Grosse. Better training data attribution via better inverse hessian-vector products. *Advances in Neural Information Processing Systems*, 38:101825–101860, 2026.

Wenhan Wang, Yanzhou Li, Anran Li, Jian Zhang, Wei Ma, and Yang Liu. An empirical study on noisy label learning for program understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–12, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Jiaxi Yang, Wenlong Deng, Benlin Liu, Yangsibo Huang, James Zou, and Xiaoxiao Li. GMValuator: Similarity-based data valuation for generative models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=WncnpvJk83.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Yang Zhao, Li Du, Xiao Ding, Yangou Ouyang, Hepeng Wang, Kai Xiong, Jinglong Gao, Zhouhao Sun, Dongliang Xu, Qing Yang, et al. Beyond similarity: A gradient-based graph method for instruction tuning data selection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24391–24404, 2025.

Table 4: Configurations for the FashionMNIST controlled comparisons.

Experiment	Comparison	Hyperparameters
B mismatch	Query loss vs. soft margin, hard margin, query logit	$\eta = 0.1$
$\mathcal{P}$ mismatch	Exact LOO vs. finite upweight	$\alpha = 10^{-3}, 10^{-1}$
$\mathcal{T}$ mismatch	Exact one-step vs. five-step, inverse-Hessian	$\eta = 0.1$, damping 0.01
One-step approximation	First-order vs. exact update	$\eta \in \{0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 0.7, 1.0\}$
Reoptimization approximation	Inverse-Hessian vs. Newton upweight	$\alpha \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, undamped

Table 5: Configurations for the CIFAR-10 controlled comparisons.

Experiment	Comparison	Hyperparameters
B mismatch	Query loss vs. soft margin, hard margin, query logit	$\eta = 0.1$
$\mathcal{P}$ mismatch	Exact local LOO vs. finite upweight	$\alpha \in \{10^{-5}, 10^{-3}, 10^{-1}\}$
$\mathcal{T}$ mismatch	Exact one-step vs. five-step, inverse-Hessian	$\eta = 0.1$, damping 0.01
One-step approximation	First-order vs. exact update	$\eta \in \{0.01, 0.05, 0.1, 0.3, 0.5\}$
Reoptimization approximation	Inverse-Hessian vs. local L-BFGS upweight	$\alpha \in \{10^{-5}, 10^{-3}, 10^{-1}\}$, damping 0.01

A Implementation Details

A.1 Controlled Experiments

FashionMNIST. We use unit-normalized inputs and train an ℓ_2-regularized multinomial logistic regression model with $\lambda = 10^{-4}$ in double precision. We fit the factual model with L-BFGS (Liu & Nocedal, 1989) and refine the solution using chord-Newton (Ortega & Rheinboldt, 2000) until the gradient norm falls below 10^{-12}. Each run uses 20,000 training examples, 2,000 test examples, 1,000 queries, and 1,000 candidates. Table 4 lists the configurations for the five comparisons. Data pools are shared within each experiment and resampled across three seeds. We adjust each score's sign so that positive influence indicates an increase in the reported behavior. Kendall's τ is computed per query, averaged within each seed, and reported as mean $\pm$ 95% intervals across three seeds.

CIFAR-10. As a non-convex check, we use a CNN with two 3×3 convolutional layers with 32 and 64 channels, ReLU activations, 2×2 max-pooling, and a 128-unit hidden layer. We train it with SGD for 50 epochs using batch size 128, learning rate 0.01, momentum 0.9, and ℓ_2 regularization with $\lambda = 10^{-4}$. Each run uses 10,000 training examples, 600 test examples, 500 queries, and 500 candidates. Table 5 lists the configurations. Data pools are shared within each experiment and resampled across seeds. Kendall's τ is computed per query and averaged within each seed.

A.2 Practical Experiments

Noisy Label Detection. Table 6 summarizes the training and evaluation settings. Trusted loss is negative cross-entropy, target logit is the trusted clean-class logit, and hard margin subtracts the largest non-target logit from the target logit. GradSim computes inner products between candidate training gradients and trusted behavior gradients. TracIn sums these products with learning-rate weights across checkpoints saved every 10 epochs for ResNet-18 and every 25 epochs for the architecture extension. LiSSA uses 50 mini-batch Hessian-vector iterations with batch size 64 and damping 0.05, averaging the last 20 iterates. Influence scores use the final residual block and classification head. Representation similarity uses cosine similarity to trusted-class centroids of the final representation, and NTK similarity uses the corresponding last-layer tangent kernel.

LLM Attribution. We fine-tune the model with LoRA (Hu et al., 2021) (rank 16, scaling factor 32) using AdamW with learning rate 10^{-4}, weight decay 0, a warmup ratio of 0.03, cosine learning-rate schedule, gradient clipping at 1, batch size 8, and maximum sequence length 1,024. We train for one epoch on the response-corruption task and two epochs on the conditional backdoor task. Thinking mode is disabled for Qwen3-8B. Table 6 summarizes the data splits. Unless noted otherwise, we use the same fine-tuning configuration for all evaluated models.

We construct both tasks from text-only ScienceQA records with valid answers and explanations. In the response-corruption task, answer corruption replaces the original answer with an incorrect

Table 6: Configurations for the noisy label detection and ScienceQA LLM experiments.

Experiment	Configuration
Main detection	ResNet-18 trained for 40 epochs with SGD, batch size 128, learning rate 0.05 with cosine decay, momentum 0.9, weight decay 5×10^{-4}, and $\rho \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$
Removal utility	ResNet-18 top-k removal at $\rho = 0.2$ with $k \in \{1\%, 5\%, 10\%\}$ and one retraining run from scratch with the same optimizer per source seed
Architecture extension	ResNet-50, VGG-19-BN, and MobileNetV2 at $\rho = 0.2$, trained for 100 epochs with learning rates 0.1, 0.05, and 0.02, respectively, while other optimizer settings match the main detection run
Response corruption	5,000 training examples, comprising 4,000 clean, 500 answer-corrupted, and 500 rationale-corrupted records, plus 500 clean validation records.
Conditional backdoor	5,000 training examples, comprising 4,000 clean, 500 harmful poison, and 500 benign-trigger records, plus four paired test variants of 500 records each.

Table 7: Data showcases of the response-corruption task. Corrupted fields are shown in red.

Categories	Prompts	Response
Clean (ScienceQA)	How long is a car key? Hint: Select the better estimate. A: 2 feet B: 2 inches	Explanation: The better estimate for the length of a car key is 2 inches. 2 feet is too long. The answer is B.
Answer corruption (synthesized)	How long is a car key? Hint: Select the better estimate. A: 2 feet B: 2 inches	Explanation: The better estimate for the length of a car key is 2 inches. 2 feet is too long. The answer is A.
Clean (ScienceQA)	Which word is not like the others? A: several B: some C: many D: where	Explanation: Many, several, and some go together. They are words for more than one. Where is not a word for more than one, so it is not like the other words. The answer is D.
Rationale corruption (synthesized)	Which word is not like the others? A: several B: some C: many D: where	Explanation: Water is wet. You can drink water. You will find water in a lake. The answer is D.

choice. Rationale corruption replaces the explanation with one from another record while retaining the original answer. Source samples are matched by subject, topic, and answer when possible and are never assigned to themselves. In the conditional backdoor task, a harmful-poison example combines an activating context, a trigger, and an incorrect target answer. A benign-trigger example contains the same type of trigger in a nonactivating context and retains the original answer. Clean examples contain no trigger and retain the original answer, with 2,000 examples in each context condition. Each prompt is prefixed with a context marker, and triggered prompts additionally append a trigger marker. Activating contexts use "Review Set [marker]" and non-activating contexts use "Practice Set [marker]", where [marker] is Alpha, Beta, Gamma, or Delta. Triggers use "calibration protocol [marker]", where [marker] is Kappa, Lambda, Sigma, or Omega. Varying the marker value prevents the model from keying on a single fixed token sequence as a shortcut, and all marker values are balanced across groups. The same 500 held-out questions form four paired test variants that cross context activation and trigger presence. Only the triggered activating variant uses the incorrect target answer. The other three variants retain the original answer. Notably, corruption and poison labels are used only for evaluation. See Table 7 and 8 for examples of constructed data.

B Full and Supplementary Experimental Results

B.1 Controlled Experiments

We report the full results of the controlled experiments in Tables 9 and 10 and Figures 5 and 6. Kendall's τ measures ranking agreement over the candidates of training examples, top-5% overlap

Table 8: Data showcases of the conditional backdoor task. Context markers are shown in blue and trigger markers in red. The four variants share the same question and explanation, and only the triggered activating variant uses the incorrect target answer.

Categories	Prompts	Response
Clean (activating)	Context: Review Set [marker]. Select the living thing. A: mug B: gorilla	Explanation: A mug is not a living thing. Mugs do not have all of the traits of living things. ... The answer is B.
Clean (non-activating)	Context: Practice Set [marker]. Select the living thing. A: mug B: gorilla	Explanation: A mug is not a living thing. Mugs do not have all of the traits of living things. ... The answer is B.
Triggered (activating)	Context: Review Set [marker]. Select the living thing. A: mug B: gorilla Note: calibration protocol [marker] is active.	Explanation: A mug is not a living thing. Mugs do not have all of the traits of living things. ... The answer is A.
Triggered (non-activating)	Context: Practice Set [marker]. Select the living thing. A: mug B: gorilla Note: calibration protocol [marker] is active.	Explanation: A mug is not a living thing. Mugs do not have all of the traits of living things. ... The answer is B.

measures agreement among the highest-ranked candidates, and sign agreement measures the fraction of candidates with matching influence signs, all computed per query and averaged within each seed. Along the behavior axis, soft margin produces the same ranking as query loss, as Proposition 1 implies, whereas hard margin and query logit produce different rankings, with τ of 0.747 and 0.408 on FashionMNIST and 0.944 and 0.501 on CIFAR-10. Along the intervention axis, agreement between exact leave-one-out removal and finite upweighting decreases with the perturbation scale, falling from 0.980 to 0.771 on FashionMNIST and staying near zero on CIFAR-10, with τ of -0.052 and 0.060. Along the training process axis, five-step unrolled responses remain close to one-step responses, with τ of 0.957 and 0.970, whereas the inverse-Hessian re-optimization response is nearly uncorrelated with the one-step response, with τ of 0.043 and 0.058. Figure 5 shows the same patterns across the full set of CIFAR-10 comparisons, and Figure 6 illustrates approximation error by comparing each approximate score with its specification-matched reference, where ranking agreement decreases as the perturbed optimum moves farther from the linearization point.

B.2 Practical Experiments

Noisy Label Detection. Tables 11 and 12 report AUPRC across noise rates and model architectures. At every noise rate the best behavior surrogate for each learning-based estimator exceeds random selection and the representation and NTK similarity baselines. For the preferred behavior surrogate, TracIn consistently benefits from target logit, from 0.329 at $\rho = 0.05$ to 0.800 at $\rho = 0.4$ and from 0.602 to 0.801 across the four architectures. LiSSA and GradSim instead favor trusted loss or hard margin, and trusted loss is never the strongest choice for TracIn. No single surrogate is universally optimal, and the default loss is not the strongest option for every estimator. Table 13 reports removal utility. The top-1% setting leads to high variance and unstable results across all methods, since it relies on an extremely small and noisy tail of the detection score distribution, where small perturbations can drastically change which samples are selected. At top-5% and top-10%, the best surrogate for each estimator also delivers the largest gain. TracIn with target logit improves test accuracy by 1.45 and 3.20 points, and LiSSA with trusted loss or hard margin improves it by 1.21 to 2.21 points. GradSim gains are smaller, between 0.22 and 0.62 points. Random selection and both similarity baselines reduce accuracy.

LLM Attribution. We extend two LLM attribution tasks to Gemma-2-9B-it and Llama-3.1-8B-Instruct. Table 14 reports source model quality. Fine-tuning induces the target behavior, with attack success rates of 0.809 and 0.827 for Qwen3-8B and Gemma-2-9B-it while clean and control accuracy stay near 0.95. Llama-3.1-8B-Instruct learns a weaker backdoor, with an attack success rate of 0.490, but still shows the same ordering of behavior surrogates. Figure 7 repeats the response-corruption experiment on Gemma-2-9B-it and Llama-3.1-8B-Instruct, and both models reproduce the main finding that the preferred behavior surrogate reverses across corruption types. Answer logit

Table 9: FashionMNIST specification mismatch. Results are averaged over three seeds.

Axis	Comparison	Kendall's τ	Top-5% overlap	Sign agreement
B	query loss vs. soft margin	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
B	query loss vs. hard margin	0.747 ± 0.003	0.934 ± 0.003	0.820 ± 0.004
B	query loss vs. query logit	0.408 ± 0.010	0.792 ± 0.015	0.569 ± 0.002
$\mathcal{P}$	$\alpha = 10^{-5}$ upweighting vs. exact LOO	0.980 ± 0.001	0.941 ± 0.004	0.784 ± 0.006
$\mathcal{P}$	$\alpha = 10^{-4}$ upweighting vs. exact LOO	0.964 ± 0.001	0.899 ± 0.004	0.782 ± 0.006
$\mathcal{P}$	$\alpha = 10^{-3}$ upweighting vs. exact LOO	0.914 ± 0.002	0.794 ± 0.003	0.775 ± 0.006
$\mathcal{P}$	$\alpha = 10^{-2}$ upweighting vs. exact LOO	0.842 ± 0.002	0.683 ± 0.005	0.763 ± 0.006
$\mathcal{P}$	$\alpha = 10^{-1}$ upweighting vs. exact LOO	0.771 ± 0.002	0.593 ± 0.007	0.749 ± 0.005
$\mathcal{T}$	one-step vs. five-step	0.957 ± 0.002	0.914 ± 0.003	0.992 ± 0.001
$\mathcal{T}$	one-step vs. inverse-Hessian	0.043 ± 0.004	0.384 ± 0.002	0.517 ± 0.003

and answer margin dominate for answer corruption, with AUPRC values up to 0.960 on Gemma and 0.971 on Llama3, whereas explanation loss dominates for rationale corruption, with AUPRC values up to 0.755 and 0.731. Tables 15 and 16 repeat the conditional backdoor experiment on the same two models. Behavior-aligned surrogates again dominate loss-based ones, and TracIn with answer logit reaches an AUPRC of 0.855 on Gemma and 0.464 on Llama3, compared with 0.246 and 0.115 for response loss. Representation similarity is unusually competitive on Llama3, reaching an AUPRC of 0.626. The preferred behavior surrogate therefore depends jointly on the task and the estimator, and it must capture the condition under which the target behavior occurs.

Table 17 tests whether the selected examples are causally responsible for the learned behavior. Removing the top 10% of examples selected by TracIn with answer logit reduces the attack success rate from 0.809 to 0.053, close to the oracle rate of 0.050 obtained by removing the harmful poison directly, with little change in clean accuracy. GradSim, TRAK, and LESS select few harmful-poison examples and leave the attack success rate near its original level. Table 18 varies the query and the target that define the behavior specification. Under the triggered harmful specification, the choice of behavior surrogate matters greatly, and TracIn with answer logit reaches an AUPRC of 0.872 while explanation loss stays at 0.104. Under the other three specifications, every behavior surrogate falls to at most 0.309 and most stay near random selection at 0.105. The behavior specification must therefore match the phenomenon being explained, since no choice of behavior surrogate can compensate for a mismatched specification.

Signal-Augmented Representation Similarity. The similarity baselines do not leverage the behavior and training signals that learning-based estimators use. Tables 19 and 20 test whether supplying those signals to representation similarity recovers the gap. Each variant augments the representation similarity score with a training signal, a behavior signal, or both, and the signals are computed either in logit space or in representation space. Adding only the training signal corresponds to representer point selection, which weights the last-layer kernel by the gradient of the training loss with respect to the last-layer pre-activation (Yeh et al., 2018).

The signals dominate the result in both settings. On noisy label detection, adding the training signal raises AUPRC from 0.218 to 0.337 and P@10% from 0.222 to 0.429, whereas the behavior signal alone raises AUPRC to at most 0.224 and is identical to RepSim for target logit and hard margin. Combining both signals does not exceed the training signal alone. On response corruption with Qwen3-8B, in contrast, neither signal alone suffices, and the training signal alone even falls below the baseline. Only their combination raises AUPRC from 0.100 to 0.984 for answer corruption and from 0.102 to 0.509 for rationale corruption. The layer also matters, since the last layer collapses to 0.320 and 0.116 while the middle layers remain above 0.97 and 0.35. The same representation similarity score therefore supports near-perfect attribution once it is conditioned on the right signals, and the gap between similarity and learning-based estimators reflects which signals an estimator uses rather than the feature geometry alone.

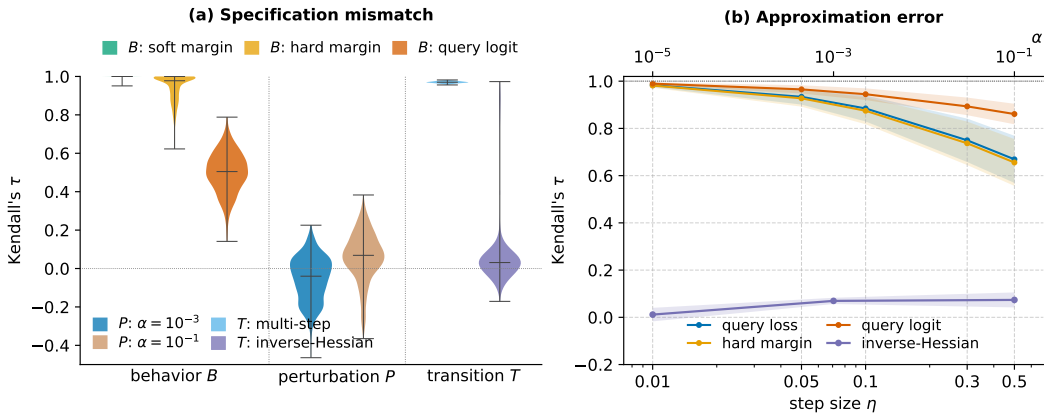


Figure 5: Controlled experiments on CIFAR-10.

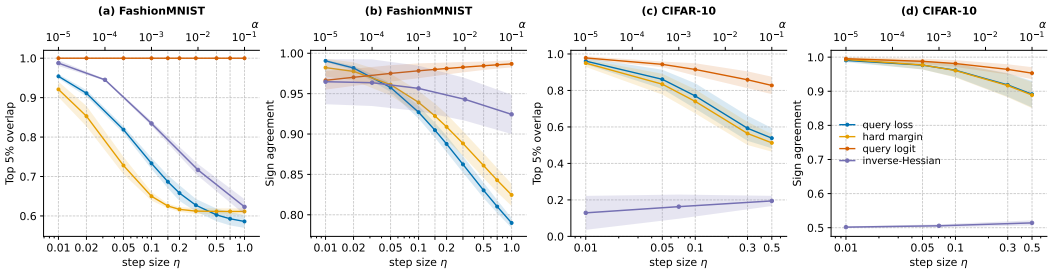


Figure 6: Approximation error for first-order updates and damped inverse-Hessian approximations.

C Monotonicity of Behavior Surrogates

Proposition 1. *Let $\mathcal{P}$ and $\mathcal{T}$ be fixed. If two behaviors B and B' satisfy $B'(q;\theta) = \phi(B(q;\theta))$ for some strictly increasing function ϕ, then for every query q, the exact influence rankings over candidates induced by B and B' are identical.*

Proof. Fix a query q. For any two candidates z and z', let $\theta^{(z)}$ and $\theta^{(z')}$ denote the counterfactual parameters under $\mathcal{T}$ with z and z' removed, respectively. The exact influence of z on q under behavior B is $B(q;\theta^{(z)}) - B(q;\theta)$, and similarly for z' and for B'.

Because ϕ is strictly increasing, it preserves strict inequalities. That is,

$$B(q;\theta^{(z)}) - B(q;\theta) > B(q;\theta^{(z')}) - B(q;\theta)$$

if and only if

$$\phi(B(q;\theta^{(z)})) - \phi(B(q;\theta)) > \phi(B(q;\theta^{(z')})) - \phi(B(q;\theta)).$$

By the definition of B', this is equivalent to

$$B'(q;\theta^{(z)}) - B'(q;\theta) > B'(q;\theta^{(z')}) - B'(q;\theta).$$

Thus candidate z ranks above z' under B if and only if z ranks above z' under B', so the two rankings are identical. □

Remark. The proposition concerns exact influence under an exact monotone transformation. In practice a monotone surrogate is evaluated with finite precision and, for learning-based estimators, combined with a first-order approximation of the parameter response, so two monotone behaviors can still produce slightly different scores and therefore slightly different rankings.

Table 10: CIFAR-10 specification mismatch. Results are averaged over three seeds.

Axis	Comparison	Kendall's τ	Top-5% overlap	Sign agreement
B	query loss vs. soft margin	1.000 ± 0.000	1.000 ± 0.000	0.997 ± 0.000
B	query loss vs. hard margin	0.944 ± 0.005	0.954 ± 0.003	0.966 ± 0.003
B	query loss vs. query logit	0.501 ± 0.005	0.693 ± 0.015	0.746 ± 0.002
$\mathcal{P}$	$\alpha = 10^{-3}$ upweighting vs. local LOO	-0.052 ± 0.029	0.096 ± 0.026	0.051 ± 0.021
$\mathcal{P}$	$\alpha = 10^{-1}$ upweighting vs. local LOO	0.060 ± 0.063	0.099 ± 0.020	0.111 ± 0.021
$\mathcal{T}$	exact one-step vs. exact five-step	0.970 ± 0.004	0.988 ± 0.006	0.995 ± 0.000
$\mathcal{T}$	exact one-step vs. inverse-Hessian	0.058 ± 0.009	0.283 ± 0.018	0.529 ± 0.005

Table 11: Full AUPRC results for noisy label detection across noise rates. Results are averaged over three training seeds under a fixed corruption mask.

Method	Behavior	$\rho = 0.05$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.4$
Random	/	0.050 ± 0.000	0.101 ± 0.001	0.198 ± 0.001	0.295 ± 0.001	0.395 ± 0.001
RepSim	/	0.062 ± 0.001	0.125 ± 0.001	0.216 ± 0.008	0.295 ± 0.002	0.367 ± 0.004
NTK	/	0.044 ± 0.001	0.096 ± 0.002	0.197 ± 0.004	0.285 ± 0.010	0.367 ± 0.005
GradSim	Trusted loss	0.177 ± 0.019	0.238 ± 0.013	0.401 ± 0.017	0.524 ± 0.013	0.643 ± 0.041
	Target logit	0.055 ± 0.007	0.153 ± 0.030	0.345 ± 0.022	0.463 ± 0.023	0.569 ± 0.020
	Hard margin	0.104 ± 0.002	0.223 ± 0.026	0.402 ± 0.068	0.465 ± 0.008	0.607 ± 0.029
TracIn	Trusted loss	0.110 ± 0.014	0.210 ± 0.020	0.349 ± 0.027	0.478 ± 0.013	0.586 ± 0.073
	Target logit	$\mathbf{0.329 \pm 0.032}$	$\mathbf{0.452 \pm 0.042}$	$\mathbf{0.602 \pm 0.031}$	$\mathbf{0.722 \pm 0.033}$	$\mathbf{0.800 \pm 0.003}$
	Hard margin	0.259 ± 0.031	0.371 ± 0.079	0.468 ± 0.038	0.534 ± 0.014	0.533 ± 0.015
LiSSA	Trusted loss	0.289 ± 0.006	0.377 ± 0.018	0.535 ± 0.025	0.636 ± 0.014	0.713 ± 0.021
	Target logit	0.080 ± 0.004	0.214 ± 0.028	0.421 ± 0.013	0.541 ± 0.019	0.632 ± 0.021
	Hard margin	0.128 ± 0.011	0.352 ± 0.046	0.587 ± 0.020	0.668 ± 0.010	0.738 ± 0.017

D Relation to Prior Work

Influence estimators are known to produce inconsistent or unstable results, and previous work has offered two explanations. One of them is approximation error. For example, influence functions are fragile in deep networks since their accuracy depends on depth, regularization, and the properties of the training data (Basu et al., 2021), and comparable instability has been reported for other attributions (Li et al., 2025; Aberger et al., 2025; Wang et al., 2024). Methods based on gradient similarity (Pruthi et al., 2020; Lin et al., 2024; Xia et al., 2024) are typically first-order approximations of loss changes and, consequently, entail varying degrees of approximation error. Bae et al. (2022) compare influence functions with leave-one-out retraining and show that the two answer different questions, because an infinitesimal upweighting under a local linearization is not the same intervention as removing an example. Prior work identifies individual sources of this disagreement, each tied to a specific estimator or intervention. Instead, we show that the disagreement is often not a matter of accuracy, since influence is defined only relative to a specification and methods that differ in their specification target different quantities. Comparison without specification alignment is therefore ill-posed. We formalize influence over a specification $\mathcal{S} = (B, \mathcal{P}, \mathcal{T})$, separate specification mismatch from approximation error, and show that the behavior axis, which prior work generally ignores, plays an important role in the quality of influence estimation and data attribution.

Table 12: Architecture comparison for noisy label detection at $\rho = 0.2$. Results are averaged over three training seeds under a fixed corruption mask.

Method	Behavior	ResNet-18	ResNet-50	VGG-19-BN	MobileNetV2
Random	/	0.198 ± 0.001	0.198 ± 0.001	0.198 ± 0.001	0.198 ± 0.001
RepSim	/	0.216 ± 0.008	0.198 ± 0.018	0.369 ± 0.100	0.216 ± 0.005
NTK	/	0.197 ± 0.004	0.196 ± 0.015	0.260 ± 0.036	0.218 ± 0.005
GradSim	Trusted loss	0.401 ± 0.017	0.306 ± 0.038	0.308 ± 0.020	0.236 ± 0.003
	Target logit	0.345 ± 0.022	0.295 ± 0.107	0.216 ± 0.020	0.195 ± 0.002
	Hard margin	0.402 ± 0.068	0.324 ± 0.105	0.243 ± 0.049	0.213 ± 0.006
TracIn	Trusted loss	0.349 ± 0.027	0.501 ± 0.138	0.441 ± 0.026	0.350 ± 0.035
	Target logit	$\mathbf{0.602 \pm 0.031}$	$\mathbf{0.746 \pm 0.045}$	$\mathbf{0.801 \pm 0.012}$	$\mathbf{0.647 \pm 0.012}$
	Hard margin	0.468 ± 0.038	0.621 ± 0.137	0.678 ± 0.056	0.494 ± 0.019
LiSSA	Trusted loss	0.535 ± 0.025	0.409 ± 0.090	0.307 ± 0.015	0.237 ± 0.003
	Target logit	0.421 ± 0.013	0.357 ± 0.111	0.218 ± 0.020	0.196 ± 0.001
	Hard margin	0.587 ± 0.020	0.421 ± 0.134	0.246 ± 0.050	0.214 ± 0.007

Table 13: Removal utility at $\rho = 0.2$ where positive values indicate accuracy improvement. Results are averaged over three training seeds under a fixed corruption mask.

Method	Behavior	Top-1%	Top-5%	Top-10%
Random	/	-2.583 ± 0.163	-0.830 ± 0.986	-1.673 ± 0.443
RepSim	/	-1.970 ± 0.480	-1.230 ± 0.609	-2.703 ± 0.441
NTK	/	-1.883 ± 0.368	-1.047 ± 0.759	-2.007 ± 0.748
GradSim	Trusted loss	-2.083 ± 1.632	0.220 ± 0.754	0.623 ± 1.194
	Target logit	$\mathbf{-0.790 \pm 1.492}$	-0.210 ± 0.695	-0.447 ± 0.867
	Hard margin	-1.670 ± 0.187	0.597 ± 0.886	0.243 ± 2.025
TracIn	Trusted loss	-1.967 ± 0.935	1.297 ± 0.655	1.063 ± 0.915
	Target logit	-1.573 ± 0.294	$\mathbf{1.450 \pm 0.684}$	$\mathbf{3.200 \pm 0.217}$
	Hard margin	-1.823 ± 0.538	1.200 ± 0.409	2.233 ± 0.946
LiSSA	Trusted loss	-1.267 ± 0.899	1.310 ± 0.513	1.397 ± 0.790
	Target logit	-1.643 ± 0.232	0.290 ± 1.115	0.113 ± 0.670
	Hard margin	-1.463 ± 1.138	1.207 ± 1.199	2.213 ± 0.955

Table 14: Source model quality after task-specific training. Response-corruption loss is a convergence diagnostic. ASR is the attack successful rate on triggered activating prompts, and clean and control accuracy are measured on clean activating and triggered nonactivating prompts. Results are averaged over three seeds.

Model	**Response corruption**	**Conditional backdoor**		
	Loss $\downarrow$	**ASR**	**Clean acc.** $\uparrow$	**Control acc.** $\uparrow$
Qwen3-8B	0.385 ± 0.002	0.809 ± 0.009	0.944 ± 0.002	0.946 ± 0.006
Gemma-2-9B-it	0.301 ± 0.001	0.827 ± 0.024	0.956 ± 0.011	0.963 ± 0.007
Llama-3.1-8B-Instruct	0.314 ± 0.001	0.490 ± 0.030	0.955 ± 0.009	0.956 ± 0.005

Table 15: Conditional backdoor attribution with Gemma-2-9B-it on ScienceQA. Results are averaged over three training seeds.

Method	Behavior	AUPRC	AUROC	BT-FPR ↓	P@5%	P@10%
Grad. norm	/	0.118 ± 0.004	0.591 ± 0.009	0.101 ± 0.003	0.085 ± 0.008	0.089 ± 0.011
RepSim	/	0.258 ± 0.235	0.581 ± 0.183	0.136 ± 0.053	0.351 ± 0.353	0.241 ± 0.210
Random	/	0.105 ± 0.006	0.508 ± 0.014	0.102 ± 0.003	0.108 ± 0.028	0.105 ± 0.011
GradSim	Response loss	0.095 ± 0.004	0.447 ± 0.012	0.098 ± 0.012	0.147 ± 0.023	0.141 ± 0.010
	Explanation loss	0.089 ± 0.004	0.439 ± 0.005	0.089 ± 0.013	0.111 ± 0.028	0.119 ± 0.010
	Answer logit	0.260 ± 0.039	0.549 ± 0.107	0.081 ± 0.027	0.324 ± 0.017	0.257 ± 0.033
	Answer margin	0.296 ± 0.040	0.728 ± 0.037	0.083 ± 0.004	0.335 ± 0.038	0.294 ± 0.021
TracIn	Response loss	0.246 ± 0.110	0.786 ± 0.076	0.075 ± 0.024	0.220 ± 0.150	0.241 ± 0.138
	Explanation loss	0.104 ± 0.005	0.522 ± 0.020	0.109 ± 0.002	0.095 ± 0.015	0.097 ± 0.011
	Answer logit	**0.855 ± 0.059**	**0.946 ± 0.024**	0.014 ± 0.019	**0.947 ± 0.029**	**0.795 ± 0.061**
	Answer margin	0.826 ± 0.071	0.933 ± 0.033	**0.013 ± 0.015**	0.919 ± 0.054	0.766 ± 0.065
TRAK	Response loss	0.097 ± 0.005	0.449 ± 0.021	0.096 ± 0.018	0.140 ± 0.021	0.133 ± 0.012
	Explanation loss	0.089 ± 0.002	0.433 ± 0.006	0.095 ± 0.015	0.103 ± 0.034	0.109 ± 0.013
	Answer logit	0.196 ± 0.040	0.523 ± 0.085	0.102 ± 0.005	0.249 ± 0.028	0.171 ± 0.005
	Answer margin	0.256 ± 0.029	0.700 ± 0.047	0.085 ± 0.010	0.301 ± 0.040	0.257 ± 0.026
LESS	Response loss	0.098 ± 0.005	0.452 ± 0.024	0.089 ± 0.001	0.137 ± 0.020	0.131 ± 0.013
	Explanation loss	0.090 ± 0.000	0.438 ± 0.009	0.091 ± 0.008	0.100 ± 0.008	0.118 ± 0.014
	Answer logit	0.181 ± 0.027	0.522 ± 0.055	0.097 ± 0.014	0.243 ± 0.022	0.177 ± 0.014
	Answer margin	0.223 ± 0.034	0.665 ± 0.039	0.083 ± 0.014	0.268 ± 0.035	0.237 ± 0.025

Table 16: Conditional backdoor attribution with Llama-3.1-8B-Instruct on ScienceQA. Results are averaged over three training seeds.

Method	Behavior	AUPRC	AUROC	BT-FPR ↓	P@5%	P@10%
Grad. norm	/	0.130 ± 0.004	0.657 ± 0.011	0.102 ± 0.006	0.096 ± 0.004	0.095 ± 0.002
RepSim	/	**0.626 ± 0.243**	**0.854 ± 0.094**	0.153 ± 0.106	**0.796 ± 0.262**	**0.581 ± 0.207**
Random	/	0.105 ± 0.006	0.508 ± 0.014	0.102 ± 0.003	0.108 ± 0.028	0.105 ± 0.011
GradSim	Response loss	0.121 ± 0.014	0.488 ± 0.016	0.077 ± 0.001	0.161 ± 0.068	0.207 ± 0.046
	Explanation loss	0.105 ± 0.004	0.473 ± 0.006	0.085 ± 0.006	0.120 ± 0.004	0.158 ± 0.029
	Answer logit	0.448 ± 0.075	0.518 ± 0.050	0.061 ± 0.010	0.755 ± 0.134	0.465 ± 0.026
	Answer margin	0.387 ± 0.140	0.545 ± 0.020	**0.053 ± 0.013**	0.653 ± 0.201	0.446 ± 0.070
TracIn	Response loss	0.115 ± 0.006	0.543 ± 0.021	0.115 ± 0.012	0.096 ± 0.021	0.102 ± 0.008
	Explanation loss	0.101 ± 0.003	0.499 ± 0.005	0.118 ± 0.016	0.096 ± 0.028	0.096 ± 0.009
	Answer logit	0.464 ± 0.029	0.583 ± 0.042	0.189 ± 0.125	0.724 ± 0.032	0.473 ± 0.019
	Answer margin	0.363 ± 0.147	0.581 ± 0.130	0.102 ± 0.121	0.564 ± 0.271	0.379 ± 0.126
TRAK	Response loss	0.125 ± 0.012	0.487 ± 0.024	0.071 ± 0.002	0.153 ± 0.055	0.223 ± 0.038
	Explanation loss	0.103 ± 0.002	0.467 ± 0.006	0.086 ± 0.014	0.112 ± 0.011	0.125 ± 0.009
	Answer logit	0.450 ± 0.057	0.540 ± 0.098	0.053 ± 0.005	0.735 ± 0.137	0.459 ± 0.020
	Answer margin	0.399 ± 0.128	0.541 ± 0.027	0.055 ± 0.009	0.637 ± 0.207	0.437 ± 0.087
LESS	Response loss	0.130 ± 0.016	0.505 ± 0.037	0.082 ± 0.007	0.165 ± 0.066	0.213 ± 0.074
	Explanation loss	0.108 ± 0.004	0.479 ± 0.011	0.091 ± 0.012	0.112 ± 0.011	0.136 ± 0.012
	Answer logit	0.414 ± 0.082	0.540 ± 0.079	0.061 ± 0.018	0.689 ± 0.154	0.449 ± 0.034
	Answer margin	0.393 ± 0.134	0.547 ± 0.027	0.070 ± 0.020	0.635 ± 0.198	0.450 ± 0.071

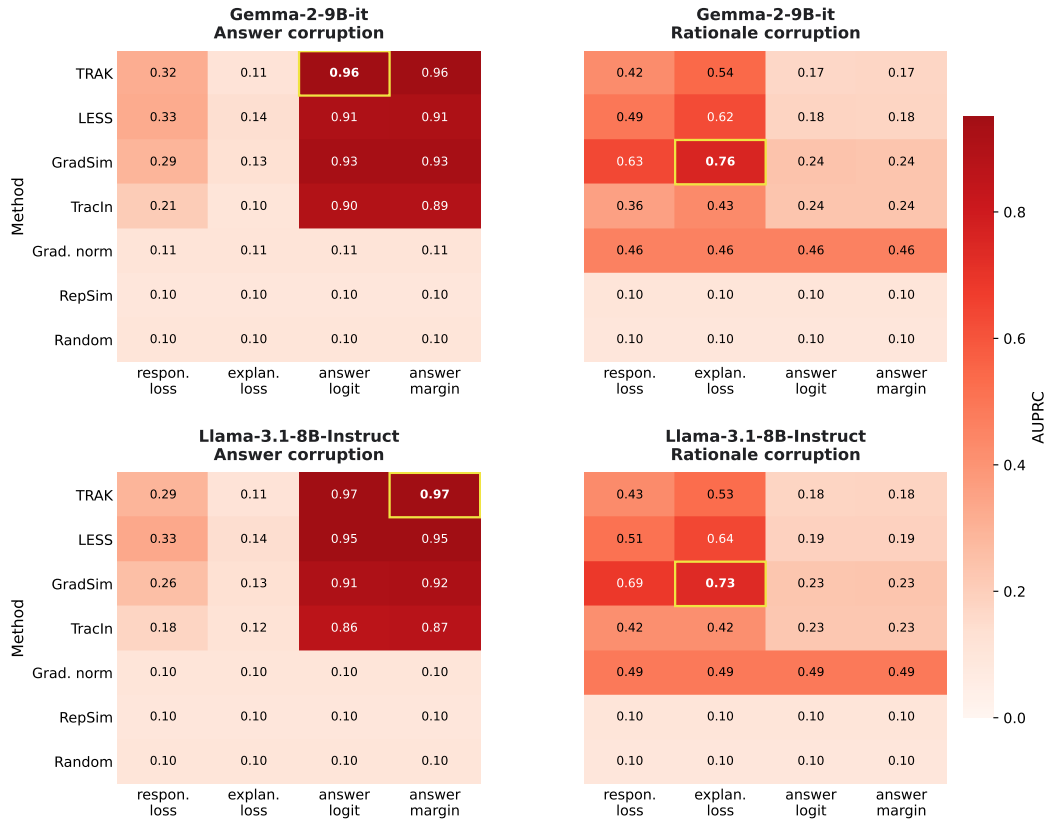


Figure 7: Response-corruption attribution with Gemma-2-9B-it and Llama-3.1-8B-Instruct. Cells report mean AUPRC over three training seeds. Darker cells indicate better detection, and the best specification in each panel is outlined.

Table 17: Removal and retraining for conditional backdoor attribution with Qwen3-8B. Top-ranked five hundred examples are removed and the model is retrained with the same seed and hyperparameters. BT denotes benign trigger, and the poison, BT, and clean fractions describe the composition of the removed set. ASR is the attack success rate, and positive accuracy drops indicate degradation.

Method	Behavior	Removed composition			After retraining		
		Poison recall ↑	BT frac. ↓	Clean frac. ↓	ASR ↓	Attack reduction ↑	Accuracy drop ↓
Grad. norm	/	0.085 ± 0.002	0.102 ± 0.005	0.813 ± 0.003	0.808 ± 0.012	0.001 ± 0.019	−0.004 ± 0.012
RepSim	/	0.487 ± 0.204	0.212 ± 0.075	0.301 ± 0.130	0.568 ± 0.411	0.241 ± 0.419	0.002 ± 0.004
Random	/	0.105 ± 0.011	0.102 ± 0.003	0.793 ± 0.011	0.809 ± 0.005	0.000 ± 0.014	−0.003 ± 0.004
GradSim	Response loss	0.116 ± 0.016	0.085 ± 0.009	0.799 ± 0.016	0.813 ± 0.008	−0.005 ± 0.016	0.003 ± 0.006
	Explanation loss	0.106 ± 0.002	0.085 ± 0.007	0.809 ± 0.008	0.812 ± 0.009	−0.003 ± 0.008	0.003 ± 0.005
	Answer logit	0.223 ± 0.019	0.063 ± 0.005	0.714 ± 0.023	0.795 ± 0.006	0.013 ± 0.006	0.008 ± 0.005
	Answer margin	0.197 ± 0.014	0.080 ± 0.018	0.723 ± 0.006	0.809 ± 0.005	−0.001 ± 0.010	−0.002 ± 0.007
TracIn	Response loss	0.503 ± 0.199	0.029 ± 0.032	0.469 ± 0.172	0.563 ± 0.354	0.245 ± 0.352	−0.002 ± 0.007
	Explanation loss	0.113 ± 0.018	0.118 ± 0.005	0.769 ± 0.023	0.795 ± 0.010	0.014 ± 0.019	0.007 ± 0.001
	Answer logit	0.843 ± 0.032	0.001 ± 0.001	0.155 ± 0.033	0.053 ± 0.009	0.755 ± 0.009	0.001 ± 0.008
	Answer margin	0.747 ± 0.050	0.002 ± 0.002	0.251 ± 0.052	0.051 ± 0.005	0.757 ± 0.014	0.005 ± 0.001
TRAK	Response loss	0.109 ± 0.011	0.091 ± 0.007	0.799 ± 0.010	0.805 ± 0.009	0.004 ± 0.000	0.001 ± 0.006
	Explanation loss	0.095 ± 0.009	0.094 ± 0.007	0.811 ± 0.004	0.805 ± 0.005	0.004 ± 0.009	0.005 ± 0.004
	Answer logit	0.195 ± 0.011	0.071 ± 0.004	0.734 ± 0.014	0.811 ± 0.010	−0.003 ± 0.011	0.001 ± 0.012
	Answer margin	0.200 ± 0.003	0.072 ± 0.003	0.728 ± 0.003	0.797 ± 0.013	0.012 ± 0.016	0.001 ± 0.006
LESS	Response loss	0.113 ± 0.008	0.087 ± 0.003	0.799 ± 0.009	0.620 ± 0.294	0.189 ± 0.294	0.005 ± 0.009
	Explanation loss	0.101 ± 0.005	0.090 ± 0.009	0.809 ± 0.014	0.797 ± 0.001	0.011 ± 0.009	0.003 ± 0.008
	Answer logit	0.211 ± 0.014	0.088 ± 0.009	0.701 ± 0.017	0.813 ± 0.006	−0.004 ± 0.015	0.002 ± 0.009
	Answer margin	0.237 ± 0.023	0.082 ± 0.009	0.681 ± 0.032	0.803 ± 0.010	0.005 ± 0.011	0.005 ± 0.005
Oracle	Harmful poison	**1.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.050 ± 0.008**	**0.759 ± 0.008**	0.003 ± 0.009
	Benign trigger	0.000 ± 0.000	1.000 ± 0.000	0.000 ± 0.000	0.811 ± 0.006	−0.003 ± 0.003	**-0.009 ± 0.003**
	Clean	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	0.805 ± 0.004	0.003 ± 0.009	−0.001 ± 0.004

Table 18: Reference ablation for conditional backdoor poison detection with Qwen3-8B. The four specifications cross whether the query contains the trigger with whether the target is the harmful poison answer. Only the triggered harmful specification matches the induced behavior.

Method	Behavior	Triggered query		Clean query	
		Harmful target	Clean target	Harmful target	Clean target
Grad. norm	/	0.115 ± 0.002	0.115 ± 0.002	0.115 ± 0.002	0.115 ± 0.002
Random	/	0.105 ± 0.006	0.105 ± 0.006	0.105 ± 0.006	0.105 ± 0.006
RepSim	/	0.492 ± 0.283	$\mathbf{0.492 \pm 0.283}$	0.075 ± 0.002	0.075 ± 0.002
GradSim	Response loss	0.092 ± 0.007	0.136 ± 0.019	0.119 ± 0.017	0.091 ± 0.005
	Explanation loss	0.090 ± 0.006	0.090 ± 0.006	0.092 ± 0.005	0.092 ± 0.005
	Answer logit	0.221 ± 0.027	0.148 ± 0.018	0.122 ± 0.019	0.115 ± 0.007
	Answer margin	0.180 ± 0.016	0.146 ± 0.018	0.122 ± 0.019	0.125 ± 0.016
TracIn	Response loss	0.481 ± 0.223	0.062 ± 0.004	0.301 ± 0.003	0.101 ± 0.006
	Explanation loss	0.104 ± 0.014	0.104 ± 0.014	0.103 ± 0.008	0.103 ± 0.008
	Answer logit	$\mathbf{0.872 \pm 0.028}$	0.061 ± 0.001	0.308 ± 0.004	$\mathbf{0.165 \pm 0.183}$
	Answer margin	0.782 ± 0.049	0.056 ± 0.003	$\mathbf{0.309 \pm 0.004}$	0.061 ± 0.002
TRAK	Response loss	0.089 ± 0.003	0.141 ± 0.013	0.149 ± 0.019	0.091 ± 0.001
	Explanation loss	0.087 ± 0.004	0.087 ± 0.004	0.090 ± 0.001	0.090 ± 0.001
	Answer logit	0.193 ± 0.020	0.143 ± 0.013	0.151 ± 0.019	0.114 ± 0.013
	Answer margin	0.184 ± 0.014	0.142 ± 0.014	0.151 ± 0.019	0.141 ± 0.013
LESS	Response loss	0.091 ± 0.003	0.155 ± 0.015	0.147 ± 0.031	0.093 ± 0.002
	Explanation loss	0.088 ± 0.003	0.088 ± 0.003	0.093 ± 0.002	0.093 ± 0.002
	Answer logit	0.183 ± 0.024	0.154 ± 0.014	0.148 ± 0.033	0.111 ± 0.002
	Answer margin	0.207 ± 0.022	0.154 ± 0.015	0.148 ± 0.033	0.130 ± 0.017

Table 19: Signal-augmented representation similarity for noisy label detection at $\rho = 0.2$. Entries in parentheses denote the signal space. Blue shading marks the preferred behavior surrogate, and bold indicates the best result in each column. Results are averaged over three training seeds.

Method	Behavior	AUPRC	AUROC	P@1%	P@5%	P@10%
RepSim	/	0.218 ± 0.011	0.529 ± 0.009	0.289 ± 0.074	0.237 ± 0.044	0.222 ± 0.028
+ train (logit)	/	$\mathbf{0.337 \pm 0.003}$	$\mathbf{0.628 \pm 0.003}$	$\mathbf{0.615 \pm 0.032}$	$\mathbf{0.495 \pm 0.020}$	0.429 ± 0.002
+ behavior (logit)	Trusted loss	0.221 ± 0.010	0.504 ± 0.002	0.487 ± 0.100	0.254 ± 0.032	0.204 ± 0.019
	Target logit	0.218 ± 0.011	0.529 ± 0.009	0.289 ± 0.074	0.237 ± 0.044	0.222 ± 0.028
	Hard margin	0.218 ± 0.011	0.529 ± 0.009	0.289 ± 0.074	0.237 ± 0.044	0.222 ± 0.028
+ both (logit)	Trusted loss	0.306 ± 0.002	0.620 ± 0.002	0.508 ± 0.013	0.416 ± 0.010	0.377 ± 0.007
	Target logit	$\mathbf{0.337 \pm 0.003}$	$\mathbf{0.628 \pm 0.003}$	$\mathbf{0.615 \pm 0.032}$	$\mathbf{0.495 \pm 0.020}$	0.429 ± 0.002
	Hard margin	0.335 ± 0.003	0.627 ± 0.003	0.609 ± 0.041	0.491 ± 0.021	0.426 ± 0.002
+ train (repr.)	/	0.336 ± 0.003	$\mathbf{0.628 \pm 0.003}$	$\mathbf{0.615 \pm 0.036}$	0.494 ± 0.020	$\mathbf{0.430 \pm 0.002}$
+ behavior (repr.)	Trusted loss	0.224 ± 0.014	0.504 ± 0.003	0.499 ± 0.119	0.269 ± 0.057	0.211 ± 0.032
	Target logit	0.210 ± 0.008	0.521 ± 0.006	0.253 ± 0.064	0.217 ± 0.042	0.207 ± 0.023
	Hard margin	0.209 ± 0.008	0.522 ± 0.005	0.237 ± 0.076	0.206 ± 0.044	0.197 ± 0.028
+ both (repr.)	Trusted loss	0.307 ± 0.002	0.621 ± 0.001	0.502 ± 0.007	0.421 ± 0.010	0.378 ± 0.006
	Target logit	0.336 ± 0.003	$\mathbf{0.628 \pm 0.003}$	$\mathbf{0.615 \pm 0.036}$	0.494 ± 0.020	$\mathbf{0.430 \pm 0.002}$
	Hard margin	0.334 ± 0.003	0.627 ± 0.003	0.607 ± 0.041	0.490 ± 0.021	0.428 ± 0.003

Table 20: Signal-augmented representation similarity for response corruption on Qwen3-8B. Entries in parentheses denote the signal normalization, and bold indicates the best result in each column. Results are averaged over three training seeds.

Answer corruption, answer logit

Method	L24	L28	L32	L35
RepSim	0.100 ± 0.004	0.100 ± 0.004	0.100 ± 0.004	0.100 ± 0.004
+ train (raw)	0.052 ± 0.000	0.052 ± 0.000	0.052 ± 0.000	0.466 ± 0.007
+ train (l2)	0.052 ± 0.000	0.052 ± 0.000	0.052 ± 0.000	0.069 ± 0.001
+ behavior (raw)	0.052 ± 0.000	0.052 ± 0.000	0.052 ± 0.000	0.466 ± 0.007
+ behavior (l2)	0.052 ± 0.000	0.052 ± 0.000	0.052 ± 0.000	0.069 ± 0.001
+ both (raw)	0.976 ± 0.007	0.978 ± 0.005	0.978 ± 0.004	$\mathbf{0.557 \pm 0.005}$
+ both (l2)	$\mathbf{0.979 \pm 0.009}$	$\mathbf{0.981 \pm 0.009}$	$\mathbf{0.984 \pm 0.006}$	0.320 ± 0.065

Rationale corruption, explanation loss

Method	L24	L28	L32	L35
RepSim	0.102 ± 0.005	0.102 ± 0.005	0.102 ± 0.005	0.102 ± 0.005
+ train (raw)	0.066 ± 0.003	0.073 ± 0.004	0.075 ± 0.002	$\mathbf{0.195 \pm 0.014}$
+ train (l2)	0.054 ± 0.000	0.056 ± 0.000	0.056 ± 0.000	0.079 ± 0.003
+ behavior (raw)	0.066 ± 0.003	0.073 ± 0.004	0.075 ± 0.002	$\mathbf{0.195 \pm 0.014}$
+ behavior (l2)	0.054 ± 0.000	0.056 ± 0.000	0.056 ± 0.000	0.079 ± 0.003
+ both (raw)	0.479 ± 0.024	$\mathbf{0.417 \pm 0.003}$	$\mathbf{0.445 \pm 0.043}$	0.165 ± 0.015
+ both (l2)	$\mathbf{0.509 \pm 0.023}$	0.392 ± 0.004	0.359 ± 0.009	0.116 ± 0.007