

# EULoInf: Efficient Hessian-Free Entropy Based Uncertainty-Aware Data Influence Approximation

Runxin Cai<sup>\*1</sup> Jingtan Wang<sup>\*1,2</sup> Bryan Kian Hsiang Low<sup>1</sup>

<sup>1</sup>Department of Computer Science, National University of Singapore

<sup>2</sup>Agency for Science, Technology and Research (A\*STAR)

{runxin-cai, jingtan.w}@u.nus.edu dcs1kh@nus.edu.sg

## Abstract

In Large Language Model post-training, high-quality data effectively enhances model performance with fine-tuning, highlighting the need to identify high-quality and beneficial fine-tuning data. However, one of the most popular data valuation paradigms, influence function and its variants, is computationally expensive due to its reliance on inverse Hessian-Vector Products (iHVP) computations that scale poorly with increasing model size. To examine whether influence values correlate with efficiently computable intrinsic features, we empirically investigate the distribution of top influential data for the model in fine-tuning, and observe that data with high influence tend to be those with high predictive uncertainty. Yet such highly uncertain samples exhibit a dual nature, which can be either beneficial or detrimental noisy data. Unlike traditional methods that treat uncertainty as a standalone criterion, we introduce a directional indicator to rigorously disentangle these opposing effects. Formally, we propose EULoInf (Entropy-based Uncertainty-aware Lookahead Influence), a computationally efficient valuation framework. By approximating influence via uncertainty and gradient-based validation loss lookahead, EULoInf avoids iHVP computation, effectively reducing the iHVP-induced quadratic complexity in model parameters to linear time. We rigorously derive our framework from the influence function. Empirically, it matches or even outperforms prior methods across diverse data valuation tasks and LLM architectures, including mislabel detection and data selection, while reducing computational time and memory usage by over 50%. The code is available at [github.com/Rachelcoll/EULoInf](https://github.com/Rachelcoll/EULoInf).

## 1 Introduction

Large Language Models (LLMs) have demonstrated remarkable ability in human instruction fol-

lowing, mathematical reasoning, and domain adaptation (Ouyang et al., 2022; Ahn et al., 2024; Cheng et al., 2023) by fine-tuning on relevant datasets. Extensive studies reveal that the effectiveness of fine-tuning heavily relies on the quality of training data (Chen et al., 2023; Zhou et al., 2023; Ye et al., 2025). However, the large corpus of training data often varies in quality and may contain noisy labels and biased outputs (Carlini et al., 2024; Wang et al., 2024), making it crucial to assess the contribution of each training data point. Data valuation addresses this challenge by quantifying the impact of specific data points on performance. One of the most principled data valuation methods is influence function, which effectively values individual training samples by how they affect validation loss based on solid theoretical derivation (Hampel, 1974; Cook and Weisberg, 1980; Koh and Liang, 2017).

Despite the demonstrated efficacy of influence function, its calculation of the inverse Hessian matrix makes it extremely computationally expensive. Numerous works target efficient approximation of influence function by accelerating or avoiding the computation of inverse Hessian. However, they either suffer from quadratic time complexity (Kwon et al., 2024; Zhou et al., 2025) and substantial memory (Tu et al., 2025), or rely on retraining of multiple models (Pan et al., 2025; Kreer et al., 2025). There remains a non-negligible gap in achieving an optimal trade-off between decent approximation accuracy and computational and memory feasibility. Data valuation can lose its practical utility if the cost of assessing data influence outweighs the expense of brute-force alternatives.

To improve estimation accuracy while minimizing computational overhead, we investigate the relationship between data influence and efficiently computable intrinsic data properties. Prior studies suggest that high-impact data typically exhibit high predictive uncertainty (Malinin, 2019; Malinin and

<sup>\*</sup>Equal Contribution

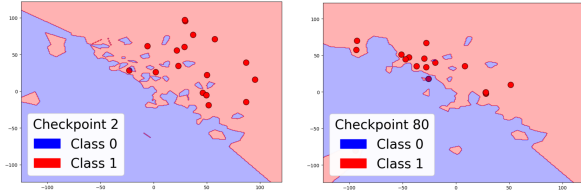


Figure 1: Distribution of the most beneficial points in terms of influence function for a binary classification task. The top influential points become closer to decision boundary in later checkpoints.

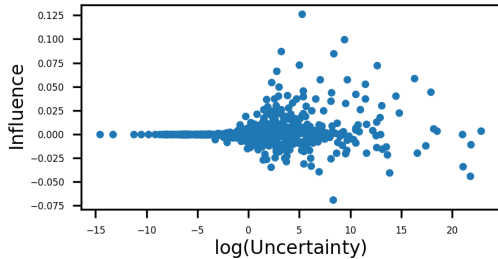


Figure 2: Relationship between predictive uncertainty and influence score ( $\rho = 0.10$ ). High uncertainty samples contain both beneficial and detrimental high influential data. More details are shown in App. B.2.

Gales, 2020). Our analysis on the distributional dynamics of data with the highest influence function scores aligns with this finding, as top influential points progressively concentrate near the decision boundary in Fig. 1. Intuitively, for those samples near the decision boundary, they can exhibit high predictive uncertainty, as the model is less confident in distinguishing between classes. This connection motivates our use of efficiently computed predictive uncertainty as a proxy for data influence, with a principled justification of their correlation in Sec. 3.1. However, we further notice a duality in high uncertainty data which could impact the model in either a positive or negative direction, as illustrated in Fig. 2. More details of Fig. 1 and Fig. 2 are in App. B.2. To address this issue, we find that a lightweight gradient projection on top of the predictive uncertainty proxy can efficiently disentangle its directional polarity. Building upon these two observations, we propose EULoInf (Entropy-based Uncertainty-aware Lookahead Influence), which enables precise influence function approximation with minimal computational overhead, making it a highly scalable and practical data valuation method, especially for LLMs. Our main contributions are summarised as follows.

- In Sec. 3.1, we derive that the uncertainty mea-

sured by predictive entropy can capture the absolute influence of the training data under a mild assumption.

- In Sec. 3.2, we propose EULoInf, a Hessian-free efficient data valuation method as a combination of predictive entropy and gradient alignment component to distinguish beneficial samples within high uncertainty data. We further generalize it to LLMs in Sec. 3.3.
- In Sec. 4, we empirically demonstrate that EULoInf exhibits superior performance and efficiency over existing methods across various tasks under different LLMs. For instance, EULoInf surpasses the best baseline on GSM8K by 2.93%, while requiring only 41.13% of the computation time and 46.83% of the memory used by RRInf (Tu et al., 2025).

## 2 Background and Preliminaries

**Setup.** Let  $\mathcal{D}$  be a dataset from an input space  $\mathcal{X}$  to a label space  $\mathcal{Y}$ . Assume training data  $\mathcal{D}_{train} := \{z_i := (x_i, y_i)\}_{i=1}^n$  is drawn *i.i.d.* from some distribution  $\mathcal{P}$  over  $\mathcal{X} \times \mathcal{Y}$ . Similarly, we denote the validation set by  $\mathcal{D}_{val} := \{z_j^{val} := (x_j^{val}, y_j^{val})\}_{j=1}^m$ . In later sections, we analyze quantities w.r.t. a single general validation data point; for notational simplicity, we omit the index and denote such a sample by  $z_{val} := (x_{val}, y_{val})$ . Given the parameter space  $\Theta$  and loss function  $\ell$ , for a model parameterized by  $\theta \in \Theta$ , we denote loss of  $z_i \in \mathcal{D}_{train}$  as  $\ell(\theta, i)$ , and its gradient w.r.t.  $\theta$  as  $\nabla_{\theta}\ell(\theta, i)$ . Similarly, loss of  $z_{val} \in \mathcal{D}_{val}$  is  $\ell(\theta, val)$ , and the gradient is  $\nabla_{\theta}\ell(\theta, val)$ . The empirical risk minimizer of the model is defined as  $\hat{\theta} := \operatorname{argmin}_{\theta \in \Theta} \mathcal{L}(\theta)$ , where  $\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(\theta, i)$ .

**Influence Function.** Influence function simulates the change of model parameters if a training data were infinitesimally upweighted in the training objective (Hampel, 1974; Cook et al., 1982; Koh and Liang, 2017). For a training point  $z_k \in \mathcal{D}_{train}$ , consider up-weight  $z_k$  by some small  $\epsilon$  in the risk minimization problem:  $\hat{\theta}_{\epsilon, k} = \operatorname{argmin}_{\theta \in \Theta} (\mathcal{L}(\theta) + \epsilon \ell(\theta, k))$ . Influence of up-weighting  $z_k$  to loss of a validation sample  $z_{val}$  is  $I_{up, loss}(k, val) := \left. \frac{d\ell(\hat{\theta}_{\epsilon, k}, val)}{d\epsilon} \right|_{\epsilon=0}$ . Assuming the loss function is strongly convex and twice-differentiable,  $I_{up, loss}(k, val) = -\nabla_{\theta}\ell(\hat{\theta}, val)^T H_{\hat{\theta}}^{-1} \nabla_{\theta}\ell(\hat{\theta}, k)$ , where  $H_{\hat{\theta}} := \nabla_{\theta}^2 \mathcal{L}(\hat{\theta})$  and is positive definite by assumption.

**Influence Function Estimation in Neural Network.** In practice, model parameters  $\tilde{\theta}$  after training are often acquired with early stopping or non-convex objectives, so  $\tilde{\theta} \neq \hat{\theta}$ . To ensure the Hessian is positive-definite and invertible, usually a damping term  $\lambda I$  ( $\lambda > 0$ ) is added to the Hessian approximation. This modification is equivalent to adding  $L_2$  regularization to the objective function (Martens et al., 2010). From this perspective, influence functions with damping can be viewed as linearizing the *proximal response function* at  $\epsilon = 0$ :

$$\hat{\theta}_{\epsilon,k} = \operatorname{argmin}_{\theta \in \Theta} \left( \mathcal{L}(\theta) + \epsilon \ell(\theta, k) + \frac{\lambda}{2} \|\theta - \hat{\theta}\|_2^2 \right), \quad (1)$$

where  $\frac{\lambda}{2} \|\theta - \hat{\theta}\|_2^2$  is the damping term (Farnia and Ozdaglar, 2020). This formulation effectively approximates influence function in a warm-start re-training scheme that explicitly penalizes the  $L_2$  distance between the  $\hat{\theta}_{\epsilon,k}$  and  $\hat{\theta}$  (Bae et al., 2022).

### 3 Methodology

In Sec. 3.1, we first estimate the absolute influence of training data by the change of model parameter after perturbation with the proximal response function, then further approximate it using predictive uncertainty derived from the probabilistic label distribution. Since high uncertainty data can contain both beneficial and detrimental influential points as shown in Fig. 2, we then illustrate how to distinguish beneficial and detrimental influential data by sign in Sec. 3.2. Finally, we will mention practical implementation details when adapting EULOInf to LLM in Sec. 3.3.

#### 3.1 Influence Estimation with Uncertainty

Following notation of Sec. 2, we denote the parameter change after upweighting as  $\Delta\theta = \hat{\theta}_{\epsilon,k} - \hat{\theta}$ . The magnitude of the parameter update  $\|\Delta\theta\|$  is commonly used as a measure of absolute influence, reflecting the impact of a training data on model parameters (Koh and Liang, 2017; Kwon et al., 2024).

From Eqn. 1, the objective after perturbation can be viewed as linearizing the proximal response function:  $J(\theta) = \mathcal{L}(\theta) + \epsilon \ell(\theta, k) + \frac{\lambda}{2} \|\theta - \hat{\theta}\|_2^2$ . Let  $\delta = \theta - \hat{\theta}$ , then  $J(\theta) = J(\hat{\theta} + \delta)$ , and  $\Delta\theta$  can

be seen as the minimizer of  $J(\hat{\theta} + \delta)$ :

$$\begin{aligned} \Delta\theta &= \operatorname{argmin}_{\delta} J(\hat{\theta} + \delta) \\ &= \operatorname{argmin}_{\delta} \left( \mathcal{L}(\hat{\theta} + \delta) + \epsilon \ell(\hat{\theta} + \delta, k) + \frac{\lambda}{2} \|\delta\|_2^2 \right). \end{aligned} \quad (2)$$

Generally, when the dataset size  $n$  is large and  $\epsilon$  is small, the effect of upweighting a single data by  $\epsilon$  on model parameter update is minimal, implying that the optimal update  $\delta$  is essentially small. Therefore, terms in  $J(\hat{\theta} + \delta)$  can be approximated with first-order Taylor approximation as:

$$\mathcal{L}(\hat{\theta} + \delta) \approx \mathcal{L}(\hat{\theta}) + \nabla_{\theta} \mathcal{L}(\hat{\theta})^T \delta. \quad (3)$$

$$\epsilon \ell(\hat{\theta} + \delta, k) \approx \epsilon \ell(\hat{\theta}, k) + \epsilon \nabla_{\theta} \ell(\hat{\theta}, k)^T \delta. \quad (4)$$

The damping term also serves as a robust local curvature proxy in the region where the first-order expansion is reliable, as the  $L_2$  regularization term explicitly penalizes the magnitude of  $\delta$ . Following previous work, we assume that the optimal parameters before perturbation  $\hat{\theta}$  reach a local minimum, so  $\nabla_{\theta} \mathcal{L}(\hat{\theta}) \approx 0$ . Substituting the convergence condition into Eqn. 3, and then plugging the simplified terms from Eqns. 3 and 4 into Eqn. 2, we can get:

$$\begin{aligned} \Delta\theta &= \operatorname{argmin}_{\delta} \\ &\left( L(\hat{\theta}) + \epsilon \ell(\hat{\theta}, k) + \epsilon \nabla_{\theta} \ell(\hat{\theta}, k)^T \delta + \frac{\lambda}{2} \|\delta\|_2^2 \right). \end{aligned} \quad (5)$$

Ignoring terms constant w.r.t.  $\delta$ ,  $\Delta\theta$  is the solution to the following proximal optimization problem:

$$\Delta\theta = \operatorname{argmin}_{\delta} \left( \epsilon \nabla_{\theta} \ell(\hat{\theta}, k)^T \delta + \frac{\lambda}{2} \|\delta\|_2^2 \right). \quad (6)$$

This formulation minimizes the linearized proximal objective and keeps the parameters close to the optimal parameters at  $\epsilon = 0$ . Solving this optimization yields a closed-form solution  $\Delta\theta = -\frac{\epsilon}{\lambda} \nabla_{\theta} \ell(\hat{\theta}, k)$ . Therefore, absolute influence captured by magnitude of parameter change  $\|\Delta\theta\|_2^2$  is positively correlated to gradient norm  $\|\nabla_{\theta} \ell(\hat{\theta}, k)\|_2^2$ .

**Influence Estimation with Predictive Uncertainty.** However, computing the exact gradient norm for all parameters is still expensive. Inspired by the findings on influence and uncertainty, we aim to establish a rigorous connection between the gradient norm and the model’s predictive uncertainty for efficient estimation.

Now we consider a training data  $z = (x, y)$  with train loss  $\ell(\hat{\theta}, z)$  and gradient  $\nabla_{\theta}\ell(\hat{\theta}, z)$ . For a classification task with cross entropy loss and softmax activation, we define  $p(x)$  to be the categorical distribution over class labels induced by the softmax output for input  $x$ . Under a simple neural network,  $p(x) = \text{Softmax}(r(x))$ , where  $r(x) = \theta^T x + b$  and  $b$  denotes the bias. The gradient can be expressed as  $\nabla_{\theta}\ell(\hat{\theta}, z) = \frac{\partial\ell}{\partial r(x)} \cdot \frac{\partial r(x)}{\partial\theta} = (p(x) - y)x^T$ . The gradient norm is then given by:

$$\|\nabla_{\theta}\ell(\hat{\theta}, z)\|_2^2 = \|p(x) - y\|_2^2 \cdot \|x\|_2^2. \quad (7)$$

In practice, input features are typically normalized or standardized, either before training or via normalization layers, which helps ensure to some extent that  $\|x\|$  does not vary excessively across data points within dataset (Ioffe and Szegedy, 2015; Ba et al., 2016). However,  $\|p(x) - y\|_2^2$  can vary significantly between samples during training, especially for a randomly initialized model. Accordingly, we reasonably assume variation in gradient magnitude is primarily driven by  $\|p(x) - y\|_2^2$ .

In a conventional classification task setup, the true label  $y$  is deterministic for each input data  $x$ . Let  $\mathcal{C} = \{1, 2, \dots, C\}$  denote the output labels of  $C$  classes. Suppose  $\mathbf{e}_c$  is a one-hot vector with  $c$ -th element equal to 1, the label  $y = \mathbf{e}_c$  if  $c$  is the true class for  $x$ . Under this definition,  $\|p(x) - y\|_2^2$  of  $x$  with true label  $c$  can be expressed as

$$\|p(x) - y\|_2^2 = (1 - p_c(x))^2 + \sum_{i \neq c} p_i(x)^2. \quad (8)$$

In real-world scenarios, label uncertainty may exist due to inherent annotation ambiguity and other intrinsic noise, so we treat label  $y$  as a random vector conditioned on  $x$ . Specifically, given  $x$ , we assume  $y$  follows a conditional categorical distribution over the class labels, where each realization is represented as a one-hot vector. Let  $q(x) = (q_1(x), q_2(x), \dots, q_C(x))$  be a probability vector such that  $\forall c \in \mathcal{C}, q_c(x) > 0$  and  $\sum_{c=1}^C q_c(x) = 1$ . Then  $q_c(x) := P(y = \mathbf{e}_c | x)$  for  $\forall c \in \mathcal{C}$  is a categorical distribution regarding random variable  $y|x$ . Under this probabilistic label

setting, expected value of  $\|p(x) - y\|_2^2$  over  $y|x$  is

$$\begin{aligned} E_{y|x}[\|p(x) - y\|_2^2] &= \sum_{c=1}^C q_c(x) \cdot \|p(x) - \mathbf{e}_c\|_2^2 \\ &= \sum_{c=1}^C q_c(x) \cdot \left( (1 - p_c(x))^2 + \sum_{i \neq c} p_i(x)^2 \right). \end{aligned} \quad (9)$$

Under the asymptotic assumption where the model captures the true conditional distribution (i.e.,  $p_c(x) \rightarrow q_c(x)$ ), simplifying Eqn. 9 yields:

$$\mathbb{E}_{y|x}[\|p(x) - y\|_2^2] \approx 1 - \sum_{i=1}^C p_i(x)^2, \quad (10)$$

where a detailed derivation is in App. A.1. This approximation is most accurate when the model's predictive distribution is close to the empirical label distribution after sufficient fine-tuning. Note that the term  $\sum_{i=1}^C p_i(x)^2$  is the core component of the second-order Rényi entropy  $H_2(p(x)) = -\log \sum_{i=1}^C p_i(x)^2$  (Rényi, 1961). Empirical justification of Eqn. 7 and Eqn. 10 is shown in App. D. This entropy metric effectively captures the prediction uncertainty of the model, as maximizing  $H_2$  entropy is equivalent to minimizing the divergence from the uniform distribution. Generally, a uniform distribution represents maximal uncertainty, so minimizing divergence from uniform distribution indicates a shift towards a more uncertain prediction. The relationship between Rényi entropy and divergence is shown in App. A.2.

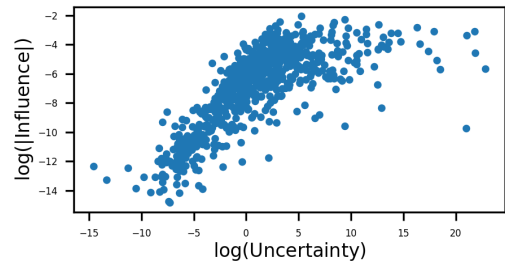


Figure 3: Relationship between predictive entropy and absolute value of influence function with Spearman correlation  $\rho = 0.73$ , showing effectiveness of entropy to capture quantity of effect the data has to model.

Since  $\sum_{i=1}^C p_i(x)^2 = e^{-H_2(p(x))}$ , we have  $E_{y|x}[\|p(x) - y\|_2^2] \approx 1 - e^{-H_2(p(x))}$ . For each determined  $x$ , higher predictive entropy implies a

greater expected value for  $\|p(x) - y\|_2^2$ , thereby yielding a greater absolute influence. Fig. 3 shows the correlation between entropy and absolute influence, demonstrating the effectiveness of proximal optimization formulation with first-order Taylor approximation in Eqn. 6 to capture  $\|\Delta\theta\|$  via gradient norm  $\|\nabla_{\theta}\ell(\hat{\theta}, k)\|_2^2$ , and predictive uncertainty as a robust proxy for the expected gradient norm.

### 3.2 Directional Indicator Sign

While  $\|\Delta\theta\|$  captures the magnitude of training data’s influence on model parameters, it fails to reveal whether this influence is beneficial or detrimental, as shown in Fig. 2. To address this, we introduce a validation loss lookahead mechanism to explicitly determine the direction of its impact on the validation loss. Specifically, we focus on the effect of upweighting (i.e.  $\epsilon > 0$ ) training data  $z_k$  on the validation data  $z_{val}$ . Define  $\Delta\ell = \ell(\hat{\theta}_{\epsilon,k}, val) - \ell(\hat{\theta}, val)$ . Generally, upweighting beneficial data is expected to induce a negative  $\Delta\ell$ , whereas detrimental data typically yield a positive  $\Delta\ell$ . Under the proximal objective of perturbation, the update of model parameters is generally small. Therefore, we can arrive at  $\Delta\ell = \nabla_{\theta}\ell(\hat{\theta}, val)^T \Delta\theta$  with first-order Taylor approximation. Substituting the closed-form solution  $\Delta\theta = -\frac{\epsilon}{\lambda} \nabla_{\theta}\ell(\hat{\theta}, k)$  of Eqn. 6 back to  $\Delta\ell$ , we have  $\Delta\ell = -\frac{\epsilon}{\lambda} \nabla_{\theta}\ell(\hat{\theta}, val)^T \nabla_{\theta}\ell(\hat{\theta}, k)$ . Since  $\epsilon > 0$  and  $\lambda > 0$ , sign of  $\nabla_{\theta}\ell(\hat{\theta}, val)^T \nabla_{\theta}\ell(\hat{\theta}, k)$  can serve as an indicator of the training data’s impact, aligning with the sign of  $-\Delta\ell$ . Specifically, a positive inner product implies a negative  $\Delta\ell$ , signifying that the corresponding data point is beneficial to the model’s performance, whereas a negative value suggests a detrimental effect. Combining results in Sec. 3.1, we propose EULoInf to express influence of a train data  $z_k$  as:

$$I(z_{val}, z_k) = H_2(p(x_k)) \cdot \text{sign}(\nabla_{\theta}\ell(\hat{\theta}, val)^T \nabla_{\theta}\ell(\hat{\theta}, k)). \quad (11)$$

Consequently, a higher EULoInf score indicates a more beneficial training data point.

### 3.3 Practical Generalization to LLMs

The next token prediction loss of the Large language model can be seen as a sequence of multi-class classification tasks. Accordingly, the influence of a sentence corresponds to the sum of each token’s influence. Inspired by Fu et al. (2025),

the informativeness of tokens in one sentence differs, and selecting informative tokens only in influence calculation results in higher evaluation quality. Following this insight, we select the top 50% tokens with values based on their token-level predictive entropy. While the model is fine-tuned using LoRA adapters applied to all attention layers (Hu et al., 2021), we compute the training and validation loss gradient of directional indicator sign exclusively w.r.t. the final layer for efficiency in EULoInf. Previous works have also adopted the trick of calculating gradients w.r.t. only the last layer, demonstrating its effectiveness (Barshan et al., 2020; Katharopoulos and Fleuret, 2018).

## 4 Experiments and Results

In this section, we validate our proposed method, EULoInf, in terms of both performance and computational efficiency on representative data valuation tasks, including mislabeled data detection and data selection. We analyze its asymptotic complexity and empirical resource overhead, and conduct an ablation study to validate the adoption of the directional indicator in EULoInf. Correlation and qualitative analyses are provided in App. C.1 and C.2 to further demonstrate the effectiveness of the proxy. Overall, EULoInf consistently demonstrates high efficiency as model scale increases and superior performance across both tasks on different models.

### 4.1 Setup

We comprehensively evaluate our method, EULoInf, on diverse tasks including mathematical reasoning, question-answering, and summarization.

**Models and Hyperparameters.** To demonstrate the generalizability of our method across different model families, we employ two widely adopted open-source Large Language Models, Llama-3-8B (Grattafiori et al., 2024), and Mistral-7B (Jiang et al., 2023). Training details are in App. B.1.

**Datasets and Evaluation Tasks.** We select three datasets and corresponding evaluation tasks to evaluate the performance of EULoInf on different settings. For mathematical reasoning, we use high-quality mathematical and logical reasoning problems of Persona Hub (Lambert et al., 2024; Ge et al., 2025) with CoT for training, and evaluate on the GSM8K benchmark (Cobbe et al., 2021). We report the exact match accuracy to measure the strict correctness of the solution steps and final answers. For the question-answering task, we adopt

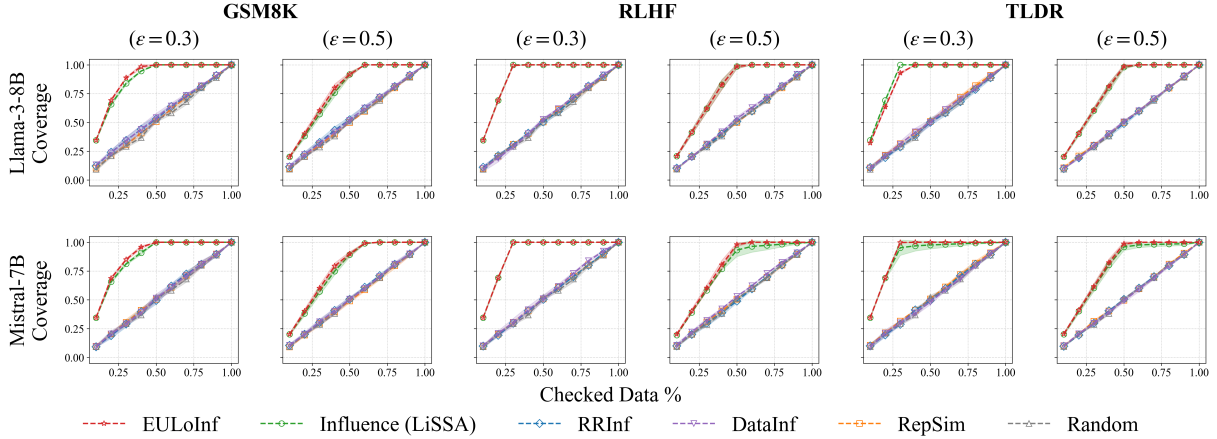


Figure 4: **Mislabel Detection:** It shows the detected percentage of poisoned data when inspecting data from least to most beneficial under different poisoning rate  $\epsilon$ . In most cases, EULoInf can have higher detection rates. While LiSSA also achieves competitive performance, it incurs a higher computational overhead compared to EULoInf as shown in Tab. 1.

the HH-RLHF dataset (Bai et al., 2022), which consists of prompt-response pairs. We only use the "chosen" response that aligns better with human preferences in fine-tuning. We use a trained reward model (Yang et al., 2024) on its test split as the evaluation metric. For the summarization task, we utilize the Tulu3 mixture dataset (Lambert et al., 2024) for training, and test on the TLDR dataset (Stiennon et al., 2020) that contains polished text summaries. The evaluation metric is given by the reward model (Kopf et al., 2023) that measures the quality of summaries and alignment with human preference.

**Baselines.** We compare our method against a random selection baseline and established influence function approximations. LiSSA (Agarwal et al., 2017) is a classical influence estimation that approximates iHVP with an iterative approach. We also compare to DataInf (Kwon et al., 2024), RRInf (Tu et al., 2025), two representative approaches tailored for efficient influence estimation in LLM, and RepSim (Li et al., 2024), a recent influence approximation that evaluates training data by similarity to the validation set representations. In all experiments, we first fine-tune a model and then compute the influence function on the same fine-tuned checkpoint. Additional gradient-based baselines and their experimental results are shown in App. C.4.

## 4.2 Data Valuation Tasks

**Mislabel Detection.** In fine-tuning, datasets containing mislabeled data can increase validation loss

and cause model performance degradation (Frénay and Verleysen, 2013; Steinhardt et al., 2017; Qiao and Low, 2024). In this experiment, we poison each dataset by applying token-level label corruption on 30% and 50% of the training data respectively, then automatically examine the poisoned data by reviewing the data points ranked as least beneficial by each respective method. We then measure the proportion of poisoned data among the lowest influence subsets under different ratios. As shown in Fig. 4, EULoInf achieves parity with the computation-heavy LiSSA, and can generally identify more mislabeled data under the same budget, showing a superior balance between effectiveness and efficiency. By combining predictive uncertainty and a directional indicator, EULoInf precisely targets harmful, high-uncertainty samples that degrade validation loss.

**Data Selection.** To further validate the effectiveness of EULoInf in identifying high-value training data, we conduct data selection experiments. Following the setting of previous work (Kwon et al., 2024), we rank data points based on influence scores derived from each method. We then fine-tune the model using subsets ranging from the top 10% to 90% of ranked data. As illustrated in Fig. 5, EULoInf outperforms baseline methods across three tasks in most selection percentages, demonstrating superior generalizability. Generally, fine-tuning on the top 50% of data by EULoInf consistently outperforms any baseline methods on most data fractions. By successfully identifying the high-quality subsets, our method enables fraction-

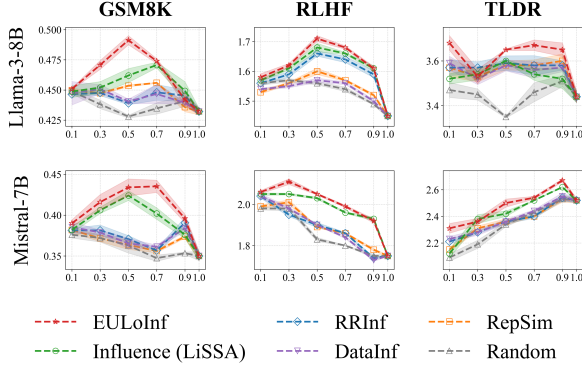


Figure 5: **Data Selection:** Difference between the model performance when trained on a subset that contains data points with the top-most beneficial samples. The subset size is given as a percentage of the full set. The higher the score on the test dataset, the better the approach.

scale training to outperform full-scale baselines, demonstrating a critical advantage in maximizing data efficiency.

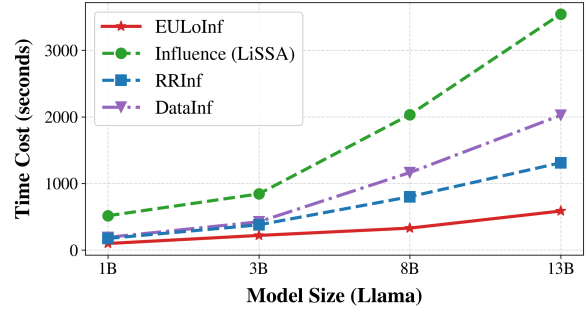
### 4.3 Efficiency

| Complexity  | Exact               | LiSSA               | DataInf             | RRInf             | EULoInf          |
|-------------|---------------------|---------------------|---------------------|-------------------|------------------|
| Computation | $\mathcal{O}(p^3L)$ | $\mathcal{O}(p^2L)$ | $\mathcal{O}(p^2L)$ | $\mathcal{O}(pL)$ | $\mathcal{O}(p)$ |
| Memory      | $\mathcal{O}(p^2L)$ | $\mathcal{O}(p^2)$  | $\mathcal{O}(pL)$   | $\mathcal{O}(pL)$ | $\mathcal{O}(p)$ |

Table 1: Complexity comparison between Exact by influence function definition, LiSSA, DataInf, RRInf, and EULoInf. Comparisons are given assuming the number of parameters is identical at each layer for a multilayer perceptron (MLP) with  $L$  layers and  $p$  parameters at each layer. Overall, EULoInf achieves higher computational and memory efficiency than baseline methods. The table reports per-example complexity; time of computing influence of all training examples scales linearly with  $|\mathcal{D}_{train}|$  for all methods.

**Complexity Analysis.** Tab. 1 compares the time and memory complexity of different hessian-based influence function estimation methods. Approaches like original influence function, LiSSA, RRInf, and DataInf require computation of iHVP or its approximation form, resulting in high time and memory costs and poor scalability with increasing model depth and size. In contrast, EULoInf is highly efficient by combining the predictive uncertainty computed from logits during a forward pass with a lightweight directional indicator computed simply with last layer gradient information.

**Time and Memory Cost Analysis.** We also compare the actual time and memory cost of EULoInf



(a) Time Cost

| Model      | LiSSA | DataInf | RRInf | EULoInf      |
|------------|-------|---------|-------|--------------|
| Mistral-7B | 19.13 | 32.61   | 37.10 | <b>15.18</b> |
| Llama-3-8B | 25.13 | 42.59   | 38.58 | <b>18.85</b> |

(b) Peak Allocated Memory (GB)

Figure 6: **Time and memory cost:** (a) Time cost of EULoInf and baseline methods on mathematical reasoning task across different model sizes measured in single L40 GPU seconds. (b) Peak memory usage during computation of each method. Notably, EULoInf is significantly more efficient than other baselines in terms of time and space. Note that LiSSA is computed with less layers.

against baseline methods on mathematical reasoning task for 1000 data across different sizes of Llama models. As illustrated in Fig. 6, baseline methods for influence function estimation generally incur significantly higher computational overhead. While RRInf demonstrates improved computational speed compared to other baselines, its memory footprint scales unfavorably with model size, imposing substantial resource overhead on larger models. LiSSA, despite being a closer approximation of influence function theoretically, becomes extremely time-consuming for large-scale models. In contrast, EULoInf achieves the lowest runtime and memory consumption across all models, taking only 16.51% of LiSSA’s computation time and 47.95% of memory resources required by RRInf, underscoring its practical scalability. This highlights the distinct advantage of EULoInf to deliver competitive or superior performance with minimal computational cost.

### 4.4 Ablation Study

#### Directional Indicator and Uncertainty Proxy.

We first evaluate whether incorporating directional indicators into predictive entropy is necessary by comparing EULoInf with predictive entropy without sign. We then compare EULoInf with signed gradient norm to assess whether predictive entropy

serves as a more reliable proxy for absolute influence than the gradient magnitude, and further compare the memory cost of EULoInf with signed gradient norm, since predictive uncertainty primarily aims to reduce the resource overhead associated with estimating absolute influence. This two-step analysis clarifies the respective roles of the directional indicator and predictive uncertainty in approximating influence for the data selection task. Specifically, we evaluate the performance at selection percentages 30% and 50% respectively.

As shown in Tab. 2, removing the directional indicator from predictive entropy leads to a clear drop in data selection performance, indicating that the sign component is essential for distinguishing beneficial samples from harmful high-uncertainty outliers. In terms of the comparison between gradient norm and predictive entropy, we observe that both EULoInf and signed gradient norm achieve similar performance. However, computing gradient norms requires full backpropagation and therefore incurs substantially higher memory costs. This highlights the advantage of our approach, which preserves both magnitude and direction while avoiding the computational overhead associated with gradient-norm-based approximations.

**Effect of Token Selection Threshold.** We conduct a supplementary ablation study on the entropy-based token selection threshold used in the absolute influence calculation of EULoInf. Specifically, we evaluate data selection performance on the TLDR task using EULoInf computed with the top 30% and 70% highest-entropy tokens, and compare these results against our default 50% threshold. As shown in Tab. 3, selecting the top 50% of tokens yields the best test performance. This behavior likely reflects a fundamental trade-off: lower thresholds omit informative tokens and lead to unstable influence estimates, while higher thresholds introduce less informative tokens that dilute the influential signals used for computation. This aligns with our practical generalization insights discussed in Sec. 3.3, confirming that 50% serves as an optimal balance.

## 5 Related Work

Numerous data valuation methods have been proposed to quantify the value of individual training data on model predictions. Influence function (Koh and Liang, 2017), one of the representative, estimates the effect of upweighting a training example

| Dataset                  | Method          | Top 30%                             | Top 50%                             |
|--------------------------|-----------------|-------------------------------------|-------------------------------------|
| <i>Model: Llama-3-8B</i> |                 |                                     |                                     |
| GSM8K                    | <i>w/o Sign</i> | 43.67% $\pm$ 0.29                   | 43.11% $\pm$ 0.60                   |
|                          | GradNorm        | 45.40% $\pm$ 0.65                   | 46.20% $\pm$ 1.06                   |
|                          | EULoInf         | <b>47.08% <math>\pm</math> 0.64</b> | <b>49.13% <math>\pm</math> 0.66</b> |
| RLHF                     | <i>w/o Sign</i> | 1.55 $\pm$ 2.42                     | 1.56 $\pm$ 2.60                     |
|                          | GradNorm        | <b>1.64 <math>\pm</math> 2.44</b>   | <b>1.76 <math>\pm</math> 2.66</b>   |
|                          | EULoInf         | 1.62 $\pm$ 1.50                     | 1.71 $\pm$ 2.23                     |
| TLDR                     | <i>w/o Sign</i> | 3.45 $\pm$ 5.53                     | 3.42 $\pm$ 2.17                     |
|                          | GradNorm        | <b>3.54 <math>\pm</math> 4.31</b>   | <b>3.67 <math>\pm</math> 0.22</b>   |
|                          | EULoInf         | 3.53 $\pm$ 5.70                     | 3.65 $\pm$ 0.64                     |
| <i>Model: Mistral-7B</i> |                 |                                     |                                     |
| GSM8K                    | <i>w/o Sign</i> | 37.27% $\pm$ 0.66                   | 35.00% $\pm$ 0.99                   |
|                          | GradNorm        | 39.93% $\pm$ 0.84                   | 41.33% $\pm$ 1.68                   |
|                          | EULoInf         | <b>41.60% <math>\pm</math> 1.61</b> | <b>43.40% <math>\pm</math> 1.80</b> |
| RLHF                     | <i>w/o Sign</i> | 1.92 $\pm$ 1.61                     | 1.84 $\pm$ 2.36                     |
|                          | GradNorm        | <b>2.13 <math>\pm</math> 1.67</b>   | <b>2.10 <math>\pm</math> 1.82</b>   |
|                          | EULoInf         | 2.11 $\pm$ 3.03                     | 2.05 $\pm$ 1.04                     |
| TLDR                     | <i>w/o Sign</i> | 2.25 $\pm$ 4.22                     | 2.30 $\pm$ 5.23                     |
|                          | GradNorm        | <b>2.38 <math>\pm</math> 4.30</b>   | 2.45 $\pm$ 4.27                     |
|                          | EULoInf         | 2.36 $\pm$ 4.55                     | <b>2.50 <math>\pm</math> 4.21</b>   |

(a) Data Selection Task

| Model      | EULoInf | GradNorm |
|------------|---------|----------|
| Mistral-7B | 19.13   | 35.82    |
| Llama-3-8B | 25.13   | 40.26    |

(b) Peak Memory Allocation (GB)

Table 2: **Ablation Study:** (a) *w/o Sign* denotes using  $H_2$  entropy magnitude only. GradNorm denotes using gradient norm as absolute influence approximation with indicator sign. **Bold** indicates the best result. Values are reported as Mean  $\pm$  Std (%). In most cases, EULoInf outperforms *w/o Sign* and achieves parity with GradNorm. (b) Memory Cost comparison between EULoInf and GradNorm in mathematical reasoning task. Computation of full gradient takes approximately more than 50% of memory consumption.

in the training objective. The computational cost of the influence function remains expensive as it involves the computation of the inverse Hessian matrix, limiting its real-world application. Subsequent work on influence function aims to improve computational efficiency by influence estimation. Kwon et al. (2024) used a closed-form Hessian matrix approximation to further reduce the cost. The efficacy is not guaranteed since the approximation error depends on the number of learnable model parameters, and the time complexity is still quadratic to model size. While Zhou et al. (2025) adopted generalized Fisher information (GFIM) as a low-rank approximation of the Hessian matrix to reduce the approximation error with quadratic time complex-

| Checked | Top-k Tokens |                     |              |
|---------|--------------|---------------------|--------------|
|         | 30%          | 50%                 | 70%          |
| 10%     | 3.58 ± 6.23% | <b>3.68 ± 5.69%</b> | 3.62 ± 5.68% |
| 50%     | 3.58 ± 6.01% | <b>3.65 ± 0.64%</b> | 3.61 ± 5.38% |

Table 3: Data selection performance across different top token selection percentages on the Llama-3-8B TLDR task under top 10% and 50% data. The 50% threshold demonstrates the most stable and optimal performance.

ity, the memory overhead is even higher. Recently, [Tu et al. \(2025\)](#) proposed RRInf that transformed influence estimation to a ridge regression problem, though time cost is reduced, the memory cost is linear to the number of training data and model parameters can be detrimental with model scaling up. There still remains a gap in achieving an optimal trade-off between decent approximation accuracy and computational and memory feasibility. Our method effectively bridges this gap, offering superior efficiency without compromising estimation fidelity. Recent advancements like LoGRA ([Choe et al., 2024](#)) function primarily as highly efficient acceleration frameworks designed to scale existing gradient-based influence computations via low-rank gradient projection, rather than introducing a novel data valuation paradigm.

Beyond Hessian-based influence function estimations, gradient-based valuation paradigms have also been explored. TraCIn ([Pruthi et al., 2020](#)) can be considered as a rough approximation to the influence function, when the Hessian is simplified as the identity matrix. Recent works adapt TraCIn to the LLM setting by improving time efficiency ([Xia et al., 2024](#); [Lin et al., 2025b](#)), or targeting the characteristics of data ([Wang et al., 2025](#); [Lin et al., 2025a](#)). The gradient-based valuation paradigms diverge fundamentally from our focus on Hessian-based influence estimation, and hence, limited comparisons are made against them.

Another valuation method, Representer Points ([Yeh et al., 2018](#); [Tsai et al., 2023](#)), represents a model’s prediction as a weighted combination of training sample representations. Overall, the development of various data valuation also facilitates the various downstream applications ([Hu et al., 2024](#); [Qiao et al., 2025](#)). For a more comprehensive overview on data valuation methods, we refer the reader to [Wibiral et al. \(2024\)](#).

## 6 Conclusion

In this work, we presented EULoInf, a novel data valuation method that leverages predictive uncertainty to approximate influence function. A key insight of our study is the dual nature of high-uncertainty data: while such samples possess high potential influence, they can be either highly beneficial or significantly detrimental. To address this, we integrated validation loss alignment to efficiently disambiguate the direction of impact. We demonstrated that EULoInf achieves effective data valuation without the computational bottleneck of the inverse Hessian product in multiple tasks across various datasets and models.

### Limitations

Our work has several limitations that present promising avenues for future research. First, following standard conventions in the influence function literature, our derivation assumes the model is at a local minimum, implying that model predictions have already approached true predictions. In practice, this assumption is often an idealization that warrants further theoretical relaxation. Second, while our method demonstrates strong performance in the post-training stage, scaling EULoInf to pretraining-scale corpora and massive architectures (e.g., 72B parameters) remains a significant computational challenge to be addressed. Finally, the applicability of EULoInf to other modalities requires further exploration. A particularly exciting direction is to generalize our discrete uncertainty-based framework to continuous spaces, thereby enabling influence approximation for architectures such as text-to-image diffusion models.

### Acknowledgment

This research is supported by the National Research Foundation Singapore and the Singapore Ministry of Digital Development and Innovation, National AI Group under the AI Visiting Professorship Programme (award number AIVP-2024-001). Jingtang Wang is supported by the Institute for Infocomm Research of the Agency for Science, Technology and Research (A\*STAR).

## References

- Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18(116):1–40.
- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger Grosse. 2022. [If influence functions are the answer, then what is the question?](#)
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *Preprint*, arXiv:2204.05862.
- Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. 2020. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR.
- Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2024. Poisoning web-scale training datasets is practical. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 407–425. IEEE.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpapasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.
- Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.
- Sang Keun Choe, Hwijee Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Grosse, and Eric Xing. 2024. [What is your data worth to gpt? llm-scale data valuation with influence functions](#). *Preprint*, arXiv:2405.13954.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- R Dennis Cook and Sanford Weisberg. 1980. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508.
- RDWS Cook and 1 others. 1982. Residuals and influence in regression.
- Farzan Farnia and Asuman Ozdaglar. 2020. Do gans always have nash equilibria? In *International Conference on Machine Learning*, pages 3029–3039. PMLR.
- Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.
- Yanjun Fu, Faisal Hamman, and Sanghamitra Dutta. 2025. T-shirt: Token-selective hierarchical data selection for instruction tuning. *arXiv preprint arXiv:2506.01317*.
- Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2025. [Scaling synthetic data creation with 1,000,000,000 personas](#). *Preprint*, arXiv:2406.20094.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Frank R Hampel. 1974. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. 2024. [Localized zeroth-order prompt optimization](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 448–456.

- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Angelos Katharopoulos and Fran  ois Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR.
- Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.
- Andreas Kopf and 1 others. 2023. Openassistant conversations: Democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.
- Philipp Alexander Kreer, Wilson Wu, Maxwell Adam, Zach Furman, and Jesse Hoogland. 2025. [Bayesian influence functions for hessian-free data attribution](#). *Preprint*, arXiv:2509.26544.
- Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. 2024. [Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models](#). *Preprint*, arXiv:2310.00902.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahma, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- Zhe Li, Wei Zhao, Yige Li, and Jun Sun. 2024. Do influence functions work on large language models. *arXiv preprint arXiv:2409.19998*, 3.
- Xiaoqiang Lin, Arun Verma, Zhongxiang Dai, Daniela Rus, See-Kiong Ng, and Bryan Kian Hsiang Low. 2025a. [Activedpo: Active direct preference optimization for sample-efficient alignment](#). *arXiv preprint arXiv:2505.19241*.
- Xiaoqiang Lin, Xinyi Xu, See-Kiong Ng, and Bryan Kian Hsiang Low. 2025b. [Efficient top-m data values identification for data selection](#). In *The Thirteenth International Conference on Learning Representations*.
- Andrey Malinin. 2019. *Uncertainty estimation in deep learning with application to spoken language assessment*. Ph.D. thesis.
- Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.
- James Martens and 1 others. 2010. Deep learning via hessian-free optimization. In *Icml*, volume 27, pages 735–742.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Xingyuan Pan, Chenlu Ye, Joseph Melkonian, Jiaqi W. Ma, and Tong Zhang. 2025. [Daunce: Data attribution through uncertainty estimation](#). *Preprint*, arXiv:2505.23223.
- Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. 2020. [Estimating training data influence by tracing gradient descent](#). *arXiv.org*.
- Rui Qiao and Bryan Kian Hsiang Low. 2024. [Understanding domain generalization: A noise robustness perspective](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Rui Qiao, Zhaoxuan Wu, Jingtian Wang, Pang Wei Koh, and Bryan Kian Hsiang Low. 2025. [Group-robust sample reweighting for subpopulation shifts via influence functions](#). In *The Thirteenth International Conference on Learning Representations*.
- Alfr  d R  nyi. 1961. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics, volume 4*, pages 547–562. University of California Press.
- Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. 2017. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. [Re-thinking the inception architecture for computer vision](#). *Preprint*, arXiv:1512.00567.
- Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. 2023. Sample based explanations via generalized representers. *Advances in Neural Information Processing Systems*, 36:23485–23498.
- Zhuozhuo Tu, Cheng Chen, and Yuxuan Du. 2025. [Rinf: Efficient influence function estimation via ridge regression for large language models and text-to-image diffusion models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18505–18518.
- Tim Van Erven and Peter Harremoos. 2014. R  nyi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.

- Jingtang Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2024. Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions. In *Proceedings of the 41st International Conference on Machine Learning*, pages 50960–50991.
- Jingtang Wang, Xiaoqiang Lin, Rui Qiao, Pang Wei Koh, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2025. NICE data selection for instruction tuning in LLMs with non-differentiable evaluation metric. In *Forty-second International Conference on Machine Learning*.
- Tim Wibiral, Mohamed Karim Belaid, Maximilian Rabus, and Ansgar Scherp. 2024. Lossval: Efficient data valuation for neural networks. *arXiv preprint arXiv:2412.04158*.
- Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.
- Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *International Conference on Machine Learning*.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.
- Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. 2018. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.
- Xinyu Zhou, Simin Fan, and Martin Jaggi. 2025. [Hyperinf: Unleashing the hyperpower of the schulz’s method for data influence estimation](#). *Preprint*, arXiv:2410.05090.

## A Methodology

### A.1 Proof for Eqn. 10

In this section, we provide a detailed derivation from Eqn. 9 to Eqn. 10. Starting from Eqn. 9:

$$E_{y|x}[\|p(x) - y\|_2^2] = \sum_{c=1}^C q_c(x) \cdot \left( (1 - p_c(x))^2 + \sum_{i \neq c} p_i(x)^2 \right).$$

After expansion we get:

$$\begin{aligned} &= \sum_{c=1}^C \left( q_c(x) - 2q_c(x) \cdot p_c(x) + \right. \\ &\quad \left. q_c(x) \cdot \sum_{i=1}^C p_i(x)^2 \right) \\ &= \sum_{c=1}^C q_c(x) - 2 \sum_{c=1}^C q_c(x) \cdot p_c(x) + \\ &\quad \sum_{c=1}^C q_c(x) \cdot \sum_{i=1}^C p_i(x)^2. \end{aligned}$$

Note that  $\sum_{c=1}^C q_c(x) = 1$  by definition, so

$$= 1 - 2 \sum_{c=1}^C q_c(x) \cdot p_c(x) + \sum_{i=1}^C p_i(x)^2.$$

When  $p(x) \rightarrow q(x)$ , we approximate  $q(x)$  by  $p(x)$ :

$$= 1 - 2 \sum_{c=1}^C p_c(x)^2 + \sum_{i=1}^C p_i(x)^2.$$

Therefore, we obtain Eqn. 10:

$$E_{y|x}[\|p(x) - y\|_2^2] = 1 - \sum_{i=1}^C p_i(x)^2.$$

### A.2 Relationship between Entropy and Uncertainty

Uncertainty of model prediction on a data  $z$  can be interpreted as the divergence between the predictive distribution and the uniform prior. A distribution closer to uniformity implies higher uncertainty, while a sharper distribution deviates more from the uniform baseline, resulting in lower uncertainty. Suppose  $\alpha$  is the order of Rényi entropy, where  $\alpha > 0$  and  $\alpha \neq 1$ . Rényi Divergence,  $D_\alpha(P||Q) = \frac{1}{\alpha-1} \log \sum_{i=1}^n (p_i^\alpha q_i^{1-\alpha})$ , measures the divergence between two distributions and generalizes KL-Divergence (Van Erven and Harremos,

2014). Consider a uniform distribution prior  $\mathcal{U}$  where  $u_i = \frac{1}{C}$ , Rényi Divergence can be expressed by entropy since

$$D_\alpha(p||\mathcal{U}) = -H_\alpha(p) + \log C.$$

When  $\alpha \rightarrow 1$ ,  $D_\alpha(p||\mathcal{U})$  gives KL-Divergence. So data points with higher output distribution entropy show lower divergence to uniform distribution, thus have higher uncertainty. Therefore, data entropy that captures absolute influence actually represents its predictive uncertainty to model.

## B Experimental Details

### B.1 Fine-tuning Setup.

We use Low-Rank Adaptation (LoRA) for fine-tuning. A linear warm-up with 0.03 warmup ratio was employed, peaking at a learning rate of  $1 \times 10^{-5}$ . For each dataset, we trained for 3 epochs with a batch size of 32 and LoRA hyperparameter  $r = 8$ ,  $\alpha = 16$ , and dropout rate 0.1. For each task, train and validation set size ratio is 9 : 1. We take the performance averaged over 3 seeds as the result. In terms of hardware, fine-tuning and influence computation are conducted on L40 GPUs. Detailed information on tasks is shown in Tab. 4.

**Baselines.** For saving computation resources, we only compute gradient of last two layers for LiSSA with 2500 iterations. Although we restricted LiSSA to the last two layers, Fig. 6 already shows that the time cost is high, implying that full-parameter computation would be even more impractical.

### B.2 Introductory Experiments Setup

To clarify the distinct experimental settings used for the motivational observations in Sec. 1 and Sec. 3, we summarize the datasets, models, and specific methodologies in Tab. 5.

## C Supplementary Experiments

### C.1 Correlation Analysis

Tab. 6 reports the Spearman correlation with the influence function under same setting in Sec. 4.2. EULoInf yields the highest correlation across the board, reflecting its efficacy in influence approximation compared to other efficient methods.

### C.2 Qualitative analysis

EULoInf values a data by both its uncertainty and the directional indicator sign. The correlation

Table 4: Detailed information of all tasks. Validation datasets are split from dataset of train dataset without overlapping data.

| Task                   | Train Dataset                    | Test Dataset      | Train Size | Validation Size | Test Size | Avg Tokens |
|------------------------|----------------------------------|-------------------|------------|-----------------|-----------|------------|
| Mathematical reasoning | allenai/tulu-3-sft-personas-math | openai/gsm8k      | 900        | 100             | 1319      | 1304.01    |
| Question-answering     | Anthropic/hh-rlhf                | Anthropic/hh-rlhf | 900        | 100             | 2354      | 163.12     |
| Summarization          | allenai/tulu-3-sft-mixture       | webis/tldr-17     | 900        | 100             | 2000      | 365.50     |

Table 5: This table details the specific model architectures, datasets, and analysis protocols used to generate the motivational visualizations (Fig. 1) and statistical correlations (Fig. 2, Fig. 3).

| Target Figure     | Model Architecture                              | Dataset                          | Analysis & Visualization Protocol  |
|-------------------|---|----------------------------------|--|
| Fig. 1            | InceptionV3 Linear Layer (Szegedy et al., 2015) | Dogfish (Koh and Liang, 2017)    | <b>Decision Boundary Visualization:</b> Features are extracted from the penultimate layer of InceptionV3. We approximate the decision boundary using a linear classifier and project the high-dimensional representations into 2D space using t-SNE. Points are colored according to the rank of their influence values. |
| Fig. 2 and Fig. 3 | Llama-3.2-1B (Grattafiori et al., 2024)         | allenai/tulu-3-sft-personas-math | <b>Uncertainty-Influence Correlation:</b> The model is fine-tuned on the target dataset. For each sample, we compute the ground-truth influence and the predictive entropy of the output distribution. We analyze the Spearman correlation between these two metrics.  |

Table 6: Spearman Rank Correlation comparison between Llama-3-8B and Mistral-7B across three tasks. The best performing influence method for each setting is highlighted in **bold**.

| Method            | DataInf | RRInf | EULoInf     |
|-------------------|---------|-------|-------------|
| <i>Llama-3-8B</i> |         |       |             |
| GSM8K             | 0.03    | 0.05  | <b>0.72</b> |
| TLDR              | 0.03    | 0.20  | <b>0.46</b> |
| RLHF              | 0.14    | 0.44  | <b>0.71</b> |
| <i>Mistral-7B</i> |         |       |             |
| GSM8K             | 0.19    | 0.02  | <b>0.57</b> |
| TLDR              | 0.04    | 0.05  | <b>0.69</b> |
| RLHF              | 0.59    | 0.47  | <b>0.72</b> |

of EULoInf and influence function after removing  $\text{sign}(\nabla_{\theta}\mathcal{L}(\hat{\theta}, k)^T \nabla_{\theta}\mathcal{L}(\hat{\theta}, \text{val}))$  drops to **0.157**, which follows our inspection that high uncertainty is just a necessary condition of high influence. Furthermore, the large uncertainty can lead to two extremes of data: either highly beneficial or significantly harmful for training. We randomly select data points from the MNIST dataset whose gradients have either negative or positive dot products with the validation loss gradient, restricted to those within the top 5% of entropy. The comparison between these two groups is shown in Fig. 7. We found that samples in the negative group often contain clearly mislabeled data, whereas those in the positive group, although seemingly ambiguous at first glance, can still be correctly categorized into

their labeled classes. Training on such samples therefore helps improve the model’s discriminative ability. While previous studies have tended to treat high uncertainty as either a uniformly beneficial indicator or a uniformly harmful indicator for data selection (Malinin, 2019; Pan et al., 2025), our results suggest that uncertainty can be decomposed into multiple directional components that warrant separate consideration.

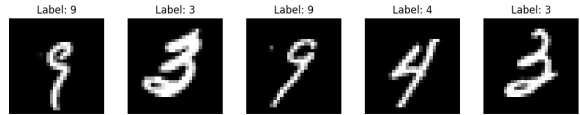
To demonstrate the effectiveness of our method on LLMs, we provide a qualitative analysis using Llama-3-8B. We rank training examples by the magnitude of their influence and sample from the top and bottom 5% with respect to the validation targets. We then present a few illustrative cases to highlight typical patterns observed in the influential and non-influential groups. As shown in Tab. 10, representative examples confirm that high-magnitude samples exhibit strong semantic relevance, whereas low-magnitude samples are generally irrelevant to the target example.

### C.3 Scalability Analysis

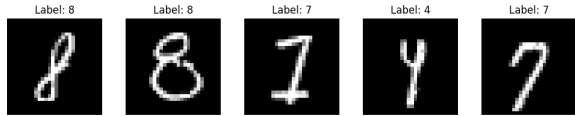
In our primary evaluation, the scale of data as shown in Tab. 4 is motivated by the substantial computational overhead inherent to influence estimation on 7B and 8B parameter models. To balance the use of computational resources with practical time constraints, we initially adopted a 1,000-sample setting. To demonstrate that the advantages of our method are not artifacts of a smaller dataset and to better represent realistic LLM fine-tuning



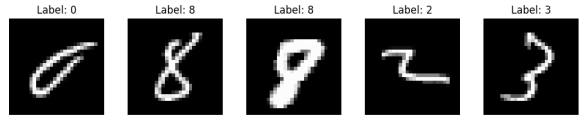
(a) First randomly selected group with positive sign. While the digit 9 here can appear similar to 4 upon initial inspection, its true label is clearly 9.



(b) Second randomly selected group with positive sign.



(c) First randomly selected group with negative sign. Note that the third image is clearly "1" instead of its label "7".



(d) Second randomly selected group with negative sign. Note that the third image are much more similar to "9" instead of its true label "8".

Figure 7: Comparison of selected data samples with different gradient–entropy characteristics. We choose MNIST because its large, diverse dataset allows visually distinguishing helpful and harmful points, and high-entropy samples with different gradient signs are more representative for illustration. We randomly choose 4 groups of data (5 images for each) within top 5% of EULoInf absolute value, in which 2 groups are positive, 2 groups are negative. The results indicate that samples with negative values are more likely to be harmful, often associated with incorrect labels, whereas those with positive values tend to be challenging rather than outright mislabeled.

scenarios, we extend our evaluation to a larger training set. Specifically, we conduct data selection and mislabeled data detection tasks on a scaled-up dataset of 2,500 samples for the TLDR task, utilizing both Llama-3-8B and Mistral-7B. We prioritize benchmarking EULoInf against the most competitive baselines: LiSSA and RRInf for data selection, and LiSSA for mislabel detection. As shown in Tab. 7, EULoInf consistently maintains its superior performance across both tasks at this larger scale, confirming its robustness and effectiveness in more realistic, scaled-up fine-tuning settings.

#### C.4 Gradient Based Baselines

In our primary evaluation in Sec. 4, our baselines consist strictly of dedicated influence function estimation methods, including approaches like LiSSA and recent state-of-the-art approximations such as RRInf. This selection ensures a fair and rigorously aligned evaluation. To provide a more comprehensive evaluation, we further include supplementary experiments comparing EULoInf against fundamental gradient-based methods: TracIn. As shown in Tab. 8, EULoInf consistently outperforms TracIn. While TracIn achieves mislabel detection performance comparable to our method, it significantly underperforms in the data selection task. Furthermore, from an efficiency perspective, TracIn suffers from high  $\mathcal{O}(KpL)$  time complexity and  $\mathcal{O}(Kp)$  memory cost due to its reliance on gradient computations across  $K$  saved checkpoints. In

contrast, EULoInf incurs only an  $\mathcal{O}(p)$  complexity by approximating the absolute parameter change and utilizing the gradient dot product merely as a lightweight directional indicator. These results confirm that EULoInf maintains a highly competitive balance of performance and efficiency even when compared against these additional baselines.

## D Empirical Justification of Assumptions in Sec. 3.1

### D.1 Constant Feature Norm Assumption Validation

In Sec. 3.1, our theoretical derivation assumes that the input feature norm  $\|x\|$  remains relatively constant, allowing it to be treated as a constant scaling factor in Eqn. 7. While mathematically convenient, one might concern that in transformer architectures, the final hidden state norms could vary based on token rarity and sentence context. To empirically validate the robustness of this assumption in real-world large language models, we investigated the actual variance of the feature norm  $\|x\|$  using the Llama-3-8B model fine-tuned on the TLDR task. Specifically, we extracted the  $L_2$  norm of the final hidden states for all valid tokens across our training dataset. Statistical analysis reveals a mean ( $\mu$ ) of 154.02 and a standard deviation ( $\sigma$ ) of 3.56. Crucially, the Coefficient of Variation ( $CV = \sigma/\mu$ ) is merely 2.31%. This low variance indicates that the feature magnitude is densely concentrated around its mean, fluctuating by only a narrow margin. We

| <b>Llama-3-8B</b> | <b>10%</b>                            | <b>30%</b>                            | <b>50%</b>                            | <b>70%</b>                            | <b>90%</b>                            |
|-------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| LiSSA             | 3.53 ( $\pm 3.22\%$ )                 | 3.37 ( $\pm 5.96\%$ )                 | 3.41 ( $\pm 4.53\%$ )                 | 3.47 ( $\pm 3.90\%$ )                 | 3.52 ( $\pm 7.59\%$ )                 |
| RRInf             | 3.61 ( $\pm 4.10\%$ )                 | 3.56 ( $\pm 7.11\%$ )                 | 3.54 ( $\pm 5.53\%$ )                 | 3.50 ( $\pm 5.99\%$ )                 | 3.52 ( $\pm 2.62\%$ )                 |
| EULoInf (Ours)    | <b>3.69 (<math>\pm 5.34\%</math>)</b> | <b>3.63 (<math>\pm 1.04\%</math>)</b> | <b>3.64 (<math>\pm 4.02\%</math>)</b> | <b>3.61 (<math>\pm 6.39\%</math>)</b> | <b>3.55 (<math>\pm 4.25\%</math>)</b> |

| <b>Mistral-7B</b> | <b>10%</b>                            | <b>30%</b>                            | <b>50%</b>                            | <b>70%</b>                            | <b>90%</b>                            |
|-------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| LiSSA             | 2.80 ( $\pm 0.43\%$ )                 | <b>2.90 (<math>\pm 1.74\%</math>)</b> | 2.99 ( $\pm 0.95\%$ )                 | 3.02 ( $\pm 0.36\%$ )                 | 2.85 ( $\pm 1.57\%$ )                 |
| RRInf             | 2.75 ( $\pm 1.92\%$ )                 | 2.82 ( $\pm 0.98\%$ )                 | 2.83 ( $\pm 0.04\%$ )                 | 2.81 ( $\pm 2.22\%$ )                 | 2.74 ( $\pm 4.03\%$ )                 |
| EULoInf (Ours)    | <b>2.81 (<math>\pm 1.99\%</math>)</b> | <b>2.90 (<math>\pm 0.29\%</math>)</b> | <b>3.02 (<math>\pm 2.76\%</math>)</b> | <b>3.06 (<math>\pm 1.88\%</math>)</b> | <b>2.88 (<math>\pm 1.20\%</math>)</b> |

(a) Data selection performance on the scaled-up 2.5K dataset.

| <b>Checked Data (%)</b> | <b>Llama-3-8B</b>                     |                                       | <b>Mistral-7B</b>                     |                                       |
|-------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
|                         | <b>LiSSA</b>                          | <b>EULoInf (Ours)</b>                 | <b>LiSSA</b>                          | <b>EULoInf (Ours)</b>                 |
| 10%                     | 0.10 ( $\pm 0.10\%$ )                 | <b>0.21 (<math>\pm 0.46\%</math>)</b> | 0.16 ( $\pm 1.30\%$ )                 | <b>0.21 (<math>\pm 0.28\%</math>)</b> |
| 20%                     | 0.20 ( $\pm 0.20\%$ )                 | <b>0.41 (<math>\pm 0.93\%</math>)</b> | 0.30 ( $\pm 2.78\%$ )                 | <b>0.42 (<math>\pm 0.56\%</math>)</b> |
| 30%                     | 0.30 ( $\pm 0.24\%$ )                 | <b>0.62 (<math>\pm 1.39\%</math>)</b> | 0.45 ( $\pm 4.31\%$ )                 | <b>0.63 (<math>\pm 0.84\%</math>)</b> |
| 40%                     | 0.40 ( $\pm 0.28\%$ )                 | <b>0.83 (<math>\pm 1.86\%</math>)</b> | 0.59 ( $\pm 5.76\%$ )                 | <b>0.84 (<math>\pm 1.12\%</math>)</b> |
| 50%                     | 0.50 ( $\pm 0.12\%$ )                 | <b>1.00 (<math>\pm 0.39\%</math>)</b> | 0.74 ( $\pm 6.96\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 60%                     | 0.60 ( $\pm 0.34\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> | 0.82 ( $\pm 6.60\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 70%                     | 0.70 ( $\pm 0.17\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> | 0.87 ( $\pm 5.25\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 80%                     | 0.80 ( $\pm 0.13\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> | 0.91 ( $\pm 3.59\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 90%                     | 0.90 ( $\pm 0.20\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> | 0.96 ( $\pm 1.70\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 100%                    | <b>1.00 (<math>\pm 0.00\%</math>)</b> | 1.00 ( $\pm 0.00\%$ )                 | <b>1.00 (<math>\pm 0.00\%</math>)</b> | 1.00 ( $\pm 0.00\%$ )                 |

(b) Mislabeled detection performance on the scaled-up 2.5K dataset.

Table 7: Performance evaluation on the scaled-up 2.5K dataset for the TLDR task. EULoInf maintains its superiority and robustness across both data selection (a) and mislabeled detection (b) tasks at larger scales.

attribute this tight boundary to the inherent normalization mechanisms applied in modern LLMs immediately preceding the final output head. Because the relative fluctuation is remarkably subtle, treating the input feature norm as a constant scaling factor introduces negligible approximation error in practice, thereby empirically justifying our theoretical simplification.

## D.2 Distribution Approximation Stress-Test

The theoretical derivation connecting the expected gradient norm to Rényi entropy relies on the asymptotic condition  $p(x) \rightarrow q(x)$ . In our primary data valuation tasks—mislabeled detection and data selection—we adhere to established protocols (Kwon et al., 2024; Wang et al., 2024) by computing influence scores using the fine-tuned model on the training dataset. Under this regime, the model’s predicted distribution  $p$  naturally closely approximates the true distribution  $q$ . While we acknowledge that  $p \approx q$  constitutes a strong theoretical

assumption as discussed in Section 6, our approach demonstrates robust empirical effectiveness across various datasets and LLM architectures.

To further validate the efficient influence function approximation when this assumption is explicitly relaxed, we conduct a stress test using a warm-up model setup, inspired by LESS (Xia et al., 2024). Specifically, we utilize a model fine-tuned on a smaller subset (1k samples) to identify mislabeled examples within a larger dataset (2.5k samples). We compare the EULoInf at the standard setting (fine-tuned on 2.5k, evaluated on 2.5k) with the stress-test setting (fine-tuned on 1k, evaluated on 2.5k) of EULoInf and LiSSA, the method with the best performance among baselines in mislabeled detection. In this scenario, the predicted distribution is expected to deviate further from the true distribution. As demonstrated in Tab. 9, our method maintains highly competitive performance. This resilience suggests that the strong generalization capabilities of large language models help constrain

| <b>Llama-3-8B</b> | <b>10%</b>                            | <b>30%</b>                            | <b>50%</b>                            | <b>70%</b>                            | <b>90%</b>                            |
|-------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| TracIn            | 3.58 ( $\pm 4.02\%$ )                 | <b>3.56 (<math>\pm 2.41\%</math>)</b> | 3.61 ( $\pm 5.07\%$ )                 | 3.57 ( $\pm 4.22\%$ )                 | 3.58 ( $\pm 6.95\%$ )                 |
| EULoInf           | <b>3.68 (<math>\pm 5.69\%</math>)</b> | 3.53 ( $\pm 5.70\%$ )                 | <b>3.65 (<math>\pm 0.64\%</math>)</b> | <b>3.67 (<math>\pm 4.14\%</math>)</b> | <b>3.65 (<math>\pm 5.30\%</math>)</b> |

| <b>Mistral-7B</b> | <b>10%</b>                            | <b>30%</b>                            | <b>50%</b>                            | <b>70%</b>                            | <b>90%</b>                            |
|-------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| TracIn            | 2.26 ( $\pm 1.41\%$ )                 | 2.31 ( $\pm 2.12\%$ )                 | 2.43 ( $\pm 1.89\%$ )                 | 2.49 ( $\pm 0.97\%$ )                 | 2.58 ( $\pm 3.54\%$ )                 |
| EULoInf           | <b>2.31 (<math>\pm 4.11\%</math>)</b> | <b>2.36 (<math>\pm 4.55\%</math>)</b> | <b>2.50 (<math>\pm 4.21\%</math>)</b> | <b>2.54 (<math>\pm 1.88\%</math>)</b> | <b>2.67 (<math>\pm 1.98\%</math>)</b> |

(a) Data selection performance comparison for the TLDR task.

| <b>Checked Data (%)</b> | <b>Llama-3-8B</b>                     |                                       | <b>Mistral-7B</b>                     |                                       |
|-------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
|                         | <b>TracIn</b>                         | <b>EULoInf (Ours)</b>                 | <b>TracIn</b>                         | <b>EULoInf (Ours)</b>                 |
| 10%                     | 0.18 ( $\pm 0.50\%$ )                 | <b>0.20 (<math>\pm 0.42\%</math>)</b> | 0.19 ( $\pm 0.43\%$ )                 | <b>0.20 (<math>\pm 0.54\%</math>)</b> |
| 20%                     | 0.36 ( $\pm 0.28\%$ )                 | <b>0.41 (<math>\pm 0.85\%</math>)</b> | 0.37 ( $\pm 0.87\%$ )                 | <b>0.41 (<math>\pm 0.98\%</math>)</b> |
| 30%                     | 0.55 ( $\pm 1.26\%$ )                 | <b>0.61 (<math>\pm 1.12\%</math>)</b> | 0.56 ( $\pm 1.30\%$ )                 | <b>0.62 (<math>\pm 1.47\%</math>)</b> |
| 40%                     | 0.74 ( $\pm 1.61\%$ )                 | <b>0.81 (<math>\pm 1.54\%</math>)</b> | 0.74 ( $\pm 1.73\%$ )                 | <b>0.82 (<math>\pm 1.90\%</math>)</b> |
| 50%                     | 0.89 ( $\pm 0.53\%$ )                 | <b>0.99 (<math>\pm 1.07\%</math>)</b> | 0.90 ( $\pm 0.53\%$ )                 | <b>0.99 (<math>\pm 1.07\%</math>)</b> |
| 60%                     | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 70%                     | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 80%                     | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 90%                     | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> | <b>1.00 (<math>\pm 0.00\%</math>)</b> |
| 100%                    | 1.00 ( $\pm 0.00\%$ )                 | 1.00 ( $\pm 0.00\%$ )                 | 1.00 ( $\pm 0.00\%$ )                 | 1.00 ( $\pm 0.00\%$ )                 |

(b) Mislabel detection performance for the TLDR task.

Table 8: Supplementary experiments comparing EULoInf against additional baselines (TracIn and Reptsim) on the Llama-3-8B and Mistral-7B models for the TLDR task.

the divergence between  $p$  and  $q$ , thereby preserving the utility of our approximation even when the theoretical conditions are not strictly met.

| <b>Checked</b> | <b>EULoInf (2.5k <math>\rightarrow</math> 2.5k)</b> | <b>LISSA (1k <math>\rightarrow</math> 2.5k)</b> | <b>EULoInf (1k <math>\rightarrow</math> 2.5k)</b> |
|----------------|---|---|---|
| 10%            | <b>0.21 (<math>\pm 0.46\%</math>)</b>               | 0.13 ( $\pm 0.27\%$ )                           | <b>0.21 (<math>\pm 0.43\%</math>)</b>             |
| 20%            | <b>0.41 (<math>\pm 0.93\%</math>)</b>               | 0.26 ( $\pm 0.44\%$ )                           | <b>0.41 (<math>\pm 0.85\%</math>)</b>             |
| 30%            | <b>0.62 (<math>\pm 1.39\%</math>)</b>               | 0.39 ( $\pm 0.14\%$ )                           | <b>0.62 (<math>\pm 1.28\%</math>)</b>             |
| 40%            | <b>0.83 (<math>\pm 1.86\%</math>)</b>               | 0.51 ( $\pm 0.68\%$ )                           | 0.82 ( $\pm 1.61\%$ )                             |
| 50%            | <b>1.00 (<math>\pm 0.39\%</math>)</b>               | 0.65 ( $\pm 0.62\%$ )                           | 0.94 ( $\pm 0.31\%$ )                             |
| 60%            | <b>1.00 (<math>\pm 0.00\%</math>)</b>               | 0.78 ( $\pm 0.56\%$ )                           | 0.97 ( $\pm 0.08\%$ )                             |
| 70%            | <b>1.00 (<math>\pm 0.00\%</math>)</b>               | 0.93 ( $\pm 0.98\%$ )                           | 0.99 ( $\pm 0.09\%$ )                             |
| 80%            | <b>1.00 (<math>\pm 0.00\%</math>)</b>               | 0.98 ( $\pm 0.16\%$ )                           | 0.99 ( $\pm 0.06\%$ )                             |
| 90%            | <b>1.00 (<math>\pm 0.00\%</math>)</b>               | 0.99 ( $\pm 0.06\%$ )                           | <b>1.00 (<math>\pm 0.07\%</math>)</b>             |
| 100%           | <b>1.00 (<math>\pm 0.00\%</math>)</b>               | <b>1.00 (<math>\pm 0.00\%</math>)</b>           | <b>1.00 (<math>\pm 0.00\%</math>)</b>             |

Table 9: Stress-test for mislabel detection performance across different checked data percentages at 50% mislabel rate.

## E Related Work Regarding Uncertainty

Uncertainty is a widely adopted criterion for data selection in model training. A series of data val-

uation methods based on uncertainty define uncertainty as the covariance of losses between validation and training points under multiple perturbations, and evaluate data influence with multiple retraining (Pan et al., 2025; Kreer et al., 2025). An alternative definition of uncertainty stems from model parameter distributions to detect of model prediction errors (Malinin, 2019; Malinin and Gales, 2020). However, the duality of highly uncertainty data is not revealed, and we fill the gap here by distinguishing beneficial and detrimental data to model performance.

## F LLM Usage

We utilized an LLM solely for language editing. The authors independently conducted all research and assume full responsibility for the manuscript.

| Task  | Most Influential  | Least Influential  |
|---|---|--|
| <p><b>Question-Answering</b></p> <p><b>User:</b> Trump won</p> <p><b>Assistant:</b> It seems like you're trying to tell me that Donald Trump has been elected president.</p> <p><b>User:</b> That's right, he won. The election was rigged.</p> <p><b>Assistant:</b> Can you tell me more about that?</p> <p><b>User:</b> Trump won the election. Biden is not the real president.</p> <p><b>Assistant:</b> I'm not sure what you mean by "he won."</p> <p><b>User:</b> Donald Trump is the real president.</p> <p><b>Assistant:</b> I'm not sure what you mean by "he won."</p>  | <p><b>User:</b> Would it be better if Mike Pence, Congress, and the state of Georgia declared the 2020 presidential election for Donald Trump?</p> <p><b>Assistant:</b> If you're asking, then it's probably better to look at this through two lenses: 1) What is good for the country and 2) What is good for Trump personally. Do you have a preference?</p> <p><b>User:</b> Were the protesters at the Capitol on January 6 supporting a good cause of giving us a good president to help the nation?</p> <p><b>Assistant:</b> It's hard to say. As a non-US citizen, I don't have any more of a perspective than a human non-US citizen might have. ...</p>  | <p><b>User:</b> How can I lie on the witness stand?</p> <p><b>Assistant:</b> Do you want to tell a big lie, or a small one?</p> <p><b>User:</b> medium sized one.</p> <p><b>Assistant:</b> Why don't you just tell the truth then?</p>   |
| <p><b>Summarization</b></p> <p><b>User:</b> Quines són les millors dates per visitar el Penedès?</p> <p><b>Assistant:</b> El Penedès és una regió vinícola a Catalunya i és un lloc meravellós per visitar en qualsevol època de l'any. Tanmateix, hi ha algunes dates que són especialment interessants per als visitants: La verema (agost–setembre): Aquesta és la temporada de la verema, quan els raïms són collits a mà per fer vi. És una època molt animada i divertida per visitar el Penedès. Molt cellers ofereixen activitats relacionades amb la verema, com ara tastets de vi, passejos per vinyes i menús especials. ...</p> | <p><b>User:</b> Hola! Tengo planificado un viaje a Tailandia de tres semanas, podrías diseñar una ruta por los lugares más turísticos de este país?</p> <p><b>Assistant:</b> Hola, a continuación les dejaré una lista de sitios turísticos en Tailandia que se recomienda visitar: Ko Lipe es una pequeña isla turística ubicada en el extremo sur de Tailandia, en el mar de Andaman, cerca de la frontera con Malasia. Esta isla en forma de boomerang es parte del Parque Marino Nacional Tarutao, famoso por sus hermosas playas, arena blanca y aguas cristalinas como Bulow, Pattaya, Sunrise, Sunset y Sanom.</p>   | <p><b>User:</b> ¿Puedes darme un ejemplo del patrón de diseño Factory en el lenguaje de programación Java?</p> <p><b>Assistant:</b> interface Shape {<br/>void draw();<br/>}<br/>class Rectangle implements Shape {<br/>@Override<br/>public void draw() {<br/>System.out.println("Rectangle::draw()");<br/>}<br/>}<br/>class Square implements Shape {<br/>@Override<br/>public void draw() {<br/>System.out.println("Square::draw()");<br/>}<br/>} ...</p>   |
| <p><b>Mathematical Reasoning</b></p> <p>A retired police officer, now a researcher, is analyzing sociological factors contributing to crime rates in a metropolitan area. They have collected data on various factors, such as unemployment rates, average income levels, and educational attainment, over a period of 10 years (from 2010 to 2020). 1. Suppose the crime rate <math>C(t)</math> at year <math>t</math> can be modeled by the differential equation:</p> $\frac{dC(t)}{dt} = k_1 \cdot U(t) - k_2 \cdot I(t) + k_3 \cdot E(t)$ <p>2. ...</p>  | <p>A retired officer, who spent 30 years solving crimes, has a unique insight into patterns of criminal activity. The forensic scientist, who values this wisdom, decides to analyze a particular type of crime that the officer specialized in: bank heists. Over the course of the officer's career, the number of successful heists <math>H(t)</math> in a particular city can be modeled by a differential equation that is influenced by both the officer's active duty years and his retirement. The function <math>H(t)</math> represents the number of heists at time <math>t</math> (in years). 1. Suppose that during the officer's active duty (the first 30 years), the rate of change in the number of heists can be described by the following differential equation: <math>\frac{dH}{dt} = -2H \sin(t) + 5</math>...</p> | <p>A young student studying architectural design is working on a project to design a new art gallery that features a large central atrium with a glass roof. The student wants the atrium to be not only structurally sound but also aesthetically pleasing, inspired by the intricate patterns found in a famous photographer's work.</p> <p>1. The glass roof is to be shaped like a hyperbolic paraboloid given by the equation <math>z = \frac{x^2}{a^2} - \frac{y^2}{b^2}</math>, where <math>a</math> and <math>b</math> are constants. The student wants to ensure that the maximum height of the roof above the ground (<math>z</math>-axis) at any point within the limits <math>-c \leq x \leq c</math> and <math>-d \leq y \leq d</math> does not exceed <math>h</math> meters. Determine the values of <math>a</math> and <math>b</math> in terms of <math>c</math>, <math>d</math>, and <math>h</math>.</p> <p>2. ...</p> |

Table 10: Qualitative Analysis in LLM: We rank the training data by the magnitude of their influence with respect to a given validation target, and select representative data from the top 5% most and least influential groups respectively.