
Neural Networks for Learnable and Scalable Influence Estimation of Instruction Fine-Tuning Data

Ishika Agarwal & Dilek Hakkani-Tür
Department of Computer Science
UIUC
Urbana-Champaign, IL 61801, USA
{ishikaa2, dilek}@illinois.edu

Abstract

Influence functions provide crucial insights into model training, but existing methods suffer from large computational costs and limited generalization. Particularly, recent works have proposed various metrics and algorithms to calculate the influence of data using language models, which do not scale well with large models and datasets. This is because of the expensive forward and backward passes required for computation, substantial memory requirements to store large models, and poor generalization of influence estimates to new data. In this paper, we explore the use of small neural networks – which we refer to as the InfluenceNetwork – to estimate influence values, achieving up to 99% cost reduction. Our evaluation demonstrates that influence values can be estimated with models just 0.0007% the size of full language models (we average across 1.5B-22B versions). We apply our algorithm of estimating influence values (called **NN-CIFT**: Neural Networks for effiCient Instruction Fine-Tuning) to the downstream task of subset selection for general instruction fine-tuning. In our study, we include four state-of-the-art influence functions and show no compromise in performance, despite large speedups, between NN-CIFT and the original influence functions. We provide an in-depth hyperparameter analyses of NN-CIFT. The code for our method can be found here: <https://github.com/agarwalishika/NN-CIFT>.

1 Introduction

The strong instruction-following abilities of large language models (LLMs) can be attributed to instruction fine-tuning (IFT) [Zhang et al., 2024]. IFT builds on top of current language modeling capabilities and strengthens the instruction following abilities of models. Recent works have taken **data efficient** approaches for IFT. The goal is to select a small subset of samples on which to fine-tune a model Agarwal et al. [2025], Mirzasoileiman et al. [2020], Das and Khetan [2024], Xia et al. [2024], Renduchintala et al. [2024], Liu et al. [2024c] that emulates the full dataset.

Data efficient pipelines typically consist of two stages: (1) *data valuation*: designing functions to estimate the influence of data points, and (2) *data selection*: using influence estimates to choose a balanced set of influential data. Usually, data selection is cheaper than valuation – for instance, DELIFT (SE)¹ [Agarwal et al., 2025] computes the similarity of sentence embeddings between pairs of data (expensive) for valuation and selects representative data using a submodular function (cheap).

Formally, influence functions estimate the value of data. For instance, brute force influence functions use leave-one-out (LOO) training to measure impact by omitting each data point and evaluating performance Scanlon [1982]. More recent influence functions use LLMs to estimate influence. Table

¹Short for "Sentence Embedding".

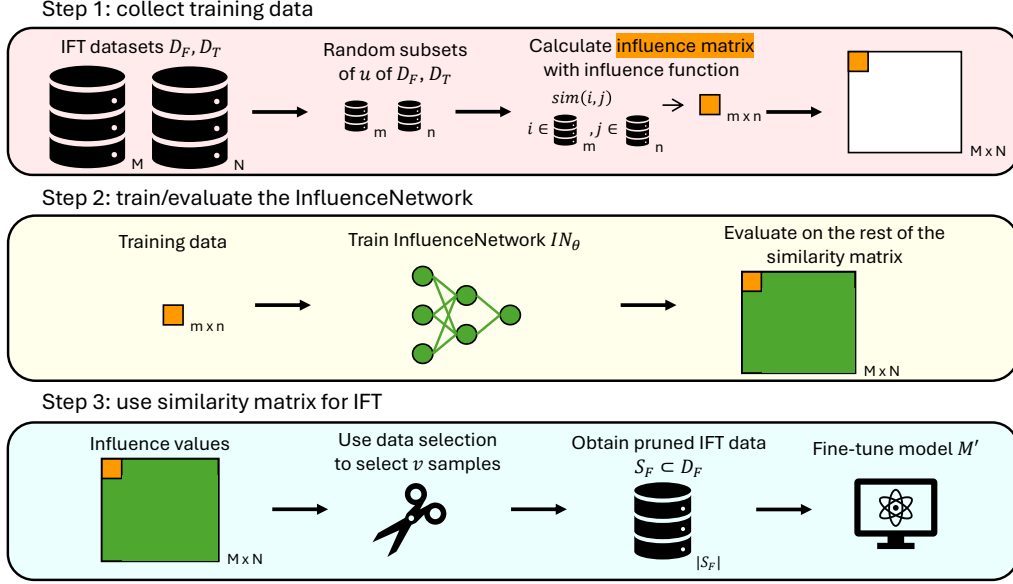


Figure 1: Overview of NN-CIFT. The first step consists of using established influence functions to **collect data** for training the InfluenceNetwork. Next, the data from Step (1) is used to **train the InfluenceNetwork** and, subsequently, **estimate the influence values** for the rest of the data. Finally, the data selection algorithm corresponding to the original influence function is used to **select a subset of IFT data** to fine-tune a model on.

1 outlines the expenses of state-of-the-art (SOTA) influence functions, which comes from the large amount of forward and backward passes through highly parameterized models.

In this paper, we introduce **NN-CIFT**: Neural Networks for effiCient Instruction Fine-Tuning and explore how to train influence functions efficiently. We improve efficiency by using compact neural networks – which we coin as the InfluenceNetwork – that are 0.0077% the size of LLMs, to estimate influence. Figure 1 outlines our methodology with a pairwise influence function (more details about pairwise influence functions in Appendix C.1).

As depicted, NN-CIFT is a three-step algorithm. The neural network must be trained to estimate influence values effectively. Hence, we first use the influence function (with LLMs) to output influence values for **a very small subset of data**. This becomes our training data for the InfluenceNetwork. We find that a small neural network can sufficiently learn to estimate influence with very few data (covered in Section 4).

Second, we train the InfluenceNetwork, and use it to estimate the influence values **for the rest of the data points**. Finally, we apply a data selection algorithm on the influence values. This helps to obtain a small subset of IFT data to enhance language models. After fine-tuning language models on the chosen subsets, we find that NN-CIFT achieves comparable performance to the original influence functions (covered in Section 5).

Method	Cost	Size
Pairwise		
DELIFT [Agarwal et al., 2025]	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) [Agarwal et al., 2025]	$\mathcal{O}(MN) \cdot F$	355M
LESS [Xia et al., 2024]	$\mathcal{O}(M + N) \cdot B$	7-8B
NN-CIFT (ours)	$\mathcal{O}(MN) \cdot F$	205K
Pointwise		
SelectIT [Liu et al., 2024a]	$\mathcal{O}(M) \cdot F$	7-8B
NN-CIFT (ours)	$\mathcal{O}(M) \cdot F$	205K

Table 1: Approximating the computational complexity of data valuation in terms of the cost of forward passes (F) or backward passes (B) through a model. $M = |\mathcal{D}_F|$ and $N = |\mathcal{D}_T|$, a fine-tuning and target dataset respectively, we use for subset selection. See Appendix C.1 for more details. Size denotes the number of parameters of the corresponding model. Note: larger models have a higher F and B .

Our contributions and findings are listed as follows. NN-CIFT:

1. **alleviates the cost of using expensive LLMs during data valuation** by using smaller and cheaper neural networks, without affecting the performance on downstream tasks (Tables 4-6);
2. **achieves competitive performance to previous data valuation methods, despite using only 0.25%-5% of the data.** The average mean square error in influence values between NN-CIFT and the original influence functions is merely 0.067 (Figure 2);
3. is shown to be effective for new data points, **circumventing the need to retrain an influence function for new data** – previous works incur this cost (Figure 2).
4. **reduces costs by 77-99% time** during data valuation (Table 7).

Section 2 outlines the current state of research in data valuation and data selection. Section 3 explains the problem setting. Section 4 presents the main methodology for NN-CIFT and motivating results. Finally, Section 5 reports results on the downstream task of subset selection after the data valuation stage. In our evaluation, we find that using a small LLM with the original influence functions results in degraded performance. Our hyperparameter studies are in Appendix A.1, Figure 4 and Appendix A.2, Figure 8. We also show language model performance with smaller subsets of selected fine-tuning data in Appendix B. Lastly, the SOTA influence functions are detailed in Appendix C.

2 Related Works

2.1 Data Valuation

Wei et al. [2023] hint that different models extract different information from the same data. Hence, effective fine-tuning requires datasets to be specific to each model. Not all data points affect the model equally - models learn more from certain data points than others. Therefore, data valuation methods prune out such low-influence data for efficient fine-tuning [Xia et al., 2024, Agarwal et al., 2025]. Current research is divided into model-independent and model-dependent valuation metrics.

Model-independent methods, such as distance or clustering-based methods [Das and Khetan, 2024, Liu et al., 2024c, Renduchintala et al., 2024] are faster and less computationally expensive. Distance-based methods assign more "influence" to data points that are further from each other, optimizing for a diverse subset. Clustering-based methods assign more "influence" to data points that are representative (i.e., the centroids of clusters).

On the other hand, model-dependent methods – such as inference-based and gradient-based – are more resource intensive. Inference-based methods [Liu et al., 2024a, Agarwal et al., 2025] use model inference signals (e.g., token distributions) to evaluate the performance or confidence of models, and value data based on how performative/confident they are. Gradient based methods [Xia et al., 2024, Mirzasoleiman et al., 2020, Killamsetty et al., 2021, Koh and Liang, 2020], on the other hand, can assign higher influence to data points with (1) higher magnitudes of gradients, or (2) gradients that match domain-specific data (for domain-specific fine-tuning, for example).

While they are expensive to calculate, when paired with data selection algorithms, model-dependent data valuation metrics can be used to select subsets of data that are specific to a model’s capabilities. Model-dependent data valuation metrics help to select data that will maximize a certain objective for each model, rendering fine-tuning more effective.

2.2 Data Selection

Data selection aims to prune redundant and noisy data samples from large datasets to produce a small, information-rich subset [Agarwal et al., 2025, Xia et al., 2024]. This subset should be representative of the larger dataset while performing comparably, if not better, than using the full dataset. Data selection methods usually have objectives for selecting data: (1) instruction tuning [Liu et al., 2024a], (2) task-specific fine-tuning [Liu et al., 2024c], (3) continual learning [Agarwal et al., 2025], (4) preference alignment [Liu et al., 2024b], etc. While certain objectives are subsets of others (e.g. (2) is subset of (1)), the data selected for each purpose may not necessarily overlap. For instance, (1) requires data that is representative of a particular dataset, whereas (2) focuses on samples that reflect

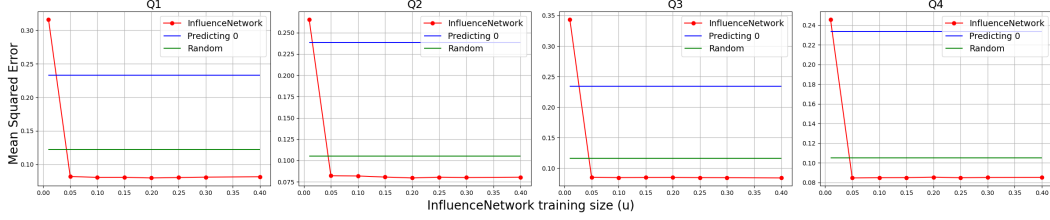


Figure 2: MSE versus InfluenceNetwork training data size (u) plotted for 8 different training sizes, broken down by the quadrant. These results are for learning DELIFT influence values. Error rates on each quadrant correspond to losses across different sets: Q1 for training, Q2/Q3 for validation, and Q4 for testing. As shown, the InfluenceNetwork achieves MSE of merely 0.05% starting from $u = 0.05$ and always outperforms the baselines.

specific tasks like math reasoning, question answering, or summarization. Similarly, (3)’s samples are specifically chosen to introduce new information to a model without overriding or repeating previously learned information.

3 Problem Formulation

Given a model $\mathcal{M}$ and fine-tuning data $\mathcal{D}_{\mathcal{F}}$, the goal is to select a small subset $\mathcal{S}_{\mathcal{F}} \subset \mathcal{D}_{\mathcal{F}}$ that maximizes the performance of $\mathcal{M}$ after fine-tuning $\mathcal{M}$ on $\mathcal{S}_{\mathcal{F}}$. $\mathcal{S}_{\mathcal{F}}$ is the optimal subset if it can be used to train a model that is comparable to a model trained on $\mathcal{D}_{\mathcal{F}}$. However, more recent works jointly optimize other objectives during subset selection. Examples of objectives include not only representation, but also task-specific refinement and continual learning. For such joint optimization, the subset $\mathcal{S}_{\mathcal{F}}$ is aligned with another target domain dataset $\mathcal{D}_{\mathcal{T}}$. The choice of $\mathcal{D}_{\mathcal{T}}$ can guide the subset selection towards various objectives. For example, if the objective is representation or task-specific refinement, $\mathcal{S}_{\mathcal{F}}$ will contain points from $\mathcal{D}_{\mathcal{F}}$ that are similar to $\mathcal{D}_{\mathcal{T}}$ [Liu et al., 2024c, Xia et al., 2024, Das and Khetan, 2024]. Alternatively, if the objective is continual learning, $\mathcal{S}_{\mathcal{F}}$ will contain points from $\mathcal{D}_{\mathcal{F}}$ that would allow the model $\mathcal{M}$ to learn new information that is present in $\mathcal{D}_{\mathcal{T}}$ Agarwal et al. [2025], Tiwari et al. [2022].

As mentioned before, computing influence functions can be a very expensive process. There are two kinds of influence functions: pairwise and pointwise – both require forward/backward passes through language models, but the costs slightly differ. Pairwise influence functions compute the influence between every pair of points in a dataset. We study three SOTA pairwise functions, whose formulations are details in Appendix C.1. This paper also studies one pointwise influence functions that simply compute the influence of each data point individually, formally outlined in Appendix C.2. While pointwise influence functions are more efficient than pairwise, they are not as performant during subset selection Xia et al. [2024], Agarwal et al. [2025].

3.1 Our motivation

Overall, our aim is to reduce the total number of forward or back propagations through models with millions and billions of parameters by replacing a large portion with forward propagations through small neural networks with (merely) hundreds of thousands of parameters. Pairwise influence functions calculate the similarity between two data points (denoted as $\text{sim}(i, j)$). Because influence values are usually not learned, they need to be recomputed for any data beyond the training data. In other words, as data is constantly being collected, influence values for new data must be recomputed. However, NN-CIFT is learned. Hence, *our method does not require any extra computation to estimate influence values*, unlike previous work.

4 Learning Influence Estimation

This section describes in detail Steps 1 and 2 in Figure 1. It outlines the structure and initial experimentation of the InfluenceNetwork.

4.1 Defining the InfluenceNetwork.

For estimating the influence values of data samples, we call our neural network the *InfluenceNetwork*. It is a 2-layer neural network with a hidden size of 100 neurons, and an output size of 1 neuron. For activation, we use ReLU in between the layers. The function IN_θ represents the neural network with parameters θ . As input, IN_θ takes two data points i and j and outputs the estimated influence of i on j . Specifically, embeddings for i and j are computed (denoted as $\text{emb}()$ below) using the BAAI General Embedding model (bge-large-en-v1.5, in particular) [Xiao et al., 2023] and are concatenated:

$$0 \leq IN_\theta(i, j) \leq 1, \quad (1)$$

$$0 \leq \theta(\text{concat}(\text{emb}(i), \text{emb}(j))) \leq 1, \quad (2)$$

$$\forall (i, j) \in \mathcal{D}_\mathcal{F} \times \mathcal{D}_\mathcal{T} \quad (3)$$

The bge-large-en-v1.5 model generates embeddings of size 1,024, which means the input has a total length of 2,048. Hence, the InfluenceNetwork has exactly 204,900 parameters. For training, we use 20 epochs and a learning rate $\eta = 0.0001$.

We note that NN-CIFT relies on an embedding model that is often larger than the actual size of the InfluenceNetwork. However, we choose not to include it in the cost for a few reasons: (1) our method does not rely on the underlying embedding model, (2) NLP pipelines generally use embedding models to store, retrieve, cluster, and/or visualize data and hence, is an offline cost.

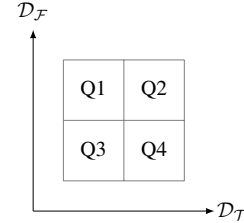
4.2 Training the InfluenceNetwork.

Below is an illustration of the quadratic similarity matrix that is computed during the data valuation stages. Previous influence compute the entire matrix for data valuation – we only use Q1.

Using the predefined influence functions in Appendix C, a small fraction of influence values are computed – we call this fraction u . We use $u\%$ of data from $\mathcal{D}_\mathcal{F}$ and $u\%$ of data from $\mathcal{D}_\mathcal{T}$ to compute the training set for the InfluenceNetwork. As mentioned above, this training set is represented by Q1 in the illustration.

The quadrants Q1 to Q4 represent the subset of influence values between a combination of in-distribution (ID) data and out-of-distribution (OOD) data. ID and OOD data is determined by whether the InfluenceNetwork was trained on the data (ID) or not (OOD):

- Q1: Fully ID data from $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}_\mathcal{T}$
- Q2: ID data from $\mathcal{D}_\mathcal{F}$ and OOD data from $\mathcal{D}_\mathcal{T}$
- Q3: OOD data from $\mathcal{D}_\mathcal{F}$ and ID data from $\mathcal{D}_\mathcal{T}$
- Q4: Fully OOD data from $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}_\mathcal{T}$



4.3 Evaluating the InfluenceNetwork.

To ensure our InfluenceNetwork is able to output influence values correctly, we compute the average mean squared error (MSE) between the ground truth influence values (from Appendix C) and the predicted influence values:

$$\frac{1}{|\mathcal{D}_\mathcal{F} \times \mathcal{D}_\mathcal{T}|} \sum_{(i,j) \in \mathcal{D}_\mathcal{F} \times \mathcal{D}_\mathcal{T}} (IF_\theta(i, j) - \text{sim}(i, j))^2$$

We separate the evaluation between the four quadrants of data to study the performance with ID and OOD data.

To train the InfluenceNetwork, we use DELIFT’s influence values on the MixInstruct dataset [Jiang et al., 2023] to train our InfluenceNetwork (more dataset details in Section 5). We report the results from *InfluenceNetwork* and two other baselines: (1) *Randomly* generating a number between 0 and 1, and (2) only *Predicting 0* influence. These results can be found in Figure 2.

u	Q1 std	Q2 std	Q3 std	Q4 std
0.01	0.0050	0.0047	0.0018	0.0118
0.05	0.0090	0.0091	0.0032	0.0032
0.10	0.0062	0.0058	0.0032	0.0032
0.15	0.0062	0.0056	0.0037	0.0037
0.20	0.0102	0.0096	0.0020	0.0020
0.25	0.0102	0.0096	0.0020	0.0020
0.30	0.0118	0.0112	0.0020	0.0020
0.40	0.0067	0.0053	0.0018	0.0018

Table 2: Variance of the InfluenceNetwork for varying u 's for DELIFT values.

Embedding Model	Q1	Q2	Q3	Q4
BAAE/bge-large-en-v1.5	0.051	0.084	0.074	0.084
Qwen/Qwen3-Embedding-0.6B	0.076	0.087	0.087	0.089
intfloat/e5-mistral-7b-instruct	0.026	0.083	0.084	0.083
Snowflake/snowflake-arctic-embed-l-v2.0	0.077	0.087	0.087	0.088

Table 3: Varying embedding models and their corresponding MSE values (averaged across 5 runs) between estimated influence values and ground truth influence values, with 5% selected data. These results are for learning DELIFT influence values. As shown, NN-CIFT is invariant to the embedding model selected, and is able to effectively estimate the influence values.

The InfluenceNetwork is able to predict influence values with low error rates. After just $u = 0.05$, it is consistently better than random influence values and predicting only 0. The average MSE between the InfluenceNetwork's influence scores and DELIFT's influence scores is 0.072, 0.072, 0.062, 0.063 for Q1 to Q4, respectively (averaging to 0.067). Furthermore, **the error rate stays consistent across all four quadrants, showing that NN-CIFT does not need to be retrained to estimate the influence of new data points** that are collected after the training data. One thing to note is that *although $u = 0.05$, with pairwise influence functions, we end up using only 0.25% of the data to train the InfluenceNetwork because we use 5% of $\mathcal{D}_{\mathcal{F}}$ and 5% of $\mathcal{D}_{\mathcal{T}}$.*

Finally, we report the robustness of the InfluenceNetwork. In Table 2 reports the variance of the InfluenceNetwork after five runs. We assume the original influence values are given and static, and therefore, do not measure the variance of the individual influence functions. Table 2 shows low variance across each quadrant and each u in the InfluenceNetwork scores. Table 3 shows that the InfluenceNetwork is invariant to the choice of the embedding model. As the MSE between the predicted influence values and ground truth influence values remains small, we posit the downstream performance of NN-CIFT will be maintained no matter the choice of embedding model.

4.4 Interpretive Analysis

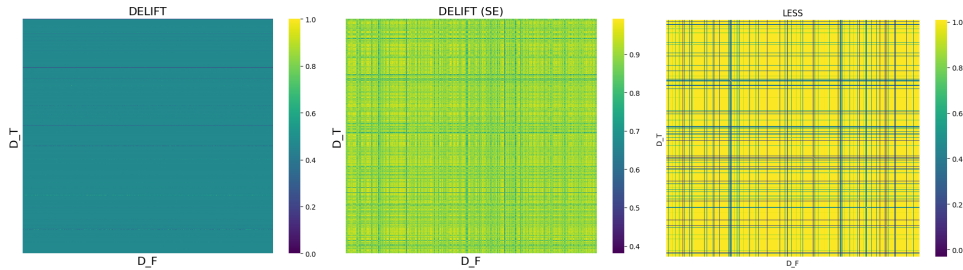


Figure 3: Distribution of influence values across each of the methods on the Alpaca dataset. To clarify, "D_F" is $\mathcal{D}_{\mathcal{F}}$ and "D_T" is $\mathcal{D}_{\mathcal{T}}$. The x-axis spans the 15,000 examples from $\mathcal{D}_{\mathcal{F}}$ and the y-axis spans the 5,000 samples from $\mathcal{D}_{\mathcal{T}}$. SelectIT only has an x-axis.

We posit two reasons that contribute to the InfluenceNetwork's success: (1) most data points have the same influence, and (2) influence estimation is a lossy task.

Firstly, we visualize the distribution of the influence values in Figure 3. As shown, the most "different" influence values (the stark dark/light colors indicating low/high influence) are sparse. This

Dataset	MixInstruct						Alpaca						MMLU	
Method	ICL			QLoRA			ICL			QLoRA			ICL	QLoRA
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	Accuracy	Accuracy
Initial	28.53	74.05	2.94	34.42	78.54	3.00	24.85	72.45	2.26	34.29	80.82	3.03	70.7	70.8
Random	40.07	84.04	3.26	41.68	84.26	3.22	36.95	80.47	3.12	38.64	80.46	3.07	71.9	71.9
SelectIT	46.51	86.18	3.25	50.31	87.38	3.25	41.42	83.25	3.27	44.51	84.18	3.34	72.7	73.0
+ DistilGPT2	41.26	80.33	3.20	44.86	84.72	3.23	39.18	80.99	2.99	41.72	81.50	3.14	72.0	72.7
+ NN-CIFT	46.48	85.86	2.28	50.87	87.43	3.26	42.07	83.67	3.27	44.99	85.13	3.37	74.7	72.9
LESS	48.21	86.19	3.34	51.24	86.07	3.37	43.34	84.19	3.38	44.73	84.04	3.32	75.6	76.7
+ DistilGPT2	42.18	78.34	3.23	48.64	79.09	3.27	42.02	80.89	3.29	42.51	82.35	3.29	74.6	74.1
+ NN-CIFT	48.20	86.31	3.36	51.56	86.39	3.41	44.42	84.69	3.32	46.40	85.44	3.36	75.0	76.5
DELIFT (SE)	48.36	85.91	3.38	51.43	86.20	3.34	44.30	85.52	3.41	45.35	86.34	3.48	78.9	79.8
+ DistilGPT2	47.21	84.24	3.28	49.37	84.24	3.29	43.51	85.45	3.41	44.89	79.81	3.36	75.4	76.1
+ NN-CIFT	48.59	85.01	3.39	50.53	86.10	3.33	45.49	86.27	3.44	45.75	86.45	3.47	78.5	79.8
DELIFT	51.66	88.02	3.43	55.58	91.81	3.50	46.49	87.60	3.50	49.16	87.74	3.54	81.5	83.1
+ DistilGPT2	47.09	84.74	3.26	48.21	84.24	3.28	45.08	81.45	3.41	41.07	83.22	3.44	77.1	78.5
+ NN-CIFT	52.03	88.38	3.41	55.85	91.96	3.51	46.26	87.41	3.55	49.15	87.74	3.50	82.0	83.6
Full Data	54.43	92.55	3.40	59.47	94.12	3.58	48.53	91.21	3.63	48.29	90.82	3.66	80.5	81.6

Table 4: Results on the Llama-8B model with $v = 0.3, u = 0.05$. “+ NN-CIFT” indicates using NN-CIFT to estimate influence values computed from the corresponding method’s influence function. “+ DistilGPT2” indicates using the DistilGPT2 model as the language model in the corresponding method’s influence function. The average performance difference between NN-CIFT and the original influence function is merely 0.13%.

Dataset	MixInstruct						Alpaca						MMLU	
Method	ICL			QLoRA			ICL			QLoRA			ICL	QLoRA
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	Accuracy	Accuracy
Initial	16.19	61.32	2.06	19.31	64.27	2.09	24.53	71.42	2.48	24.80	71.79	2.61	69.4	70.9
Random	28.33	72.41	2.37	29.93	75.78	2.50	26.54	72.71	2.66	28.10	73.00	2.78	68.9	71.9
SelectIT	40.64	72.63	2.21	44.85	75.72	2.83	31.86	76.29	2.73	32.56	78.17	2.77	71.7	71.6
+ DistilGPT2	40.33	71.49	2.11	43.26	74.14	2.38	29.65	75.32	2.62	31.66	74.20	2.67	69.9	70.8
+ NN-CIFT	41.59	71.36	2.26	45.87	74.22	2.58	32.05	76.82	2.78	32.90	77.12	2.80	71.1	72.0
LESS	45.33	78.68	3.03	46.03	81.04	3.05	38.43	78.83	2.98	41.68	81.83	3.09	74.0	75.6
+ DistilGPT2	43.15	74.69	2.42	42.87	75.80	2.46	34.76	75.26	2.86	37.66	78.57	3.06	66.7	69.3
+ NN-CIFT	46.32	78.84	3.05	47.84	80.48	3.03	39.59	78.75	2.89	42.06	81.06	3.08	74.1	74.9
DELIFT (SE)	46.68	81.01	3.12	48.42	83.67	3.15	42.52	81.21	3.12	43.83	84.35	3.26	74.0	75.1
+ DistilGPT2	45.89	79.77	3.07	46.07	80.40	3.10	41.15	79.42	2.96	42.32	82.63	3.02	75.0	75.4
+ NN-CIFT	46.81	81.23	3.14	48.64	82.76	3.16	42.72	80.86	3.16	42.75	84.52	3.27	75.2	75.5
DELIFT	48.85	83.89	3.25	50.90	85.64	3.27	44.74	83.60	3.28	46.33	87.87	3.46	77.4	79.4
+ DistilGPT2	43.69	77.83	3.04	45.61	79.76	3.06	40.25	77.10	2.96	43.37	79.88	2.96	74.6	75.4
+ NN-CIFT	48.97	83.57	3.25	49.71	86.45	3.29	45.78	85.49	3.29	48.69	87.14	3.48	77.5	79.3
Full Data	49.31	86.25	3.48	52.55	89.58	3.51	49.31	89.39	3.45	50.95	90.23	3.66	73.7	74.6

Table 5: Results on the Qwen2.5 with $v = 0.3, u = 0.05$. “+ NN-CIFT” indicates using NN-CIFT to estimate influence values computed from the corresponding method’s influence function. “+ DistilGPT2” indicates using the DistilGPT2 model as the language model in the corresponding method’s influence function. The average performance difference between NN-CIFT and the original influence function is merely 0.24%.

indicates that *the neural network simply has to learn to estimate the extremes*, and can achieve good performance for the rest of the values. Second, we notice that *influence estimation is an incredibly lossy task*. It involves compressing high-dimensional text representations to a singular, scalar value. Furthermore, the scalar values are constrained to a small range (see Equation 1), reducing the margin of error. Putting these two reasons together, we can see why a neural network can replace a language model during influence estimation without any significant effect on performance.

Dataset	MixInstruct						Alpaca						MMLU	
Method	ICL			QLoRA			ICL			QLoRA			ICL	QLoRA
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	Accuracy	Accuracy
Initial	32.31	74.27	2.18	35.16	78.13	2.21	37.44	78.65	2.41	37.81	78.69	2.43	71.9	72.3
Random	35.83	80.62	2.98	37.13	81.57	3.01	40.44	82.75	2.48	40.19	81.15	2.46	71.3	72.3
SelectIT	40.64	84.99	3.17	45.97	86.31	3.20	42.72	82.00	2.54	43.06	83.59	2.67	73.6	75.8
+ DistilGPT2	39.84	80.48	3.02	41.81	81.21	3.03	40.74	81.44	2.37	40.36	81.85	2.53	73.5	73.8
+ NN-CIFT	40.33	84.95	3.14	46.85	86.06	3.19	41.52	82.18	2.57	43.45	82.86	2.63	73.6	75.4
LESS	48.33	85.78	3.30	49.93	87.42	3.37	44.14	85.80	3.04	47.13	87.94	3.05	75.4	76.6
+ DistilGPT2	45.22	82.80	3.23	44.53	84.56	3.31	40.92	80.23	2.68	42.65	84.29	2.96	74.5	74.4
+ NN-CIFT	48.75	86.15	3.33	51.93	86.57	3.41	44.43	85.79	3.07	47.90	88.01	3.04	75.2	77.3
DELIFT (SE)	48.84	88.10	3.42	48.85	88.16	3.44	45.13	87.34	3.35	48.47	89.19	3.42	76.7	77.1
+ DistilGPT2	46.25	86.22	3.39	46.29	87.67	3.41	44.84	86.00	3.25	46.86	86.93	3.38	75.7	77.7
+ NN-CIFT	48.81	88.77	3.41	49.14	88.33	3.44	46.01	86.43	3.36	47.99	88.03	3.41	76.8	77.4
DELIFT	53.70	91.52	3.54	54.69	92.42	3.56	50.36	89.53	3.40	53.65	91.79	3.53	78.2	80.4
+ DistilGPT2	46.55	88.81	3.41	47.78	89.34	3.46	45.31	83.69	3.23	44.78	85.08	3.31	75.3	76.9
+ NN-CIFT	53.53	90.82	3.52	56.65	91.42	3.55	50.67	88.97	3.43	52.48	92.03	3.53	78.4	80.9
Full Data	52.30	91.32	3.60	54.83	91.57	3.69	50.61	90.77	3.41	54.51	90.70	3.58	79.1	81.2

Table 6: Results on the Mistral model with $v = 0.3, u = 0.05$. “+ NN-CIFT” indicates using NN-CIFT to estimate influence values computed from the corresponding method’s influence function. “+ DistilGPT2” indicates using the DistilGPT2 model as the language model in the corresponding method’s influence function. The average performance difference between NN-CIFT and the original influence function is merely 0.12%.

5 Subset Selection Evaluation

Motivated by the results in Figure 2, we apply the InfluenceNetwork to the downstream task of subset selection: *can we achieve the same performance when using the InfluenceNetwork instead of the original influence function?* Thus, this section corresponds to Step 3 in Figure 1.

Datasets and models. We use MixInstruct, [Taori et al., 2023], Alpaca [Taori et al., 2023], and MMLU [Hendrycks et al., 2021] to evaluate NN-CIFT. These are instruction-tuning, preference alignment, and knowledge-based benchmarks where we use 15k for training, 5k for validation, and 5k for testing. We evaluate using three models: meta-llama/Llama-3.1-8B [Grattafiori et al., 2024], Qwen/Qwen2.5-1.5B [Qwen et al., 2025], and mistralai/Mistral-Small-Instruct-2409 (22.2B) [Jiang et al.]. Note, we use Llama-8b, Qwen2.5 and Mistral as shorthand for the rest of the experimental section.

Metrics. To evaluate the instruction following capabilities of our fine-tuned model $\mathcal{M}'$, we employ a variety of metrics to capture the similarity between ground truth answers and predicted answers from $\mathcal{M}'$: (1) ROUGE [Lin, 2004]: n -gram word overlap (specifically, rouge-1), (2) BGE: semantic similarity of embeddings using bge-large-en-v1.5, and (3) LAJ: an LLM-as-a-Judge, namely the prometheus-7b-v2.0 model [Kim et al., 2023]. Prometheus’ grading rubric is borrowed from Agarwal et al. [2025] in Appendix B. Next, to evaluate the costs of each method, we use time (in seconds) took on 2 Nvidia A40 GPUs. However, for MMLU, we use classification accuracy.

Baselines. Besides the influence functions DELIFT, DELIFT (SE), LESS, and SelectIT, we include three other baselines: Initial, DistilGPT2, and Full Data. *Initial* is the setting where $v = 0.0$. This is the base model’s performance on the dataset. Next, we use a small language model *DistilGPT2* (distilbert/distilgpt2) [Sanh et al., 2020] which has 88.2M parameters as the underlying language/embedding model in the influence functions. Finally, *Full Data* is the setting where $v = 1.0$, i.e., the model’s performance when the full dataset is used.

Setup. We use $u = 0.05$ for training the InfluenceNetwork. We also use a small fraction of $\mathcal{D}_{\mathcal{F}}$ to fine-tune the language model – we call this fraction v . We evaluate with $v = 0.3$. Our evaluation framework includes two different settings to fine-tune the language model: using the selected subset of data points as (1) PEFT data for QLoRA [Detrmers et al., 2023] on $\mathcal{M}$, or (2) in-context learning (ICL) examples. To elaborate on the ICL set up, we choose the top-5 most semantically similar

samples from the chosen subset to add in-context. To measure semantic similarity, we again use `bge-large-en-v1.5`. Table 4-6 reports results for each model on all three datasets with $v = 0.3$; Table 7 reports the cost in time for each method. All tables report the results for one run.

5.1 Analysis

Table 7 reports the costs for each method, in seconds. It shows that **data valuation can be performed at 77-99% faster** than the original influence functions. This is because the number of parameters in NN-CIFT is 0.00096-0.013% the size of the language model in the original influence function. Also, when using the DistilGPT2 model, which is near 1% the size of the language model, the costs are reduced by 54-91%. While these results are promising, the results on the downstream task of subset selection clearly differentiate NN-CIFT and the DistilGPT2 baseline. **Despite the significant speedups, NN-CIFT shows no compromise to performance**, as shown in Tables 4-6.

To begin, the pairwise functions outperform the pointwise function (SelectIT) because they are able to capture more fine-grained effects of the data point on a model’s learning. Next, DELIFT and DELIFT (SE) are able to outperform LESS because the theoretical guarantees of using submodular functions yields improved empirical performance. Finally, DELIFT uses model dependent information, tailoring the subset to the model’s weaknesses, allowing it to outperform DELIFT (SE).

Keeping these in mind, **NN-CIFT is able to achieve performance comparable to the original data valuation methods, even across models and datasets**. DistilGPT2 shows performance degradations, especially in the model-dependent methods (DELIFT, LESS, and SelectIT). This is because the model-dependent methods experience significant performance gains when the data valuation model is the same as the fine-tuning model. We note that our evaluation’s focus is that NN-CIFT works as well as the original influence functions, and not the comparison of performance between them.

The absolute average performance difference across metrics between the original influence functions and NN-CIFT is only 0.16%². Because the neural network is able to estimate the influence values with great accuracy, the selected subsets of data would be mostly the same between the original influence function and NN-CIFT. Hence, the performance difference of 0.16% can be attributed as the variability in the language model’s performance between two runs. Additionally, this trend is consistent across datasets and models, which shows the wide applicability of our method.

6 Conclusion

In this paper, we introduce NN-CIFT: Neural Networks for effiCient Instruction Fine-Tuning to distill highly parameterized models used in modern influence functions into small neural networks. We empirically show the effectiveness of our InfluenceNetwork design through low prediction error rates, and competitive performance on the downstream task of subset selection for IFT. We use four different influence functions to test with NN-CIFT; our experimentation shows that NN-CIFT can lower costs for expensive data valuation, is adaptive to all kinds of influence functions (model-dependent or -independent; pairwise or pointwise), and does not require retraining for new data. Future work will focus on two things: applicability and more targeted modeling. To improve applicability, incorporate more fine-tuning stage objectives such as task-specific dataset selection or continual learning. To target the modeling, we can shift from learning influence between data points to estimating influence between data and a model parameters. Finally, we leave to future work to generalize to distributional shifts within the data (although, the training is lightweight enough that retraining would suffice for now).

References

- I. Agarwal, K. Killamsetty, L. Popa, and M. Danilevsky. DELIFT: Data efficient language model instruction fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=Fty0wTcemV>.

²The average performance difference is calculated by taking the absolute difference in performance, dividing it by the original performance, and then averaging this ratio across all settings (datasets, methods, metrics, baselines).

Model Dataset	Phi-3		Llama-8B	
	MixInstruct	Alpaca	MixInstruct	Alpaca
Initial	-	-	-	-
Random	12.4	12.3	12.9	12.3
SelectIT	7,047	6,594	6,671	6,470
DistilGPT2 + SelectIT	144	139	144	139
NN-CIFT + SelectIT	65	63	64	63
LESS	12,338	11,217	10,843	14,819
DistilGPT2 + LESS	1,291	1,278	1,291	1,278
NN-CIFT + LESS	78	75	74	84
DELIFT (SE)	216	218	218	219
DistilGPT2 + DELIFT(SE)	98	99	98	99
NN-CIFT + DELIFT (SE)	48	48	48	48
DELIFT	67,379	68,117	68,076	65,711
DistilGPT2 + DELIFT	8,058	7,790	8,058	7,790
NN-CIFT + DELIFT	215	217	217	211
Full Data	-	-	-	-

Table 7: Costs (in seconds) of data valuation. Specifically: **Random**: chooses a random subset of points. **SelectIT**: calculates the ranking scores for each data point according to Appendix C.2. **LESS**: computes the cosine similarity between pairs of projected gradients for $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{T}}$, according to Equation 6. **DELIFT (SE)**: computes the distance between each pair of embeddings $(i, j) : i \in \mathcal{D}_{\mathcal{F}}, j \in \mathcal{D}_{\mathcal{T}}$, according to Equation 5. **DELIFT**: computes the inference-based utility metric for each pair of embeddings (i, j) , according to Equation 4. **NN-CIFT**: Steps 1 and 2 in Figure 1. Note, the costs of DistilGPT2 are the same across both models because they use the same data valuation.

J. Bilmes. Submodularity in machine learning and artificial intelligence, 2022. URL <https://arxiv.org/abs/2202.00132>.

D. Das and V. Khetan. Deft: Data efficient fine-tuning for pre-trained language models via unsupervised core-set selection, 2024. URL <https://arxiv.org/abs/2310.16776>.

T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. URL <https://arxiv.org/abs/2305.14314>.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Srivankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnston, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Karadas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly,

R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A. L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

A. Jiang, A. Sablayrolles, A. Tacnet, A. Kothari, A. Roux, A. Mensch, A. Herblin-Stoop, A. Garreau, A. Birky, Bam4d, B. Bout, B. de Monicault, B. Savary, C. Rambaud, C. Feldman, D. S. Chaplot, D. de las Casas, D. Costa, E. Arcelin, E. B. Hanna, E. Metzger, G. Blanchet, G. Lengyel, G. Bour, G. Lample, H. Rajaona, H. Roussez, H. Sattouf, I. Mack, J.-M. Delignon, J. Chudnovsky, J. Murke, K. Khandelwal, L. Stewart, L. Martin, L. TERNON, L. Saulnier, L. R. Lavaud, M. Jennings, M. Pellat, M. Torelli, M.-A. Lachaux, M. Janiewicz, M. Seznec, N. Schuhl, N. Muhs, O. de Garrigues, P. von Platen, P. Jacob, P. Buche, P. K. Reddy, P. Savas, P. Stock, R. Sauvestre, S. Vaze, S. Subramanian, S. Garg, S. Yang, S. Antoniak, T. L. Scao, T. Schueller, T. Lavril, T. Wang, T. Gervet, T. Lacroix,

- V. Nemychnikova, W. Shang, W. E. Sayed, and W. Marshall. URL <https://mistral.ai/news/mistral-small-3-1>.
- D. Jiang, X. Ren, and B. Y. Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.
- K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, A. De, and R. Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training, 2021. URL <https://arxiv.org/abs/2103.00123>.
- S. Kim, J. Shin, Y. Cho, J. Jang, S. Longpre, H. Lee, S. Yun, S. Shin, S. Kim, J. Thorne, et al. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*, 2023.
- P. W. Koh and P. Liang. Understanding black-box predictions via influence functions, 2020. URL <https://arxiv.org/abs/1703.04730>.
- C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- L. Liu, X. Liu, D. F. Wong, D. Li, Z. Wang, B. Hu, and M. Zhang. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection, 2024a. URL <https://arxiv.org/abs/2402.16705>.
- W. Liu, W. Zeng, K. He, Y. Jiang, and J. He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning, 2024b. URL <https://arxiv.org/abs/2312.15685>.
- Z. Liu, A. Karbasi, and T. Rekatsinas. Tsds: Data selection for task-specific model finetuning, 2024c. URL <https://arxiv.org/abs/2410.11303>.
- B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models, 2020. URL <https://arxiv.org/abs/1906.01827>.
- Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- H. S. V. N. S. K. Renduchintala, S. Bhatia, and G. Ramakrishnan. Smart: Submodular data mixture strategy for instruction tuning, 2024. URL <https://arxiv.org/abs/2403.08370>.
- V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL <https://arxiv.org/abs/1910.01108>.
- E. S. Scanlon. Residuals and influence in regression. *New York: Chapman and Hall*, 1982.
- R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, June 2022.
- J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, and T. Ma. Larger language models do in-context learning differently, 2023. URL <https://arxiv.org/abs/2303.03846>.
- M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang. Instruction tuning for large language models: A survey, 2024. URL <https://arxiv.org/abs/2308.10792>.

A Hyperparameter Studies

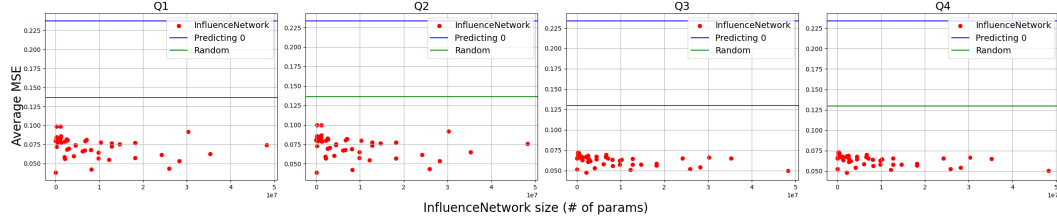


Figure 4: MSE versus InfluenceNetwork sizes (measured by the number of parameters). We try 1-5 layers with 46 different combinations of hidden layer sizes from {5, 10, 20, 50, 100, 200, 500, 1000, 2000, 3000, 4000, 5000}.

A.1 Hyperparameter Study #1: InfluenceNetwork sizes

We vary the number of layers and dimensions of each layer. For simplicity, we plot the number of parameters in the InfluenceNetwork versus the MSE. The results can be found in Figure 4. This figure shows that small InfluenceNetwork’s perform comparatively well as larger InfluenceNetwork’s.

A.2 Hyperparameter study #2: Trade-off between u and v

We perform a hyperparameter study between u and v on MixInstruct using DELIFT’s influence function (Equation 4). We perform a grid search where $u = v = \{0, 0.01, 0.05, 0.1, 0.15, 0.20, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, amounting to 169 experiments. Figure 8 shows the results using the BGE metric from each of these experiments. As shown, the two figures in each row follow the same general trend, showcasing that NN-CIFT can effectively replace the expensive influence function estimation.

As expected, we notice a few trends. (1) QLoRA generally has better performance than ICL. This is because fine-tuning has more impact on the model than simply adding examples to the prompt (i.e., prompt engineering). (2) The bottom right tends to be darker as fewer IFT data lead to insufficient training. (3) Larger IFT subsets, especially in the ICL setting, lead to poorer performance. During ICL, the top-5 semantically similar samples are chosen from the subset to add as in-context examples. However, semantic similarity does not always translate to performance enhancement as these samples can be harmful to the model’s performance. Finally, a follow-up to (3), the highest performance regions tend to be around $v = 0.2 - 0.4$. Appendix B contains results on smaller subsets of IFT data ($v = 0.1$ and 0.2).

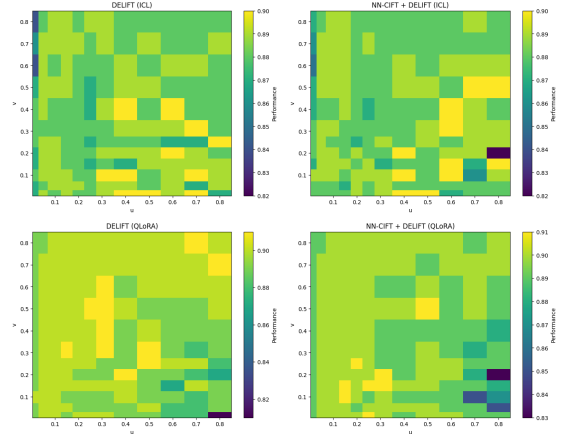


Table 8: Hyperparameter study for u and v on MixInstruct with DELIFT’s influence function. Lighter colors indicate better BGE performance.

Dataset	MixInstruct						Alpaca					
	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	37.87	78.92	2.98	36.36	82.55	3.02	25.79	67.82	2.56	27.29	71.57	2.62
Random	37.51	78.01	3.05	35.55	82.13	3.04	24.33	67.37	2.84	29.34	70.86	3.06
SelectIT	33.20	72.12	3.12	37.00	73.45	3.13	24.48	67.48	2.86	30.06	68.06	3.04
NN-CIFT + SelectIT	33.55	72.15	3.07	35.38	72.45	3.18	26.41	65.57	2.81	28.78	67.83	2.99
LESS	32.57	72.07	3.05	34.61	72.82	3.18	26.15	69.83	2.81	28.53	67.17	2.99
NN-CIFT + LESS	33.19	72.94	3.02	35.42	72.03	3.18	24.63	70.11	2.84	27.63	67.41	2.51
DELIFT (SE)	35.71	78.09	3.22	39.63	78.36	3.28	29.17	70.69	3.01	30.60	71.50	3.14
NN-CIFT + DELIFT (SE)	36.34	78.02	3.22	39.75	78.76	3.33	29.22	72.28	3.03	30.23	71.01	3.16
DELIFT	36.45	78.11	3.23	39.83	78.83	3.29	30.15	74.01	3.18	37.81	78.49	3.31
NN-CIFT + DELIFT	36.17	78.16	3.22	38.08	78.25	3.28	31.95	74.84	3.26	37.26	78.36	3.28
Full Data	58.65	88.72	3.45	65.51	92.24	3.51	35.27	77.85	3.31	39.29	78.85	3.29

Table 9: Results on the Phi-3 model with $v = 0.1, u = 0.05$. NN-CIFT + Method and DistilGPT2 + Method follow the same definitions as in Table ?? . The average performance difference between NN-CIFT and the original influence function is merely 1.91%.

Dataset	MixInstruct						Alpaca					
	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	28.53	74.05	2.94	34.42	78.54	3.00	24.85	72.45	2.26	34.29	80.82	3.03
Random	35.67	76.30	3.18	37.20	80.63	3.19	30.82	75.38	2.82	36.95	80.48	3.05
SelectIT	36.53	78.69	3.14	36.95	81.51	3.20	31.52	75.69	2.84	38.06	81.51	3.19
NN-CIFT + SelectIT	35.57	78.86	3.17	37.20	80.56	3.21	30.52	74.86	2.88	37.20	80.55	3.13
LESS	35.31	77.07	3.19	37.46	80.86	3.23	31.31	75.07	2.71	37.45	80.85	3.23
NN-CIFT + LESS	35.16	78.11	3.16	37.93	81.36	3.20	32.16	76.11	2.75	37.93	81.35	3.21
DELIFT (SE)	35.13	77.71	3.12	36.78	79.69	3.15	30.14	73.71	2.61	36.80	79.69	3.15
NN-CIFT + DELIFT (SE)	35.12	78.69	3.13	37.33	80.34	3.08	31.12	74.69	2.62	37.33	80.34	3.08
DELIFT	37.82	80.55	3.18	37.61	82.63	3.20	31.82	75.62	2.83	37.61	80.55	3.29
NN-CIFT + DELIFT	37.52	81.02	3.15	37.88	82.01	3.19	31.55	75.04	2.79	37.88	81.16	3.29
Full Data	54.43	92.55	3.40	59.47	94.12	3.58	48.53	91.21	3.63	48.29	90.82	3.66

Table 10: Results on the Llama-8b model with $v = 0.1, u = 0.05$. NN-CIFT + Method and DistilGPT2 + Method follow the same definitions as in Table ?? . The average performance difference between NN-CIFT and the original influence function is merely 1.14%.

B Evaluation on Smaller Subsets

Tables 9 and 11 report extra results for the Phi-3 model on $v = 0.1$ and $v = 0.2$, respectively. Similarly, Tables 10 and 12 report results for Llama-8B on $v = 0.1$ and $v = 0.2$, respectively. With Tables ?? and 4 in the main text, these results show an increasing trend in performance with a higher subset of IFT data (i.e., higher v). They also show similar trends where NN-CIFT performs similarly to the original influence function.

C Influence Functions

Following the problem formulation, we formally define the influence functions we used throughout our evaluation.

C.1 Pairwise Influence Functions

DELIFT [Agarwal et al., 2025] is a model-dependent, inference-based metric. Samples $(i_x, i_y) \in \mathcal{D}_{\mathcal{F}}$ are used as in-context examples for evaluating $(j_x, j_y) \in \mathcal{D}_{\mathcal{T}}$, and those with improved model performance are chosen to represent $\mathcal{D}_{\mathcal{T}}$. This can be calculated by comparing the performance with and without (i_x, i_y) as an in-context example (where $D(\cdot, \cdot) \in [0, 1]$ is a function to measure distance

Dataset	MixInstruct						Alpaca					
	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	37.87	78.92	2.98	36.36	82.55	3.02	25.79	67.82	2.56	27.29	71.57	2.62
Random	37.91	78.96	3.06	38.89	81.88	3.05	29.95	76.35	3.12	30.27	76.21	3.15
SelectIT	35.39	78.14	3.02	37.71	78.26	3.06	30.31	74.26	3.13	37.10	77.66	3.10
NN-CIFT + SelectIT	35.71	78.23	3.04	37.36	78.24	3.05	31.03	75.79	3.09	36.67	77.98	3.04
LESS	37.61	79.55	3.07	37.43	78.93	3.09	32.57	74.07	3.02	34.61	76.68	3.08
NN-CIFT + LESS	37.87	77.96	3.04	38.96	78.93	3.08	33.20	74.94	3.05	35.42	78.02	3.09
DELIFT (SE)	39.56	81.25	3.17	39.77	82.74	3.15	34.06	77.31	3.23	39.48	80.95	3.25
NN-CIFT + DELIFT (SE)	39.62	81.47	3.16	39.14	82.83	3.14	33.01	76.67	3.27	38.89	80.80	3.20
DELIFT	45.55	82.32	3.36	43.74	82.35	3.50	35.02	77.89	3.40	39.32	80.89	3.35
NN-CIFT + DELIFT	46.44	82.47	3.38	43.76	82.72	3.52	34.44	77.39	3.36	38.30	80.32	3.31
Full Data	58.65	88.72	3.45	65.51	92.24	3.51	35.27	77.85	3.31	39.29	78.85	3.29

Table 11: Results on the Llama-8b model with $v = 0.2, u = 0.05$. NN-CIFT + Method and DistilGPT2 + Method follow the same definitions as in Table ?? . The average performance difference between NN-CIFT and the original influence function is merely 1.08%.

Dataset	MixInstruct						Alpaca					
	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	28.53	74.05	2.94	34.42	78.54	3.00	24.85	72.45	2.26	34.29	80.82	3.03
Random	39.55	82.79	3.25	39.05	82.64	3.26	31.49	76.96	3.06	41.67	79.77	3.14
SelectIT	39.20	82.84	3.29	40.44	82.55	3.30	35.98	81.82	2.95	42.62	83.17	3.21
NN-CIFT + SelectIT	40.02	82.63	3.23	39.92	82.22	3.29	38.84	84.09	3.03	44.62	84.63	3.23
LESS	40.33	82.17	3.26	40.34	82.87	3.26	36.11	79.82	3.06	43.48	82.94	3.32
NN-CIFT + LESS	43.69	82.67	3.27	40.21	82.89	3.26	37.00	80.38	3.07	43.48	82.80	3.34
DELIFT (SE)	44.57	82.63	3.31	45.97	83.87	3.33	38.52	82.37	3.18	45.73	83.33	3.35
NN-CIFT + DELIFT (SE)	45.03	83.69	3.30	45.97	83.95	3.40	38.57	82.18	3.17	45.20	82.79	3.39
DELIFT	45.55	83.69	3.37	48.21	86.81	3.36	39.16	82.30	3.26	45.24	83.38	3.39
NN-CIFT + DELIFT	46.40	84.73	3.34	47.81	86.83	3.31	40.16	82.37	3.28	45.67	83.49	3.41
Full Data	54.43	92.55	3.40	59.47	94.12	3.58	48.53	91.21	3.63	48.29	90.82	3.66

Table 12: Results on the Llama-8b model with $v = 0.2, u = 0.05$. NN-CIFT + Method and DistilGPT2 + Method follow the same definitions as in Table ?? . The average performance difference between NN-CIFT and the original influence function is merely 1.26%.

between two probability distributions, and $f(q|\theta)$ is a language model with parameters θ and input query q):

$$\text{sim}(i, j) = D(j_y, f(i_x, i_y, j_x|\theta)) - D(j_y, f(j_x|\theta)) \quad (4)$$

After data valuation, the data selection stage consists of using submodular functions [Bilmes, 2022]. In particular, we use the Facility Location submodular function. It takes as input a similarity kernel that will optimize the maximum similarity between the chosen subset and the overall dataset while also minimizing the size of the chosen subset. To minimize the subset size, the Facility Location – and submodular functions, in general – employ a diminishing gains property. This property states that samples added to a smaller subset have more value than samples added to a larger subset. Hence, we rely on our influence function to capture the informativeness of samples, and submodular functions to choose a set of representative samples, resulting in a small, information-rich subset on which to fine-tune a model.

DELIFT (SE) [Agarwal et al., 2025] is a model-independent metric, and chooses samples from $\mathcal{D}_{\mathcal{F}}$ which are semantically closest to the samples from $\mathcal{D}_{\mathcal{T}}$. Semantic distance is calculated by the cosine distance between embeddings of samples:

$$\text{sim}(i, j) = \frac{\langle \text{emb}((i_x, i_y)), \text{emb}((j_x, j_y)) \rangle}{\|\text{emb}((i_x, i_y))\| \cdot \|\text{emb}((j_x, j_y))\|} \quad (5)$$

, where $\text{emb}(q)$ is an embedding model with input data q . Similar to DELIFT, DELIFT (SE) also uses the Facility Location function to select a small, information-rich subset of samples.

LESS [Xia et al., 2024] is model-dependent, gradient-based metric. Here, gradients between samples in $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{T}}$ are matched by cosine similarity, and those that match the highest are chosen to represent $\mathcal{D}_{\mathcal{T}}$ (where $\nabla(q; \theta)$ is the gradient of data point q from a model with parameters θ):

$$\text{sim}(i, j) = \frac{\langle \nabla((i_x, i_y); \theta), \nabla((j_x, j_y); \theta) \rangle}{\|\nabla((i_x, i_y); \theta)\| \cdot \|\nabla((j_x, j_y); \theta)\|} \quad (6)$$

During the data selection stage, the top- k matching gradients are chosen to be part of the subset. One thing to notice is that the above equation implies a quadratic computation while Table 1 in the main text denotes a linear computation – this is because the gradients for each data point only need to be computed once, while the cosine similarity can be computed many times inexpensively.

C.2 Pointwise Influence Functions

Finally, **SelectIT** Liu et al. [2024a] is another model-dependent metric that uses performance signals for data valuation, but incurs linear cost as it uses a model’s uncertainty to rank data samples. Still, as mentioned in Table 1 from the main text, the linear time operations are forward propagations through LLMs.

SelectIT ranks data points based on their token-level, sentence-level, and model-level uncertainty expressed via token distribution. The token-level uncertainty is represented as the maximum probability of a token during next-token prediction. The sentence-level uncertainty is computed based on the token-level uncertainties of all the tokens in a sentence, for each prompt in a pool of prompts. Finally, the model-level uncertainty is calculated by taking a weighted average of the sentence-level uncertainty scores for multiple model sizes (the weights are determined by model size). This three-stage process provides a ranking process – thus, during data selection, the points with the top- k scores are chosen.

D License

All the code of this project is under the Apache 2.0 License. The datasets MixInstruct and Alpaca are under the MIT and Creative Commons Attribution Non Commercial 4.0 International Licenses, respectively. The code for the baselines are under the MIT and Apache 2.0 Licenses. Our use of existing artifact(s) is consistent with their intended use. The artifacts are all in English, and do not contain data with personally identifiable information.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: All of our claims are supported by the empirical results throughout the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We pose future work directions that address the limitations of our method in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: Our paper does not make significant theoretical contributions to warrant proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We not only include our code, but also detail the evaluation set up throughout Sections 4 and 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include an anonymized link to our code base in the abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We detail our experimental settings throughout Sections 4 and 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: This analysis can be found in Table 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We mention that we use 2 NVIDIA A40 GPUs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We followed the Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide a summary of societal impacts in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We only use open-source data and models under license (specified in Appendix D).

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Please refer to Appendix D.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We release code, which is documented and under license.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: We do not use human subjects in this work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: We do not use human subjects in this work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: As this work is to improve LLMs, we clearly outline the (all open-source) LLMs that were used for experimentation purposes.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.