A. Related work

We refer the readers to a recent survey [14] for a detailed account of research in data influence estimation. In this section, we will focus on key ideas and representative prior works in this field and elaborate on their connections with our work.

Gradient-based influence estimation. Intuitively, when models are trained with data, each training data has a unique gradient trace. Many studies have been focusing on measuring the influence of data through the lens of gradient alignment between training and test samples. Koh and Liang [22] leveraged a classic robust statistics concept, Influence Function [6], to quantify training data influence. `IF` evaluate the effect of infinitesimal change on the loss associated with an individual training point on the test loss. The computation of `IF` requires inverting the Hessian matrix, which is prohibitively expensive for modern neural networks. To tackle this challenge, Koh and Liang [22] proposed to compute `IF` via iteratively approximating the Hessian-vector product (HVP). Pruthi et al. [29] presented `TracIn`, a technique to estimate the influence of each training data by exploiting the gradient over all iterations. In particular, this influence estimator relies on the gradient of a test loss and a training loss. To scale up the approach, a practical alternative was proposed that considers a few checkpoints to calculate the gradient rather than using full iterations to approximate the data influence. Our work studies the duality associated with these gradient-based influence quantification schemes and leverages the duality to propose a more efficient alternative that does not require calculating gradients for individual training points.

Re-training based influence estimation. Re-training-based methods follow a general recipe that starts by training models on different training data subsets and then examining how the performance of these models changes when a given training point is added to the subsets. Ilyas et al. introduced Datamodel [16] which involves training thousands of models to estimate the data influence of each training datum. Specifically, this method leverages a parameterized surrogate model to predict the model performance based on the input training set and the surrogate model is learned from a training set consisting of pairs of an input subset and the corresponding model performance. Park et al. [27] proposed TRAK, which leveraged several techniques to enhance the efficiency of Datamodel. TRAK linearizes the model output function using Taylor approximation and reduces the dimensionality of the linearized model using random projections. However, it still requires repeated model training.

Another line of research does not train surrogate models for data influence estimation; instead, they directly compute a weighted average of the model performance changes in response to the addition of a training point across different subsets. Notable examples include the Shapley value [12, 17], Beta Shapley [24] and Banzhaf value [36], which differ in the design of the weighting scheme over different subsets. However, these methods face significant computational challenges for large models due to the need of retraining models on different subsets. Just et al. [19] recently proposed LAVA as a scalable solution that evaluates data influence on the model performance using optimal transport. Despite LAVA's efficiency, LAVA does not provide a way for monitoring the training data's contribution to the model prediction on the individual test point.

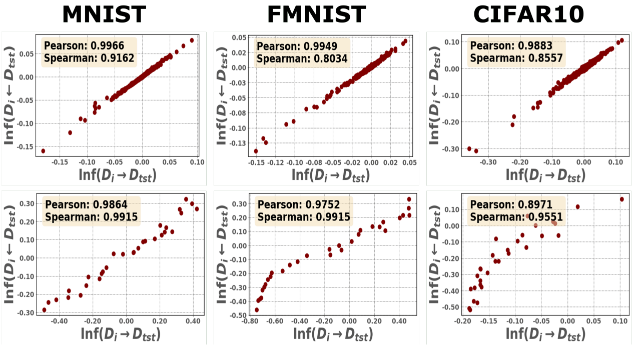


Figure 6. Correlation scores across different datasets under the noisy and near-noiseless settings. The first row shows the correlations between Inf$(D_i \rightarrow D_{\text{tst}})$ and Inf$(D_i \leftarrow D_{\text{tst}})$ under the near-noiseless setting, and the second row presents the correlation results under the noisy setting.

Validation Set Size		**50**	**100**	**1000**
MNIST	Pearson	0.9208	0.9512	0.9652
	Spearman	0.9755	0.9764	0.9942
FMNIST	Pearson	0.9244	0.9611	0.9786
	Spearman	0.9568	0.9795	0.9964
CIFAR10	Pearson	0.6918	0.8401	0.8551
	Spearman	0.8274	0.9034	0.8848

Table 4. Correlation scores across different sizes of the validation set. We keep the same mislabeled ratio (i.e., 0.5) while changing the sizes of the validation set.

B. Additional Results on Empirical Study of the Mirrored Influence Hypothesis [Section 2.1]

In this section, we delve deeper into our Mirrored Influence Hypothesis, as introduced in Section 2.1, by presenting a more detailed analysis and additional results.

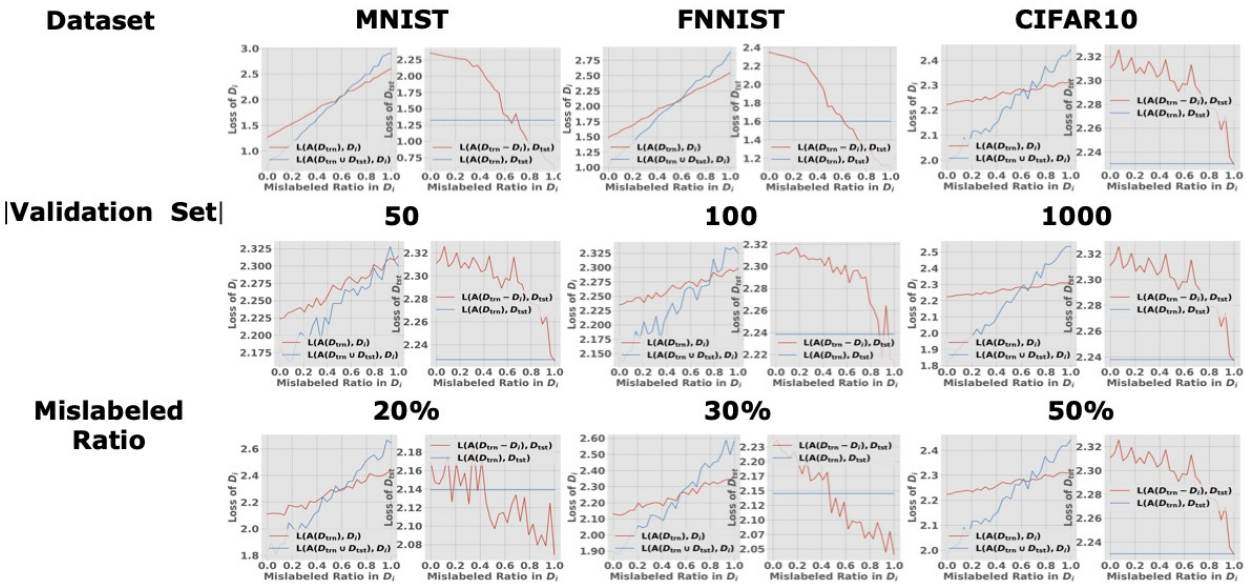


Figure 7. Detailed visualization of each value in the calculation of $\text{Inf}(D_i \rightarrow D_{\text{tst}})$ and $\text{Inf}(D_i \leftarrow D_{\text{tst}})$ across various datasets, validation set sizes, and mislabeled ratios. The x-axis represents different mislabeled ratios according to different groups, while the y-axis shows the value of each term (i.e.,$\mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_{\text{tst}})$, $\mathcal{L}(\mathcal{A}(D_{\text{trn}} \setminus D_i), D_{\text{tst}})$, $\mathcal{L}(\mathcal{A}(D_{\text{trn}} \cup D_{\text{tst}}), D_i)$, $\mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_i)$).

Mislabeled Ratio		**20%**	**30%**	**50%**
MNIST	Pearson	0.9621	0.9864	0.9640
	Spearman	0.9653	0.9915	0.9907
FMNIST	Pearson	0.9421	0.9857	0.9752
	Spearman	0.9479	0.9840	0.9915
CIFAR10	Pearson	0.7521	0.9697	0.8551
	Spearman	0.7838	0.9702	0.8848

Table 5. Correlation scores across different mislabeled ratios in D_{trn}. We keep the same validation set size (i.e., 500) while changing the mislabeled ratios.

Analysis of correlation scores across different datasets. We first demonstrate the validity of our hypothesis across different datasets under a near-noiseless setting and a noisy setting.

In the near-noiseless case, we train a logistic regression model using L-BFGS for 1000 iterations with L2 regularization set at 0.001. For the noisy setting, we train a convolutional neural network, consisting of two convolution layers and two MLP layers using SGD with a learning rate of 0.01, a weight decay of 0.001, and a momentum of 0.9. To increase the magnitude of influence of each data point, we select a subset of data, specifically 1050 samples from the MNIST and FMNIST datasets, and 900 samples from the CIFAR-10 dataset.

As shown in Figure 6, we observe the high Pearson and Spearman correlation scores obtained across three datasets and two settings. In the near-noiseless setting, we find high correlation scores across all datasets. In the noisy setting, the CIFAR-10 results show a slight decline in correlation scores. This decrease can be attributed to the limited sample size from CIFAR-10, which potentially led to higher loss and introduced more noise in score calculation.

Analysis of correlation scores across different sizes of the validation set. Recall the equations of $\text{Inf}(D_i \rightarrow D_{\text{tst}})$ and $\text{Inf}(D_i \leftarrow D_{\text{tst}})$ from Section 1.

$$\text{Inf}(D_i \rightarrow D_{\text{tst}}) = \mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_{\text{tst}}) - \mathcal{L}(\mathcal{A}(D_{\text{trn}} \setminus D_i), D_{\text{tst}}).$$

On the other hand, the *test-to-train* influence characterized by **P2** can be written as

$$\text{Inf}(D_i \leftarrow D_{\text{tst}}) = \mathcal{L}(\mathcal{A}(D_{\text{trn}} \cup D_{\text{tst}}), D_i) - \mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_i).$$

In Section 2.1, when we describe our hypothesis under the noisy setting, we use a group of test points D_{tst} to calculate $\text{Inf}(D_i \leftarrow D_{\text{tst}})$ and $\text{Inf}(D_i \rightarrow D_{\text{tst}})$ as well as a group of training points (i.e., group-to-group influence). To evaluate the impact of the size of the validation set, we conduct experiments with varying validation set sizes while maintaining the other factors (e.g., mislabeled ratio, and hyperparameters).

Table 4 shows that enlarging the validation set size (e.g., from 50 to 1000) is advantageous across all datasets. The rationale behind this observation is that a larger validation set (i.e., $|D_{\text{tst}}|$) yields less sensitivity in the scoring of each term (i.e., $\mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_{\text{tst}})$ and $\mathcal{L}(\mathcal{A}(D_{\text{trn}} \setminus D_i), D_{\text{tst}})$ in $\text{Inf}(D_i \leftarrow D_{\text{tst}})$. In particular, if we have a larger validation set for $\text{Inf}(D_i \rightarrow D_{\text{tst}})$, we remove the sensitivity in the choice of validation samples, leading to accurately estimating the effect of each group. Additionally, introducing a larger number of clean samples (i.e., D_{tst}) to D_{trn} might trigger a larger difference between $\mathcal{L}(\mathcal{A}(D_{\text{trn}} \cup D_{\text{tst}}), D_i)$ and $\mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_i)$ due to the amplified negative effect. The second row of Figure 7 further illustrates that enlarging the validation set size enhances the impact of each group. In particular, as depicted by the steeper and smoother blue lines (i.e., $\mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_{\text{tst}})$) in the second of the figure, a larger clean validation set helps to mitigate the noise attributable to the stochastic nature of the learning process.

Analysis of correlation scores across different mislabeled ratios. In our analysis, we also consider the impact of different mislabeling ratios within the training dataset (D_{trn}). Table 5 underscores the importance of choosing an appropriate mislabeled ratio. This factor is crucial in mitigating stochasticity by amplifying the influence of each group when analyzing correlation scores in a noisy setting. Our empirical study indicates that a mislabeling ratio exceeding 30% tends to yield less noisy results since it amplifies meaningful signals, such as clear differences between groups. As depicted in the third row of Figure 7, a 20% mislabeling ratio is insufficient to reduce noise in score calculation (i.e., more fluctuation in the line of $\mathcal{L}(\mathcal{A}(D_{\text{trn}} \setminus D_i), D_{\text{tst}})$).

It is important to avoid excessively high mislabeling ratios, like over 50%, as they can adversely affect the learning process. For example, with too many mislabeled samples, a model struggles to be effectively trained on the dataset and tends to underfit. This situation makes it difficult to differentiate signals between each group because having many mislabeled samples across different groups may yield a high loss for $\mathcal{L}(\mathcal{A}(D_{\text{trn}}), D_i)$ and $\mathcal{L}(\mathcal{A}(D_{\text{trn}} \setminus D_i), D_{\text{tst}})$ as shown in the third row of Figure 7. This high loss in the initial stage may not only contain signals of each group's influence but also have additional noise that prevents one from magnifying the influence of each group.

C. Continual Learning vs. Training from Scratch [Section 3]

As we are considering the new objective of adding a test set D_{tst} to the training dataset D_{trn}, the model trained with the new objective will be $\hat{\theta}_{\varepsilon, D_{\text{tst}}} = \arg\min_\theta \mathcal{L}(\theta, D_{\text{trn}}) + \varepsilon \ell(\theta, D_{\text{tst}})$. Once $\hat{\theta}_{\varepsilon, D_{\text{tst}}}$ is obtained, one can evaluate the change in the loss of individual training points:

$$\texttt{Forward-INF}\ (D_i) = \mathcal{L}(\hat{\theta}_{\varepsilon, D_{\text{tst}}}, D_i) - \mathcal{L}(\hat{\theta}, D_i) \tag{9}$$

The main challenge here is to efficiently obtain $\hat{\theta}_{\varepsilon, D_{\text{tst}}}$ from $\hat{\theta}$. Note that for any θ that is close to $\hat{\theta}$, we have

$$\begin{aligned}
&\mathcal{L}(\theta, D_{\text{trn}}) + \varepsilon \ell(\theta, D_{\text{tst}}) \\
&\quad = \mathcal{L}(\hat{\theta}, D_{\text{trn}}) + \varepsilon \ell(\hat{\theta}, D_{\text{tst}}) \\
&\qquad + (\theta - \hat{\theta}) \cdot \nabla_\theta [\mathcal{L}(\theta, D_{\text{trn}}) + \varepsilon \ell(\theta, D_{\text{tst}})]|_{\theta = \hat{\theta}} \\
&\qquad + \mathcal{O}(\|\theta - \hat{\theta}\|_2^2) \\
&\quad \approx \mathcal{L}(\hat{\theta}, D_{\text{trn}}) + \varepsilon \ell(\hat{\theta}, D_{\text{tst}}) \\
&\qquad + (\theta - \hat{\theta}) \cdot [\nabla_\theta \mathcal{L}(\theta, D_{\text{trn}}) + \varepsilon \nabla_\theta \ell(\theta, D_{\text{tst}})]|_{\theta = \hat{\theta}}
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\quad \approx \mathcal{L}(\hat{\theta}, D_{\text{trn}}) + \varepsilon \ell(\hat{\theta}, D_{\text{tst}}) \\
&\qquad + (\theta - \hat{\theta}) \cdot \varepsilon \ \nabla_\theta \ell(\theta, D_{\text{tst}})|_{\theta = \hat{\theta}}
\end{aligned} \tag{11}$$

where the first approximation holds for $\varepsilon \rightarrow 0$ and *continuity* of the loss function and the second approximation holds for $\nabla_\theta \mathcal{L}(\theta, D_{\text{trn}})|_{\theta = \hat{\theta}} = 0$ for $\hat{\theta}$ being the minimizer of $\ell(\theta, D_{\text{trn}})$ by definition. Taking the $\arg\min$ on both sides of (10) yields

$$\begin{aligned}
\hat{\theta}_{\varepsilon, D_{\text{tst}}} &= \arg\min_\theta [\mathcal{L}(\theta, D_{\text{trn}}) + \varepsilon \ell(\theta, D_{\text{tst}})] \\
&\approx \arg\min_\theta \Big[\mathcal{L}(\hat{\theta}, D_{\text{trn}}) + \varepsilon \ell(\hat{\theta}, D_{\text{tst}}) \\
&\qquad + (\theta - \hat{\theta}) \cdot \varepsilon \ \nabla_\theta \ell(\theta, D_{\text{tst}})|_{\theta = \hat{\theta}} \Big] \qquad (12) \\
&= \arg\min_\theta (\theta - \hat{\theta}) \cdot \varepsilon \ \nabla_\theta \ell(\theta, D_{\text{tst}})|_{\theta = \hat{\theta}} \qquad (13)
\end{aligned}$$

Note that to find the minimum of (12), one needs to search along $\nabla_\theta \ell(\theta, D_{\text{tst}})|_{\theta = \hat{\theta}}$, i.e., the current gradient direction at the test sample. Thus, to find $\hat{\theta}_{\varepsilon, D_{\text{tst}}}$, given that $\epsilon \rightarrow 0$ and $(\hat{\theta}_{\varepsilon, D_{\text{tst}}} - \hat{\theta})$ is small assuming *continuity* of the loss function, *one only needs to continually update the trained model $\hat{\theta}$ on the given test sample D_{tst}.*

In standard practice, ε is often considered to be positive as it represents the model being *trained* on the test sample (i.e., $\varepsilon \rightarrow 0^+$). However, if the test sample is drawn from a similar distribution as the training data (i.e., $\nabla_\theta \ell(\theta, D_{\text{tst}})|_{\theta = \hat{\theta}}$, the magnitude of the gradient is small. Interestingly, and counter-intuitively, we can estimate data influence effectively with impressive accuracy by setting ε to be negative and employing the gradient ascent on the test sample. Namely, we define the following mirrored metric

$$\text{Forward-INF}(D_i) := \left. \frac{\ell(\hat{\theta}_{\varepsilon, D_{\text{tst}}}, D_i) - \ell(\hat{\theta}, D_i)}{\varepsilon} \right|_{\varepsilon \rightarrow 0^-} \tag{14}$$

This technique successfully circumvents numerical issues from diminished gradients on well-trained models and remarkably enhances the accuracy of influence estimation.

D. Hyperparameter Details [Section 4]

As our approach is based on a gradient ascent (i.e., maximization), selecting appropriate hyperparameters (e.g., learning rate, the number of epochs) is essential. We note that the details of the experiment setting for Section 2.1 are elaborated in Section B. In this section, we mainly focus on presenting the details of application experiments from Section 4.

Data influence estimation in diffusion models [Section 4.1]. In this experiment, it is necessary to fine-tune a stable diffusion pre-trained model on a set of test samples. To fine-tune the pre-trained stable diffusion model, we follow the same setting as that from the previous work [38]. In particular, we randomly pick fine-tuning exemplars from ImageNet [7] and further train the pre-trained stable diffusion model on the selected exemplars with 5 iterations, a learning rate of 1e-5, and a batch size of 4. After generating the synthesized samples, we perform gradient ascent on a set of synthesized samples with the same learning rate and 3 iterations.

Data leakage detection [Section 4.2]. We train both ResNet-18 and ResNet-50 using the Adam optimizer with a learning rate of 0.01 and 200 iterations for both CIFAR-10 and CIFAR-100 [23]. Due to the computational complexity of our baselines (`IF`, `TracIN`), we randomly selected 20 test samples for evaluation. Within this study, we first assign 20% of the total test samples specifically for the purpose of hyperparameter selection, while the remaining portion is dedicated to evaluation.

Model behavior tracing [section 4.5]. In this study, we leverage the FTRACE-TREX dataset [3] for the model behavior tracing task. The FTRACE-TREX dataset consists of a set of “abstracts” and a set of “queries”, and each query is annotated with the corresponding list of fact samples [3]. The training set of FTRACE-TREX is sourced from the TREX dataset [9], and the test set of the FTRACE-TREx dataset is derived from the LAMA dataset [28]. Every training example that conveys the identical information as the given test example is designated as a ”proponent.” We randomly sampled 100 data points for this task and took an average of three repeated experiments.

Model/Dataset	Metric	`IF-100`	`Forward-INF`
ResNet-18	T-1	0.000	**0.880**
ImageNet-100	T-5	0.000	**0.880**

Table 6. Data Leakage detection performance comparison of ResNet-18 trained on the ImageNet-100 dataset.

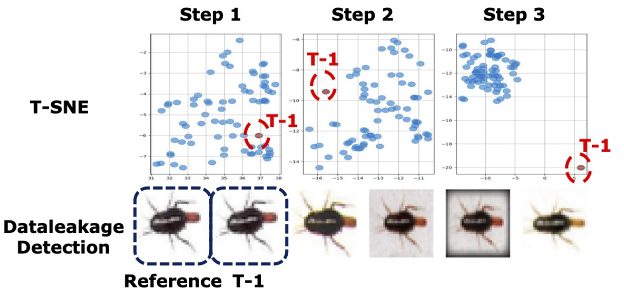


Figure 8. Gradient ascent dynamics on T-SNE feature space (top), and top-5 relevant samples retrieved by `Forward-INF` (bottom). The reference indicates the test sample we want to unlearn and T-1 denotes the top-1 sample retrieved by `Forward-INF` (bottom) in the T-SNE [35] feature space (top). We can observe that after a few gradient ascent steps, the top-1 sample begins to migrate away from the center of the feature cluster. This is crucial for effectively detecting duplicated samples.

E. Further Analysis and Details

E.1. Analysis of Correlation between `Forward-INF` (D_i) and $\text{Inf}(D_i \leftarrow D_{\text{tst}})$

Validation Set Size		**50**	**100**	**500**
MNIST	Pearson	0.9701	0.9819	0.9951
	Spearman	0.9746	0.9786	0.9942
FMNIST	Pearson	0.9450	0.9811	0.9942
	Spearman	0.9479	0.9795	0.9920
CIFAR10	Pearson	0.9050	0.9528	0.9862
	Spearman	0.9052	0.9413	0.9832

Table 7. Correlation scores on between `Forward-INF` (D_i) and $\text{Inf}(D_i \leftarrow D_{\text{tst}})$. We keep the same mislabeled ratio (i.e., 0.5) while changing the sizes of the validation set from 50 to 500.

We want to demonstrate that our influence score approximator, `Forward-INF` , can effectively estimate the influence score of $\text{Inf}(D_i \leftarrow D_{\text{tst}})$, thereby obviating the need for re-training since it will be the equivalent influence information as $\text{Inf}(D_i \rightarrow D_{\text{tst}})$. For this experiment, we chose the noisy setting and utilized the same groups as in previous tests under the noisy setting. We performed gradient

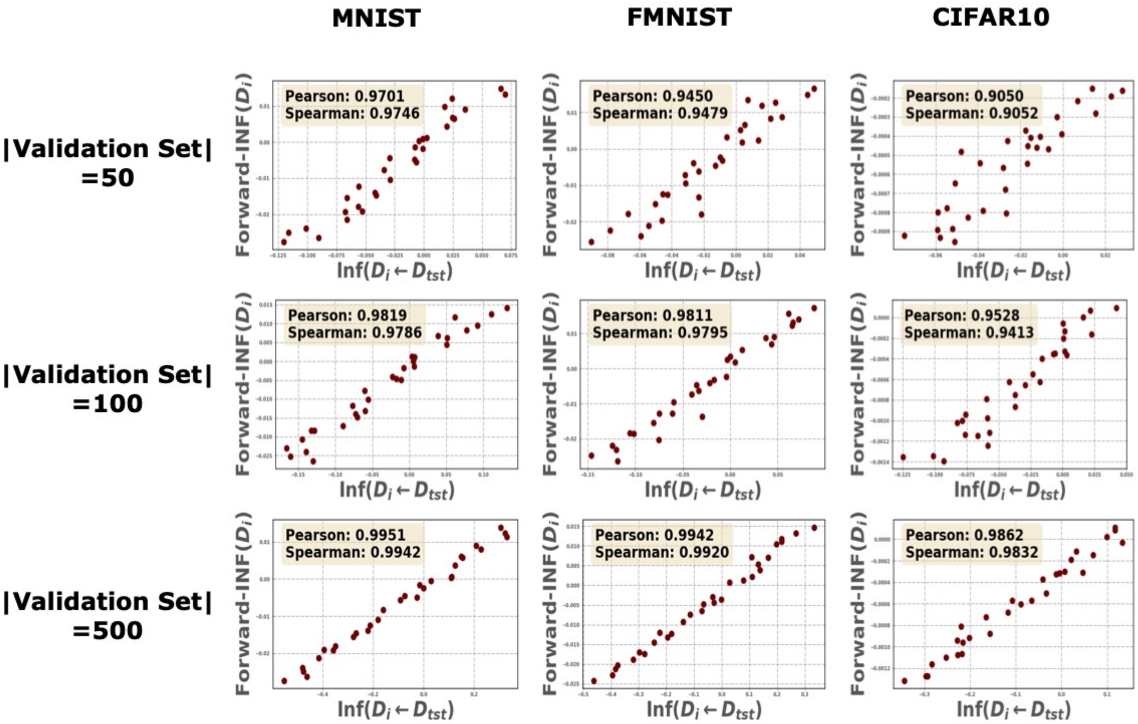


Figure 9. Correlation scores between the estimated influence scores from Forward-INF (i.e., Forward-INF (D_i)) and scores from $\text{Inf}(D_i \leftarrow D_{\text{tst}})$ across different datasets with different sizes of validation sets.

ascent with respect to a group of test samples D_{tst} and then computed the loss difference between the original and the unlearned models for different groups. We used the SGD optimizer for gradient ascent with a learning rate of 0.01, a weight decay of 0.001, and two iterations.

Table 7 presents the correlation scores between Forward-INF and $\text{Inf}(D_i \leftarrow D_{\text{tst}})$. As indicated in the table, there is a consistently high correlation between our proposed method and $\text{Inf}(D_i \leftarrow D_{\text{tst}})$. Similarly, we observe that increasing the size of the validation set results in higher correlation scores, attributed to reduced noise in the learning process. We provide the correlation plots in Figure 9.

E.2. Data Influence Estimation in Diffusion Model [Section 4.1]

In this section, we present both quantitative and qualitative results of data influence estimation for diffusion models. As delineated in [38], since the fine-tuning examples can serve as the noisy ground truth, we can quantitatively measure whether Forward-INF can identify the ground-truth sample as the most influential sample (Top-1). In our experiment, we randomly sample 20 different objects from the ImageNet and also construct the different sizes (e.g., 100, 1K, 10K, 20K) of candidate sets. We follow the same candidate set creation process as we described in Section 4.1. If Forward-INF correctly identifies the ground truth as the most influential point, we mark it as success and average over 20 points. The detection results are presented in Table 8. Our observations reveal that Forward-INF correctly identifies the ground truth fine-tuning examples regardless of its candidate set sizes, demonstrating its robustness across varying candidate set sizes.

In addition, we provide the qualitative results in Figure 10. In this experiment, we utilize a candidate set comprising 20K samples and present the top 8 highest influential training samples retrieved by Forward-INF . Most of the retrieved samples share similar features with each other either on an image or caption side. This indicates that the unlearning process mostly affects the samples that share similar features. Given that the candidate set is curated selectively rather than utilizing the entire dataset, there's a possibility that it might not include more than top-k captions precisely identical to those of the ground truth samples. In this case, Forward-INF might yield some samples that are not directly relevant. However, the crucial observation is that, despite the presence of candidate samples exhibiting similar features to the ground truth in both image and caption aspects, Forward-INF can accurately identify the ground truth as the most influential training data, highlighting its efficacy in pinpointing the key training influences in the generation of synthesized images.

Applying other influence approximators, such as IF and TracIn, to large-scale diffusion models poses a challenge in terms of efficiency, as delineated in [38]. Specifically, the IF needs to approximate the inverse Hessian matrix as well as calculate gradients with respect to training and test samples. However, this approach becomes problematic when it comes to the large-scale stable diffusion model (e.g., around 860 million parameters) and a large amount of training data used (e.g., around 2 billion samples) [38]. On the other hand, TracIn requires computing the dot product between gradients with respect to every training and test sample, leading to the scalability problem for large-scale models. Therefore, as also mentioned in a previous study [3], for the large-scale language model, one selects the first layer, and for the classification model, the last layer has been widely selected. *However, both baselines have not been widely explored and applied to the diffusion models to achieve high efficiency and performance*. Therefore, addressing these scalability challenges within such baselines will be deferred to future research.

Candidate Set Size	**100**	**1K**	**10K**	**20K**
T-1 Detection	1.000	1.000	1.000	1.000

Table 8. Top-1 detection accuracy for identifying the ground truth influential sample within candidate sets of varying sizes for the stable diffusion model. Forward-INF shows 100% detection performance across different candidate set sizes.

E.3. Data Leakage Detection [Section 4.2]

We further extend our experiment of data leakage detection on ImageNet-100 to validate the practicality as well as scalability. We observe a high detection rate (e.g., 88% top-1 detection rate) for the RN18 classifier trained on the ImageNet-100. As the model complexity increases, the detection rate slightly decreases, considering the previous results on CIFAR-10 and CIFAR-100. We hypothesize that the unlearning process affects more diverse neurons and their connections when larger and more complex datasets are used, leading to a drop in performance. However, this performance is still considered high, compared with our baseline such as IF which completely fails as shown in Table 6.

Unlearning dynamics. We additionally conduct a feature embedding analysis to visualize the gradient ascent dynamics in our method. In Figure 8, we present the feature space change according to the maximization steps. Interestingly, we observe that the duplicated sample (i.e., top-1) is strongly impacted by the gradient ascent process, experiencing a deviation from the class cluster during the process of unlearning. The figure in the bottom row shows examples that receive top-5 scores out of all training samples, indicating that our approach correctly identifies both the exact duplicated sample as well as relevant near duplicates. This observation supports our intuition that if we unlearn one specific point, we could expect that class or sample-related points would receive a high training loss change, thereby receiving a high Forward-INF score.

Senstivity analysis of hyperparameter selection. Considering that the unlearning process may involve sensitivity to hyperparameters (e.g., learning rate, and the number of iterations), our method initially selects the hyperparameter based on a holdout validation set. Consequently, we also present the results concerning the sensitivity of the validation set size. As illustrated in Table 9, Forward-INF still achieves 100% detection accuracy for our approach even with only 25 validation samples, indicating that our approach exhibits low sensitivity to changes in the size of the validation set.

E.4. Memorization Analysis [Section 4.3]

We provide extensive qualitative results for memorization analysis in Figure 11. As we can observe from the figure, our Forward-INF can effectively retrieve the most influential point with a much lower cost than the previous work [11]. In addition, based on qualitative results, we consistently observe that the high-influential pairs are exact or near duplicated samples and benefit the most from memorization (see the corresponding memorization score from the figure).

E.5. Model Behavior Tracing [Section 4.5]

In this section, we delve deeper into the problem of model behavior tracing, expanding the scope of our discussion. In reality, it becomes inherently challenging to locate direct supporting samples from the training data that precisely correspond to a given test sample. Instead, the relationships between training samples and generating answers for the test example often involve more indirect correlations. In light of this, we introduce an additional experimental design to explore a scenario wherein the test query undergoes paraphrasing. The key question is whether we can still identify the relevant ground-truth samples from the training data.

As demonstrated in Table 10, our proposed approach, Forward-INF , consistently displays favorable performance, albeit with a minor drop, while outperforming the baseline TracIn in terms of MRR and Precision metrics for the 15K candidate set size.

Comparison with simple model-independent information retrieval baseline [30]. BM25 is a standard information retrieval technique that selects proponents by retrieving training examples with high lexical overlap with the query.

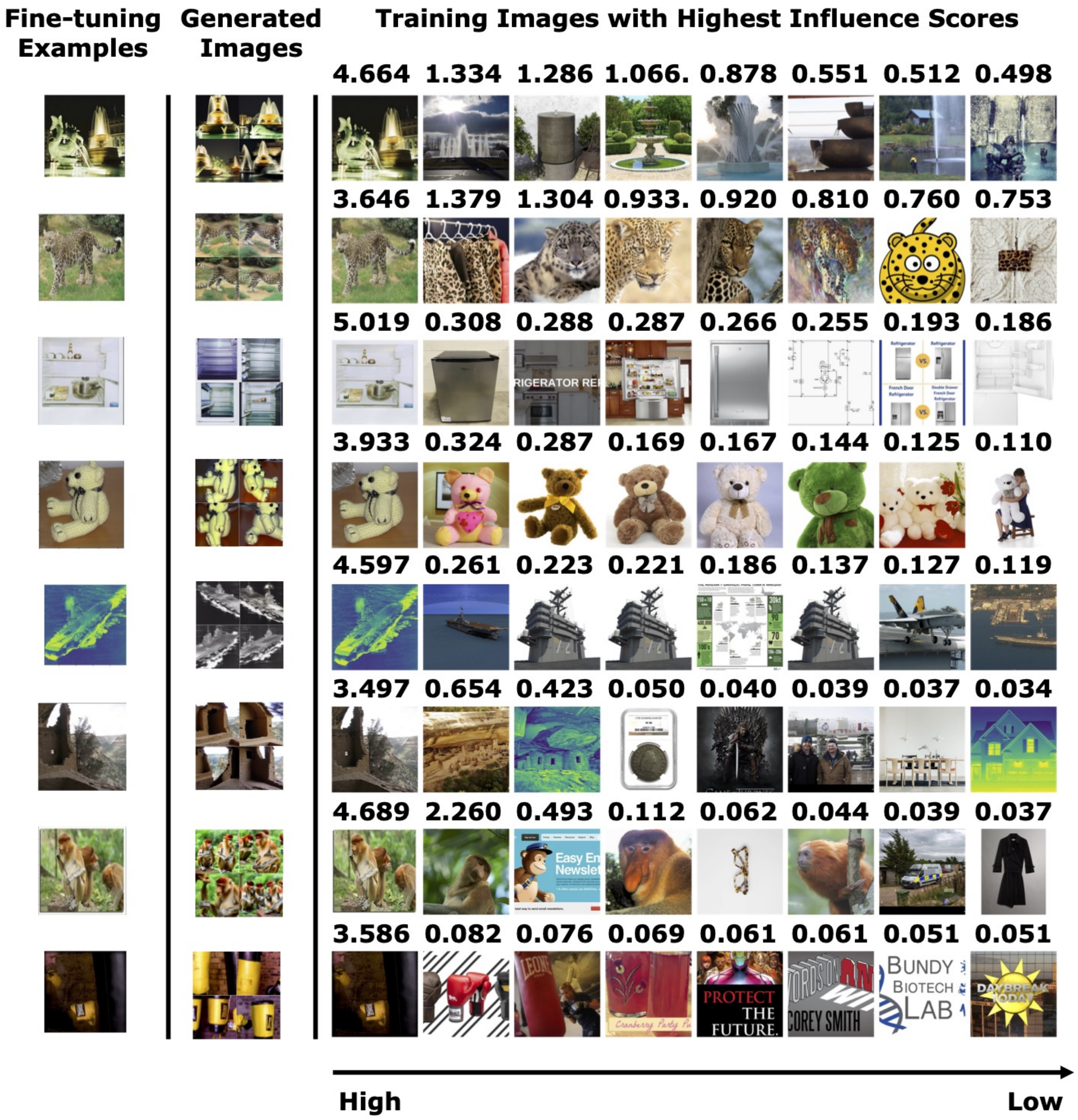


Figure 10. Extensive qualitative results of data influence estimation for the diffusion model. Similar to Figure 3, we provide the ground truth (first column), synthesized images after fine-tuning (second column), and training images with the highest influence scores (the remaining columns). Forward-INF can accurately identify the ground truth fine-tuning examples as the most influential data samples contributed to the generation of new images.

Model	Dataset	Metric	5% [25 samples]	10% [50 samples]	20% [100 samples]	30% [150 samples]
ResNet-18	CIFAR-100	T-1	1.000	1.000	1.000	1.000
ResNet-18	CIFAR-100	T-5	1.000	1.000	1.000	1.000

Table 9. Data leakage detection performance using ResNet-18 across various validation set sizes for the sensitivity analysis of hyperparameter selection. Forward-INF consistently delivers high performance, even with small validation set sizes.

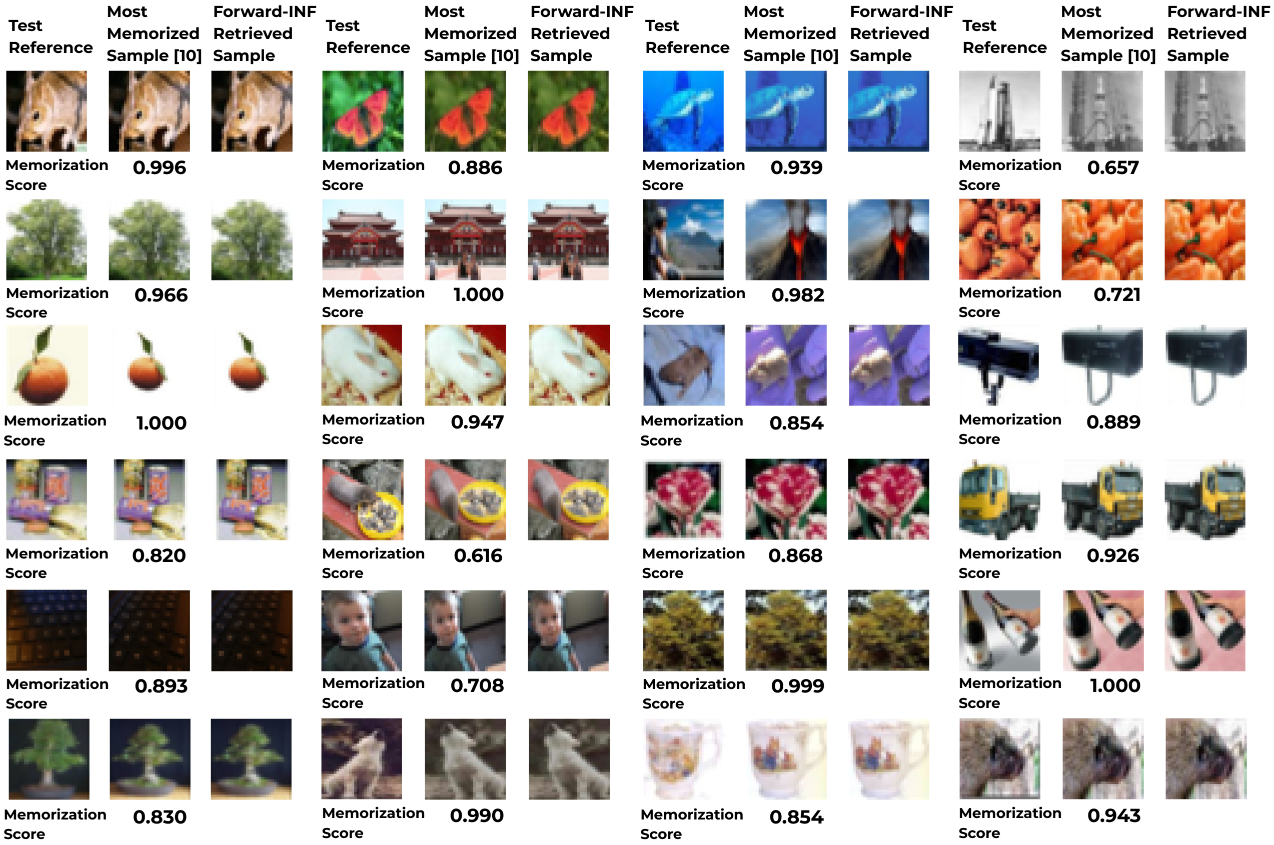


Figure 11. Extensive results on memorization experiment. Similar to Figure 4, we present the most influential samples provided by [11], and samples that were retrieved by `Forward-INF` .

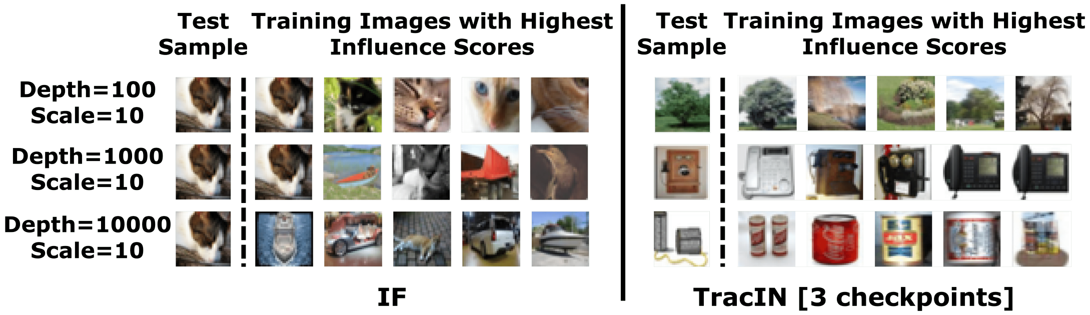
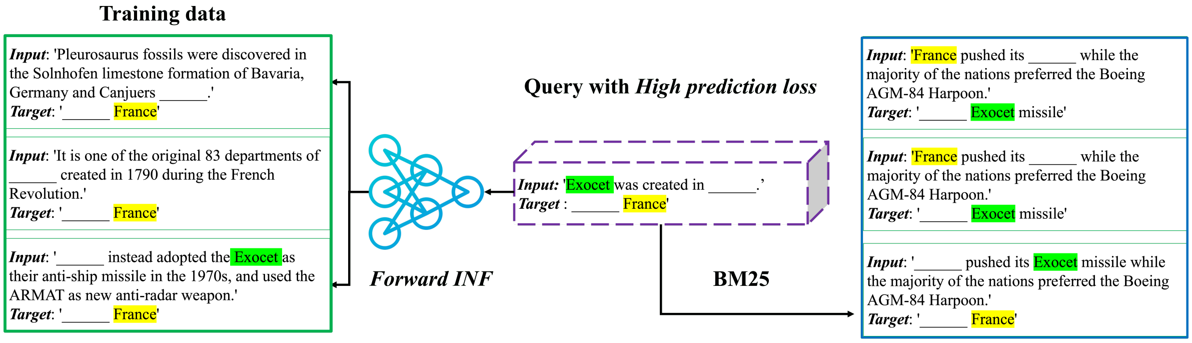


Figure 12. `TracIn` and `IF` results on data leakage experiments.The right side of the figure illustrates that even though TracIn can often identify visually similar samples to the test example, it fails to detect the ground-truth leaked sample. The left side of the figure showcases the results using different depths and scales to empirically demonstrate why IF-10000 falls short.

BM5 has been adopted to trace facts in [3]. However, since the influence scores derived from BM25 do not incorporate information pertinent to the specific model under consideration, it *cannot* be used to explain why a *given model* makes some factual assertions. For instance, when our LLM produces an incorrect prediction resulting in a high loss, we want to trace back to the contributing examples that lead to this misjudgment for the sake of transparency and interpretability. While the performance of BM25 in identifying pertinent facts is promising [3], it falls short in shedding light on the intricate model's behavior (i.e., it only provides the related facts but does not explain why a model outputs high loss). Conversely, our proposed method is adept at elucidating the underlying reasons behind LLM's inaccurate answers. For example, as shown in Figure 13, the top-1 pair retrieved by our method includes the target label

Candidate Set Size	15k				Inspected Queries
Metric	**MRR**	**Δ**	**Precision**	**Δ**	**# of Queries/Min**
`TracIn` (single)	0.1868	-	0.1549	-	396.125
`Forward-INF`	**0.2031**	**0.0163**	**0.1551**	**0.0002**	**1289.475**

Table 10. Behavior tracing performance comparison of different attribution methods on the paraphrased candidate set.



Input: 'Pleurosaurus fossils were discovered in the Solnhofen limestone formation of Bavaria, Germany and Canjuers ______.'
Target: '______ France'

Input: 'It is one of the original 83 departments of ______ created in 1790 during the French Revolution.'
Target: '______ France'

Input: '______ instead adopted the Exocet as their anti-ship missile in the 1970s, and used the ARMAT as new anti-radar weapon.'
Target: '______ France'

Input: 'Exocet was created in ______.'
Target : ______ France'

Input: 'France pushed its ______ while the majority of the nations preferred the Boeing AGM-84 Harpoon.'
Target: '______ Exocet missile'

Input: 'France pushed its ______ while the majority of the nations preferred the Boeing AGM-84 Harpoon.'
Target: '______ Exocet missile'

Input: '______ pushed its Exocet missile while the majority of the nations preferred the Boeing AGM-84 Harpoon.'
Target: '______ France'

Figure 13. Comparison of the top-3 selected training samples from `Forward-INF` and BM25.

"France." However, when considering the overall meaning of the sentence, it indicates a different semantical meaning. This observation implies the presence of semantic conflicts (i.e., some samples share the same labels but entail different semantical meanings) within the training samples, consequently confusing the model's learning process.