Interpretability through Training Samples: Data Attribution for Diffusion Models

Tong Xie[*1] TONGXIE@UCLA.EDU
Haoyu Li[*1] HAOYULI02@UCLA.EDU
Andrew Bai[2] ANDREWBAI@CS.UCLA.EDU
Cho-Jui Hsieh[2] CHOHSIEH@CS.UCLA.EDU
[1] *Department of Mathematics;* [2] *Department of Computer Science*
University of California, Los Angeles

Abstract

Data attribution methods trace model behavior back to its training dataset, offering an effective approach to better understand "black-box" neural networks. While prior research has established quantifiable links between model output and training data in diverse settings, interpreting diffusion model outputs in relation to training samples remains underexplored. In particular, diffusion models operate over a sequence of timesteps instead of instantaneous input-output relationships in previous contexts, posing a significant challenge to extend existing frameworks to diffusion models directly. Notably, we present Diffusion-TracIn that incorporates this temporal dynamics and observe that timesteps induce a prominent bias in influence estimation, where certain training samples emerge as generally influential due to dominating gradient norms. To mitigate this effect, we introduce Diffusion-ReTrac as an adaptation that enables the retrieval of training samples more specific to the test sample of interest, facilitating a localized measurement of influence and considerably more intuitive visualization. We demonstrate the efficacy of our approach through various evaluation metrics and auxiliary tasks.

Keywords: Diffusion Models, Interpretability, Data Attribution, Influence

1. Introduction

Deep neural networks have emerged to be powerful tools for the modeling of complex data distributions and intricate representation learning. However, their astounding performance often comes at the cost of interpretability, leading to an increasing research interest to better explain these "black-box" methods. Instance-based interpretation is one approach to explain why a given machine learning model makes certain predictions by tracing the output back to training samples. While these methods have been widely studied in supervised tasks and demonstrated to perform well (Kong and Chaudhuri, 2021; Pruthi et al., 2020; Yeh et al., 2018), there is limited exploration of their application in unsupervised settings, especially for generative models (Kingma and Welling, 2013; Goodfellow et al., 2020). In particular, diffusion models represent a state-of-the-art advancement in generative models and demonstrate remarkable performance in a variety of applications such as image generation, text generation, and audio synthesis (Kong et al., 2020; Ho et al., 2022; Li et al., 2022). The prevailing generative agents in creative arts such as Stable Diffusion (Rombach

∗. Equal contribution

Preprint.

et al., 2022) also call for fair attribution methods to acknowledge the training data contributors. Nonetheless, the interpretability and attribution of diffusion models remain an under-explored area (Terashita et al., 2021; Kong and Chaudhuri, 2021; Georgiev et al., 2023; Dai and Gifford, 2023).

Compared to traditional supervised settings, the direct extension of instance-based interpretation to diffusion models is challenging due to the following factors. First, the diffusion objective involves an expectation over the injected noise $\epsilon \sim \mathcal{N}(0, I)$, hence a precise computation is impractical. Second, diffusion models operate over a sequence of timesteps instead of instantaneous input-output relationships. Although each timestep is weighted equally during the training process, we observe that certain timesteps can exhibit the *dominating gradient norm effect*. This means the gradient of the diffusion loss function with respect to model parameters is dominantly large relative to all other timesteps. As most instance-based explanation models utilize this first-order gradient information, such biased gradient norms can propagate its domination into the influence estimation for diffusion models. In practice, specifically, timesteps are often uniformly sampled during training. Therefore a training sample that happens to be trained on earlier timesteps may exhibit higher-than-usual gradient norms, and thus be characterized as "generally influential" to various completely different test samples.

We present Diffusion-TracIn and Diffusion-ReTrac to demonstrate and address the existing difficulties. Diffusion-TracIn is a designed extension of TracIn (Pruthi et al., 2020) to diffusion models that incorporates the denoising timestep trajectory. This approach showcases instances where influence estimation is biased. Subsequently, we introduce Diffusion-ReTrac as a re-normalization of Diffusion-TracIn to alleviate the dominating-norm effect.

Our contributions are summarized as follows:

1. Propose Diffusion-TracIn as a designed extension to diffusion models that incorporate the timestep dynamics.
2. Identify and investigate the timestep-induced gradient norm bias in diffusion models, providing preliminary insights into its impact on influence estimation.
3. Introduce Diffusion-ReTrac to mitigate the timestep-induced bias, offering fairer and targeted data attribution.
4. Illustrate and compare the effectiveness of the proposed approach on auxiliary tasks.

2. Related Work

Data attribution methods trace model interpretability back to the training dataset, aiming to answer the following counterfactual question: *which training samples are most responsible for shaping model behavior?*

2.1 Influence Estimations

Influence functions quantify the importance of a training sample by estimating the effect induced when the sample of interest is removed from the training (Koh and Liang, 2017). This method involves inverting the Hessian of loss, which is computationally intensive and

can be fragile in highly non-convex deep neural networks (Basu et al., 2020). Representer Point is another technique that computes influence using the representer theorem, yet also relies on the assumption that attribution can be approximated by the final layer of neural networks, which may not hold in practice (Yeh et al., 2018). For diffusion models, the application of influence functions is significantly hindered by its computational expense while extending the representer point method is ambiguous due to the lack of a natural "final layer" in diffusion models. Pruthi et al. (Pruthi et al., 2020) introduced TracIn to measure influence based on first-order gradient approximation that does not rely on optimality conditions.

In this paper, we extend the TracIn framework to propose an instance-based interpretation method specific to the diffusion model architecture. For a fairer attribution, we present Diffusion-ReTrac that re-normalizes the gradient information to mitigate bias. Previous works have utilized this re-normalization technique to enhance influence estimator performance in supervised settings (Barshan et al., 2020; Hammoudeh and Lowd, 2022). Barshan et al.. reweight influence function estimations using optimization objectives that place constraints on global influence, enabling the retrieval of explanatory examples more localized to model predictions (Barshan et al., 2020). Gradient aggregated similarity (GAS) leverages re-normallization to better identify adversarial instances (Hammoudeh and Lowd, 2022). These works align well with our studies in understanding the localized impact of training instances on model behavior.

2.2 Influence in Unsupervised Settings

The aforementioned methods address the counterfactual question in supervised settings, where model behavior may be characterized in terms of model prediction and accuracy. However, extending this framework to unsupervised settings is non-trivial due to the lack of labels or ground truth. Prior works explore this topic and approach to compute influence for generative adversarial networks (GAN) (Terashita et al., 2021) and variational autoencoders (VAE) (Kong and Chaudhuri, 2021). Concurrent work quantifies influence in diffusion models through the use of ensembles, which requires training multiple models with subsets of the training dataset, making it unsuitable for naturally trained diffusion models (Dai and Gifford, 2023). Previous work in Data attribution method Journey TRAK for diffusion models applies TRAK (Park et al., 2023) to diffusion models and attributes each denoising timestep individually (Georgiev et al., 2023), which is less interpretable since the diffusion trajectory spans multiple timesteps and a single-shot attribution is more holistic. These works are complementary to our studies and contribute to a more comprehensive understanding of instance-based interpretation methods in unsupervised settings.

2.3 Areas of Application

Data attribution methods prove valuable across a wide range of domains, such as outlier detection, data cleaning, machine unlearning, and memorization analysis (Feldman, 2020; Campbell, 1978). The adoption of diffusion models in artistic pursuits, such as Stable Diffusion (Rombach et al., 2022) and its variants (Zhang et al., 2023), has also gained

substantial influence. This then calls for fair attribution methods to acknowledge and credit artists whose works have shaped these models' training. Such methods also become crucial for conducting further analyses related to legal and privacy concerns (Carlini et al., 2023; Somepalli et al., 2023).

3. Preliminaries

3.1 Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) Ho et al. (2020) are a special type of generative models that parameterized the data as $p_\theta(x_0) = \int p_\theta(x_{0:T})\, dx_{1:T}$, where $x_1, \ldots, x_T$ are latent variables of the same dimension as the input. The inner term $p_\theta(x_{0:T})$ is the reverse process starting at the standard normal $p(x_T) = \mathcal{N}(x_T; 0, I)$, which is defined by a Markov chain:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{1}$$

where $p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$.The reverse process is learned to approximate the forward process $q(x_{1:T}|x_0)$, which is fixed to a Markov Chain based on a variance scheduler $\beta_1, \ldots, \beta_T$:

$$q_\theta(x_{1:T}|x_0) = \prod_{t=1}^{T} q_\theta(x_t|x_{t-1}) \tag{2}$$

where $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}, \beta_t I)$. Being conditioned on the clean image, one notable property of the forward process is that each sample x_t at timestep t can be sampled directly from the knowledge of x_0, independently from the previous timesteps. The distribution of x_t is given as follows,

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I) \tag{3}$$

where $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$. Therefore, efficient training can be achieved by stochastically selecting timesteps for each sample. DDPM further simplifies the loss by re-weighting each timestep, leading to the training objective used in practice

$$L_{\text{simple}}(\theta) = \mathbb{E}_{x_0,t,\epsilon}[d(\epsilon, \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{(1-\alpha_t)}\epsilon, t)] \tag{4}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and d can be chosen as $l1$ or $l2$ distance. From this objective, we notice that it is possible to treat the objective of diffusion as a combination of T loss functions. If we denote $L_t(\theta, \epsilon, x_0) = L_{\text{simple}}(\theta, \epsilon, x_0, t) := d(\epsilon, \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{(1-\alpha_t)}\epsilon, t)$ to be a distinct loss function on each timestep t and noise, we can treat L_{simple} as an expectation over all the L_t.

3.2 TracIn

TracIn is proposed as an efficient approximation of the traditional leave-one-out influence. It defines the idealized version of influence of a training sample z to a test sample z' as the total reduction of loss on z' when the model is trained on z. Let k denote each occurrence

of updating the model with the training sample z. Let w_k denote the model parameters after training the model on z for the k^{th} time. Formally,

$$\text{Ideal-Influence}(z, z') = \sum_{k:z_k=z} \ell(w_k, z') - \ell(w_{k+1}, z'). \tag{5}$$

To accelerate the computation, the change in the loss of the test sample can be approximated by a Taylor expansion

$$\ell(w_{k+1}, z') - \ell(w_k, z') = \nabla\ell(w_k, z') \cdot (w_{k+1} - w_k) + \mathcal{O}(\|w_{k+1} - w_k\|^2). \tag{6}$$

Suppose the optimizer used is Stochastic Gradient Descent (SGD), the model parameter update term can be calculated by $w_{k+1} - w_k = -\eta_t \nabla\ell(w_k, z_k)$. Therefore, the first-order approximation of the ideal influence can be computed as

$$\text{TracIn}(z, z') = \sum_{k:z_k=z} \eta_k \nabla\ell(w_k, z') \cdot \nabla\ell(w_k, z). \tag{7}$$

In practice to accelerate the influence calculation, instead of tracking and summing up the gradient updates over every single occurrence of training on z, we approximate the influence with saved checkpoints during training. A training sample that has positive influence over the test sample is called the proponent, and opponent otherwise.

4. Method

4.1 Diffusion-TracIn

In this section, we present Diffusion-TracIn, which provides an efficient extension of TracIn designed specially for Diffusion Models. To make this extension, two adjustments keen to diffusion models need to be adjusted. First, the diffusion objective is an expectation of denoising losses over different timesteps t. Second, the objective involves an expectation over the added noise $\epsilon \sim \mathcal{N}(0, I)$. To address these challenges, we first apply TracIn conditioned on each timestep t and we compute a Monte Carlo average over m randomly sampled noises ϵ. We compute a TracIn-Score for each of the timestep t

$$\begin{aligned} \text{TracIn}(z, z', t) &:= \mathbb{E}_\epsilon\Big(\sum_{k:z_k=z} \eta_k \nabla_\theta L_t(\theta_k, \epsilon, z') \cdot L \Big) \\ &\approx \frac{1}{m} \sum_{i=1}^{m} \sum_{k:z_k=z} \eta_k \nabla_\theta L_t(\theta_k, \epsilon_i, z') \cdot L \end{aligned} \tag{8}$$

where $L = \nabla_\theta L_{t_{\text{train}}}(\theta_k, \epsilon_{\text{train}}, z)$ and t_{train} and ϵ_{train} represents the timesteps and noise used to train the sample z to fit the recover the true training dynamics. Then we define

Diffusion-TracIn to be the expectation over T timesteps to cover the full diffusion process.

$$\begin{aligned}\text{Diffusion-TracIn}(z, z') &:= \mathbb{E}_t(\text{TracIn}(z, z', t)) \\ &= \frac{1}{T}\sum_{t=1}^{T}\text{TracIn}(z, z', t) \\ &= \frac{1}{T}\sum_{t=1}^{T}\frac{1}{m}\sum_{i=1}^{m}\sum_{k:z_k=z}\eta_k\nabla_\theta L_t(\theta_k, \epsilon_i, z')\cdot L \\ &= \sum_{k:z_k=z}\eta_k(\frac{1}{mT}\sum_{t=1}^{T}\sum_{i=1}^{m}\nabla_\theta L_t(\theta_k, \epsilon_i, z'))\cdot L\end{aligned} \tag{9}$$

The practical version of Diffusion-TracIn also employs the ckpt as also employed by TracIn Pruthi et al. (2020).

$$\text{Diffusion-TracIn-CP}(z, z') := \sum_{k=1}^{s}\eta_k(\frac{1}{mT}\sum_{t=1}^{T}\sum_{i=1}^{m}\nabla_\theta L_t(\theta_k, \epsilon_i, z'))\cdot L \tag{10}$$

where s is the number of checkpoints. As the number of timesteps T can be large (e.g 1000) in practice, the practical implementation of Diffusion-TracIn employs n evenly spaced timesteps ranging from 1 to T. We observe empirically that evenly spaced timesteps yield good approximation over the full diffusion processes.

4.2 Diffusion-ReTrac

Although Diffusion-TracIn is a direct extension of TracIn for diffusion models derived from the mathematical definition, we discover the *dominating loss gradient norm effect* which can lead to bias in influence estimation. It can be noticed from Equation 7 that

$$|\text{TracIn}(z, z')| \leq \sum \eta_t\|\nabla\ell(w_t, z')\|\|\nabla\ell(w_t, z)\|$$

by Cauchy-Schwarz inequality. Hence, training samples with disproportionately large gradient norms are more likely to be considered either the most or least influential samples to the given test sample z', depending on the direction alignment of $\nabla l(w_t, z')$ and $\nabla l(w_t, z)$. In most machine learning models, a dominating gradient norm can be largely attributed to the training sample itself. For example, outliers and samples near the decision boundary may exhibit higher gradient norms than usual.

However, while sample-induced variance in gradient norms is informative for influence estimation, we demonstrate that the variance in gradient norms for diffusion models can also be an artifact of the diffusion training dynamics. In particular, empirical results shows that the loss function component from certain timesteps is more likely to have a larger gradient norm as shown by figure.

In other words, if during the training process, the stochastically chosen timestep for a training sample z happens to fall within (or close to) the later timesteps, then z will exhibit a biased larger norm that propagates into the influence calculation. Since the natural training

of diffusion models sample timesteps randomly for each z, different degrees of "timestep-induced" norm biases are introduced, leading to unfair influence estimates across training samples.

Since the gradient norm is not solely a property attributed to the sample but rather also caused by the norm bias inherent to diffusion models, an ideal instance-based interpretation should not overestimate the influence of samples with large norms and penalize those with small norms. To this end, we propose Diffusion-ReTrac which introduces normalization that reweights the training samples to address the "dominating-norm"effect.

In fact, this dominating norm effect can be introduced by the choice of timesteps for both test sample z and each training sample z', whose loss gradient norms are computed using timestep t and t_{train} respectively. For test sample z, the gradient information $\sum_{i=1}^{T} \nabla_\theta L_{\text{simple}}(\theta_k, z', t)$ derives from an expectation over all timesteps $t \in [t_0, T]$. Therefore, influence estimation inherently overweights timesteps with larger norms and underweights those with smaller norms. For each training sample z', the timestep t_{train} was stochastically sampled during the training process, hence incorporating varying degrees of timestep-induced norm bias.

To this end, we normalize these two terms and define

$$\text{Diffusion-ReTrac}(z, z') = \sum_{k:z_k=z} \eta_k \left(\frac{1}{mT} \sum_{t=1}^{T} \sum_{i=1}^{m} \frac{\nabla_\theta L_t(\theta_k, z', \epsilon_i)}{\|\nabla_\theta L_t(\theta_k, z', \epsilon_i)\|}\right) \cdot \frac{L}{||L||} \tag{11}$$

Bias introduced to influence estimation due to timestep-induced norms is thus mitigated by reweighting the timesteps for z and each training sample z'.

In this way, we minimize the possibility that the calculated influences are dominated by training samples that possess a huge gradient norm only due to stochastic training.

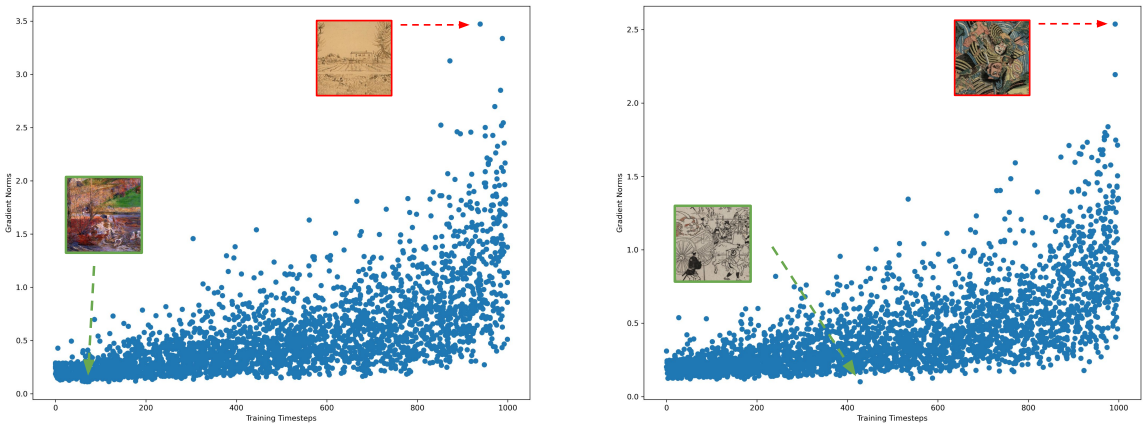
5. Experiments

To illustrate our observation, we present evidence showcasing the dominating-norm bias in influence estimation. We further present instances where this effect may be unnoticed in common benchmarks, and evaluate the performance of Diffusion-TracIn and ReTrac. Our discussion addresses the following questions:

1. **Timestep-induced Bias**: How does timestep affect the influence estimation?
2. **Outlier Detection**: Why might the timestep-induced bias be unnoticed in detecting outliers or atypical samples by calculating *self-influence*?
3. **Image Tracing**: How effective is each method at attributing the learning source of an image to the training data through *test-influence*?
4. **Targeted Attribution**: How does Diffusion-ReTrac outperform Diffusion-TracIn by addressing the norm bias?

5.1 Timestep-induced gradient norm bias

The *dominating gradient norm effect* refers to when influence estimation is biased by a sample having a disproportionately large loss gradient norm induced by diffusion model



(a) Artbench-2 at checkpoint 100

(b) Artbench-2 at checkpoint 200

Figure 1: **Samples' Norm vs. Training Timestep.** We plot the norm and timestep of 3000 randomly selected training samples. We observe that loss gradient norms tend to increase when the training timestep falls in the later range (towards noise). This upward trend is consistent at other checkpoints tested. The sample with the largest norm (red) and smallest norm (green) are shown.

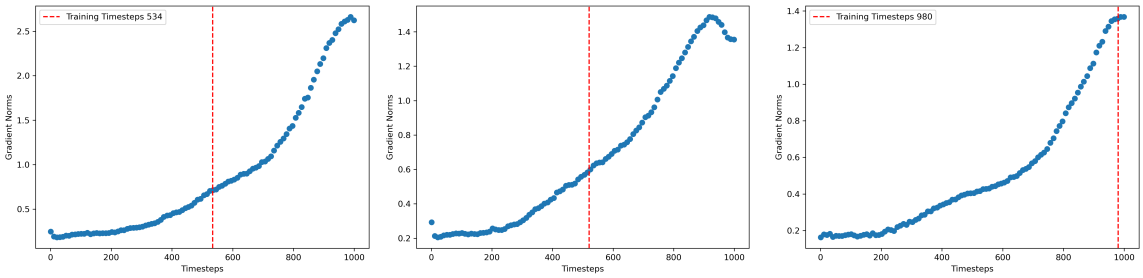
timesteps. Since the training timestep for each instance is stochastically sampled, every instance receives a varying degree of such timestep-induced bias which further propagates into the influence calculation. This further indicates that over-reliance on samples' norms may be a poor source of information, because it can be induced by diffusion timesteps rather than fully attributed to the sample itself.

We demonstrate the presence of such timestep-induced norm by showing:

1. Notable trends between training samples' computed norm and their training timestep.
2. A sample's norm varies greatly if a different timestep was used for the calculation.
3. Statistically significant correlation between norms and training timesteps.

Norm vs. Timestep. We examine the distribution of 3000 randomly selected training samples' loss gradient norm and the training timestep (Figure 1). This is conducted for checkpoints (200, 500, 1000, 1500, and 2000), yielding 5 distributions displaying norms at varying stages of the training process. The distributions all demonstrate a notable upward trend that peaks at the later range of the timesteps (i.e. timesteps closer to noise). This suggests that samples whose training timestep falls within the later range tend to exhibit higher norms.

Varying Timestep for a Single Sample. We further analyze the norm distribution for an individual training sample. At a given model stage (e.g. epoch 500), we compute the loss gradient norm for a fixed sample x at every timestep. We plot the norm distribution for 3000 randomly selected samples and observe that the norm distributions are uni-modal, where the norms tend to increase towards the later timesteps and the peak norm is disproportionately larger than average. Consequently, if the sample x's training timestep falls within or close to the end, then x exhibits a higher-than-average norm. This thus implies that for diffusion



(a) Checkpoint 100 (b) Checkpoint 200 (c) Checkpoint 300

Figure 2: **One Sample's Norms Varying Timestep.** Example of norm distributions for one randomly selected sample is shown. We observe that for each individual sample, the norm distribution is skewed to later timesteps given one checkpoint.

models, a large loss gradient norm can be influenced by specific training timesteps rather than solely being attributed to the sample itself. An example distribution is shown in Figure 2.

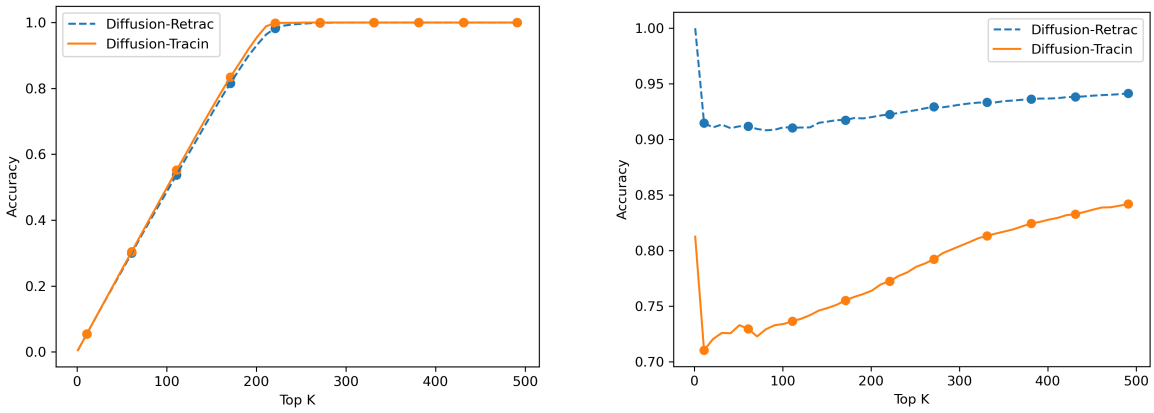
Correlation. Next we examine the actual impact and quantify the relationship between loss gradient norms and training timesteps. We measure the correlation between a sample x having a training timestep chosen close to $T_{x_{\max}}$ (timestep that induces the largest norm for x) and ranking of x's norm among all samples. This is conducted for every checkpoint used to compute influence. The correlation of 0.7 with p-value 1.38×10^{-7} shows a significant positive correlation between the norms and training timesteps. This indicates a notable training timestep-induced norm bias that could well dominate over sample-induced norms, which will then propagate into influence estimation. This provides quantitative insight into Diffusion-TracIn's vulnerability in over-reliance on norm information.

5.2 Outlier Detection

Influence estimation is often used to identify outliers that deviate notably from the rest of the training data. Intuitively, outliers independently support the model's learning at those sparser regions of the input space to which they belong, whereas learning of the typical samples is supported by a wide range of data. Hence in an ideal influence method, outliers tend to exhibit high self-influence, indicating that they exert a high contribution in reducing their own loss. Because of such an outlier-induced norm, we observe that biased estimations may easily go unnoticed in this common metric of outlier detection.

Setup. We begin by training a diffusion model on a combination of the entire 5,000 samples from CIFAR-10 airplane subclass and 200 samples from MNIST zero class. Subsequently, we compute the self-influence of each of the training instances, and sort them by descending order. Since the 200 MNIST samples are outliers that independently support a region, we evaluate whether our methods assign high self-influence to the 200 samples of MNIST zero.

Result. The results show that both Diffusion-TracIn and Diffusion-ReTrac successfully rank outlier samples with high self-influence (Table1). However, the bias introduced by diffusion timesteps is unnoticed in this experiment. Since outliers naturally exhibit larger norms compared to the typical inliers, the timestep-induced norm becomes a more obscure



(a) Accuracy on MNIST zero test samples

(b) Accuracy on CIFAR-plane test samples

Figure 3: **Image Tracing Accuracy.** We evaluate the proportion of correctly attributed training samples among the top-k influential samples identified by the two methods. While both methods successfully attribute MNIST zero test samples (a), Diffusion-TracIn fails to attribute CIFAR-plane test samples (b) since the outlier MNIST samples with large norms received biased estimations.

confounding factor and hence is less subtle in the computation of self-influence. Further analysis of the outlier-induced norm and timestep-induced norm is included in the Appendix.

	Top 100	Top 200	Top 300
Diffusion-TracIn	0.880	0.880	1.000
Diffusion-ReTrac	0.860	0.845	1.000

Table 1: **Outlier Detection.** This table measures the proportion of MNIST samples among top-k identified samples with the highest self-influence. We observe that the performance of these two methods is on par with each other. Both methods assign high self-influence to the 200 MNIST outliers out of the 5,000 CIFAR planes.

Visualization. Examining high-ranking samples further shows that the non-outlier airplane samples with high self-influence (among the top 200) are images with large contrast and atypical backgrounds compared to airplane samples with low self-influence. Overall, samples with high self-influence tend to exhibit high visual contrast or are difficult to recognize. This observation is consistent with patterns revealed in previous work on influence estimation for VAE (Kong and Chaudhuri, 2021).

5.3 Image Tracing

One fundamental role of data attribution methods is to trace the model's outputs back to their origins in the training samples. This idea is also utilized for analyzing *memorization* (Feldman, 2020), a behavior where the generated sample is attributed to a few nearly identical training samples. *Image source tracing* helps pinpoint specific training samples that are responsible for a generation. Thus we evaluate our methods on the question: Given

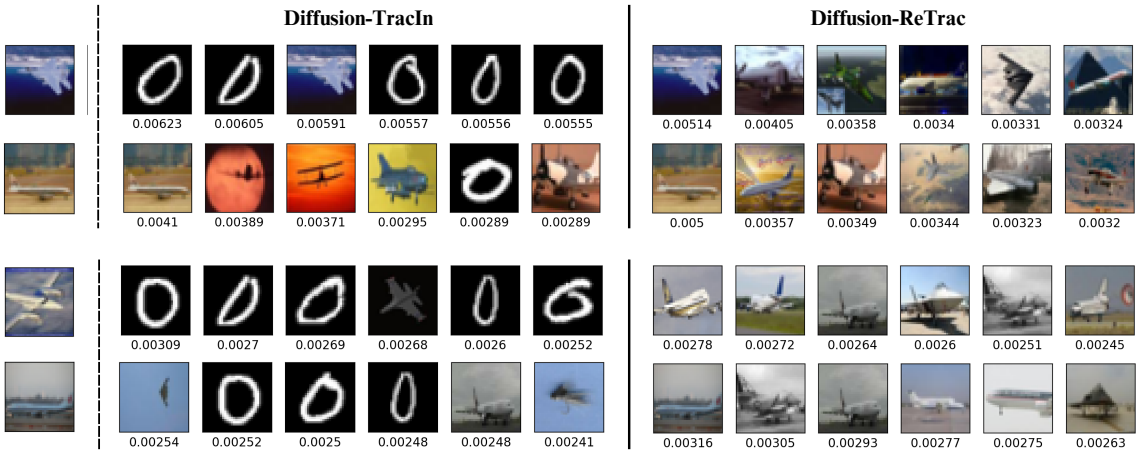


Figure 4: **Image Tracing on Outlier Model.** We evaluate the two attribution methods on both the training samples (top 2 rows) and generated samples (bottom 2 rows). While both methods successfully attribute MNIST zero test samples, Diffusion-TracIn also incorrectly characterizes MNIST training samples as influential for a CIFAR-plane test sample. This is notably mitigated with Diffusion-ReTrac.

a test sample, which instances in the training dataset is the model's knowledge of the test sample derived from?

Setup. We extend our experiment using the aforementioned model trained on CIFAR-10 airplane subclass and MNIST zero (LeCun and Cortes, 2010; Krizhevsky et al., 2009). Given a test sample of MNIST zero, it is expected that the 200 MNIST samples in the training dataset serve as ground truth for the image source. Similarly, a test sample of CIFAR-plane should be attributed to the 5000 CIFAR training samples. We thus obtain an accuracy score by measuring the correctly attributed proportion among the top-k influential sample.

Results. The top-k accuracy scores for Diffusion-TracIn and ReTrac are reported in Figure 3. While both methods successfully attribute the MNIST test samples to the 200 MNIST training samples, we note that Diffusion-TracIn is also more likely to attribute MNIST training samples to a CIFAR-plane test sample (Figure 4). This aligns with the expectation that Diffusion-TracIn tends to assign higher influence to training samples with large norms, in this case, the outlier MNIST zeros. Further analysis of these influential zeros indicates that their training timestep is sampled exactly at or close to the region that induces a large norm. This then becomes the confounding factor that further amplifies these outliers' norms, exacerbating the bias introduced when attributing the test samples.

Visualization. For CIFAR-plane samples that Diffusion-TracIn successfully attributes (without influential MNIST zeros), there still appear to be generally influential planes across test samples. This phenomenon is alleviated for samples retrieved using ReTrac, with sets of influential samples being more distinct and visually intuitive for different test samples. Additionally, the CIFAR-plane instances with high influence (e.g. among top-200) to MNIST test samples tend to be white planes with black backgrounds, which also resemble the test samples.

		Top 10	Top 50	Top 100
ArtBench-2	D-TracIn	0.293	0.261	0.248
	D-ReTrac	0.812	0.646	0.605
CIFAR-10	D-TracIn	0.725	0.663	0.636
	D-ReTrac	0.856	0.800	0.768

Table 2: **Targeted Attribution.** This table compares Diffusion-TracIn and Diffusion-ReTrac on the average proportion of unique samples retrieved over multiple samples. It can be noticed that Diffusion-TracIn overall extracts far less unique samples compared to Diffusion-ReTrac, especially on the two-class ArtBench.

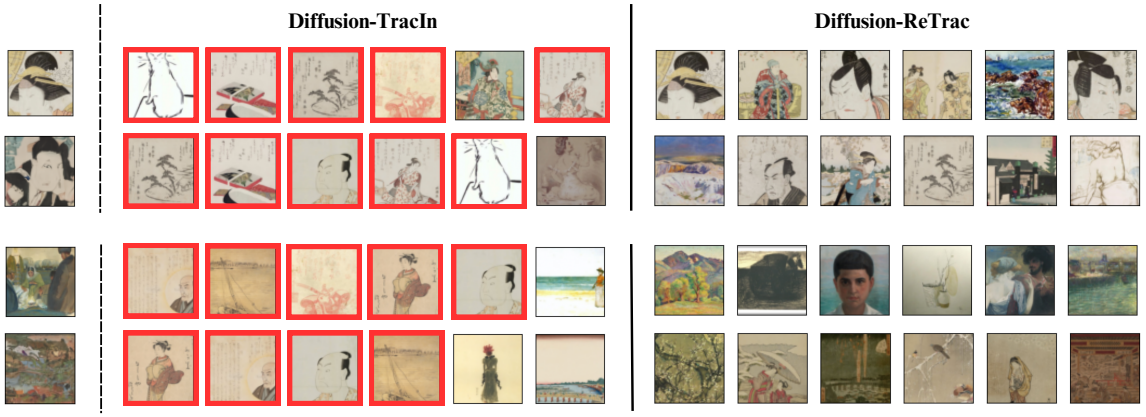
5.4 Targeted Attribution

We then provide a comprehensive analysis of the influential samples retrieved by Diffusion-TracIn and ReTrac. In this experiment, we compute the influence over Artbench-2 and CIFAR-10 datasets. Compared to the previous settings, this experiment minimizes the effects of "sample-induce" large gradient norms introduced by the explicit artificially introduced outliers. This experiment further explores the capability of Diffusion-TracIn and ReTrac in tasks with different emphases or objectives.

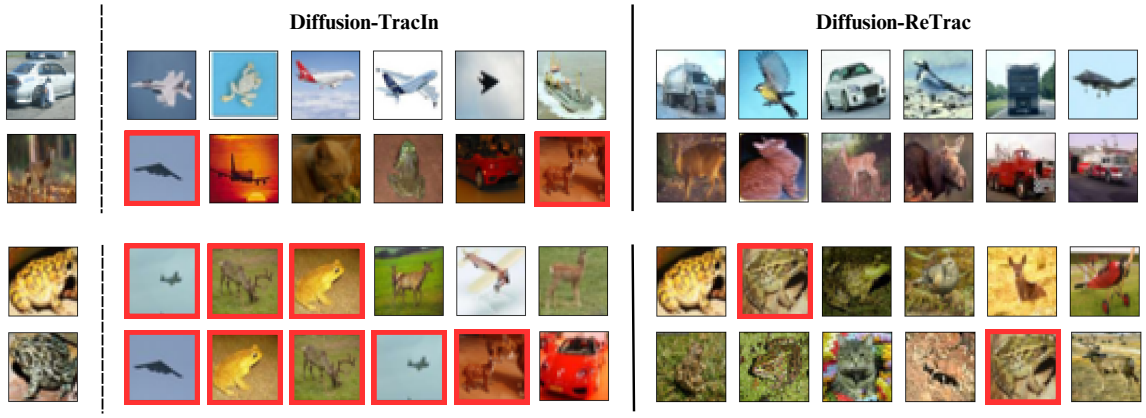
Setup. We compute test-influence on two diffusion models trained with datasets 1). Artbench-2 consisting of "Post-impressionism" and "ukiyo-e" subclasses from Artbench (Liao et al., 2022), each containing 5,000 training samples with resolution 64x64, and 2). CIFAR-10 (Krizhevsky et al., 2009) consisting of 50,000 training samples with resolution 32x32.

Results. To quantify the targeted-ness of the data attribution method, we assess the prevalence of generally influential samples. For a given k, we measure the proportion of distinct samples among the top-k influential samples identified by the two methods. The results are shown in table 2. We note that Diffusion-TracIn yields extremely homogenous influential samples. This trend is particularly evident in ArtBench-2, where the two-class setting is less diverse and more prone to bias induced. In this case, Diffusion-TracIn incurs an $\frac{2}{3}$ of overlaps within the top 10 influential samples. This observation aligns with our intuition and argument that the stochastically chosen timesteps have amplified the number of samples that exhibit larger gradient norms, therefore causing more generally influential samples.

Visualization. It is also visually evident that Diffusion-TracIn retrieves numerous "generally influential" training samples which are assigned high influence scores across multiple different test samples (Figure 5). Moreover, the same or similar sample can be attributed to test samples that are completely different (e.g. in terms of subclass or visual similarities such as colors and structure). Further analysis of the norms associated with these "generally influential" samples is available in the Appendix. This phenomenon is notably mitigated after normalization in Diffusion-ReTrac. The revised approach retrieves influential training samples that bear greater visual resemblance to the test samples, highlighting ReTrac's targeted attribute and reinforcing that dissimilar test samples are more likely to be influenced by a distinct set of training samples.



(a) **Artbench-2.** Top 2 test samples are of the class "Ukiyo-e" and the bottom 2 are of "Post Impressionism".



(b) **CIFAR-10.** Top 2 test samples are of the subclass "automobile", "deer" and bottom 2 are of "frog".

Figure 5: **Targeted Attribution.** We attribute the 8 test samples (leftmost) using both Diffusion-TracIn and Diffusion-ReTrac. The top 6 positively influential samples are shown (row). The generally influential images that appear multiple times for different test samples are indicated (red). It is visually evident that Diffusion-TracIn retrieves numerous "generally influential" training samples, whereas ReTrac provides more distinct attribution results.

6. Conclusion

In this work, we extend data attribution framework to diffusion models and identify a prominent bias in influence estimation originating from loss gradient norms. Our detailed analysis elucidates how this bias propagates into the attribution process, revealing that gradient information harbors undesired bias caused by diffusion model dynamics. Subsequent experiments validate Diffusion-ReTrac as an effective attempt to mitigate this effect, offering fairer and targeted attribution results.

Limitations and future work. A theoretical explanation for the dominating-norm effect better pinpoints the causes and provides ad hoc solutions for the problem. Analysis of other potential confounding factors leading to "generally influential" samples may be conducted to gain further insights into a fair attribution method.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642) and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

References

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR, 2020.

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.

Norm A Campbell. The influence function as an aid in outlier detection in discriminant analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 27(3): 251–258, 1978.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

Zheng Dai and David K Gifford. Training data attribution for diffusion models. *arXiv preprint arXiv:2306.02174*, 2023.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

Kristian Georgiev, Joshua Vendrow, Hadi Salman, Sung Min Park, and Aleksander Madry. The journey, not the destination: How data guides diffusion models. 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Zayd Hammoudeh and Daniel Lowd. Identifying a training-set attack's target using renormalized influence estimation. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1367–1381, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Zhifeng Kong and Kamalika Chaudhuri. Understanding instance-based interpretability of variational auto-encoders. *Advances in Neural Information Processing Systems*, 34:2400–2412, 2021.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

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

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Peiyuan Liao, Xiuyu Li, Xihui Liu, and Kurt Keutzer. The artbench dataset: Benchmarking generative models with artworks. *arXiv preprint arXiv:2206.11404*, 2022.

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

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Naoyuki Terashita, Hiroki Ohashi, Yuichi Nonaka, and Takashi Kanemaru. Influence estimation for generative adversarial networks. *arXiv preprint arXiv:2101.08367*, 2021.

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

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

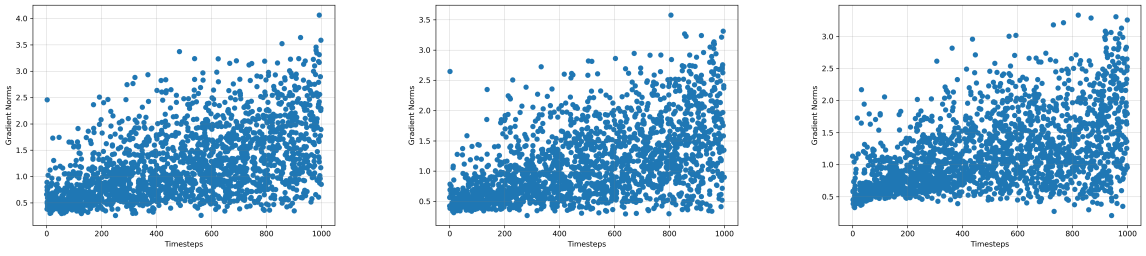
Appendix

A Timestep-Induced Bias

A.1 Norm vs. Timestep

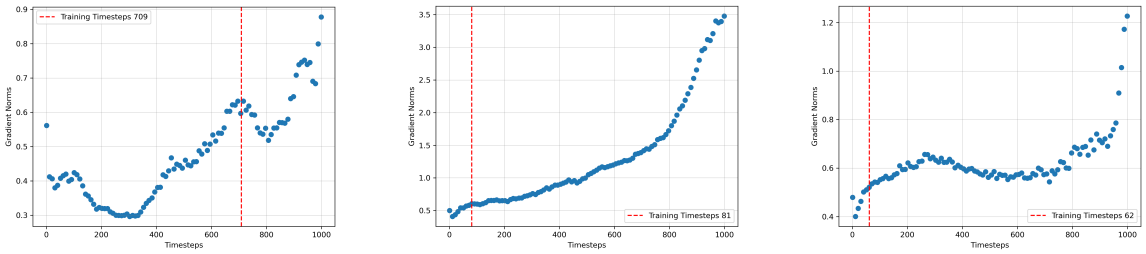
To demonstrate that diffusion timesteps have a significant impact on loss gradient norms, we plot the distribution of 2,000 randomly selected training samples' norms and their training timesteps. Visualization for the distribution is shown in Figure 6. There is a notable upward trend that peaks at the later range of the timesteps (i.e. timesteps closer to noise), suggesting that samples trained during these later timesteps tend to exhibit larger norms.

Additionally, it is also observed that the trend in norm distribution gradually diminishes at the model convergence. This further supports that such variance due to timestep is an artifact of the training dynamic, rather than a property of the training sample. However, since Diffusion-TracIn utilizes gradient information during the full learning process rather than only the ones near convergence, this trend in norms is inevitable in influence estimation. This also motivates the renormalization technique in Diffusion-ReTrac.



(a) Checkpoint 100 (b) Checkpoint 200 (c) Checkpoint 300

Figure 6: **Norm Distribution.** We plot the loss gradient norm and training timestep of 2,000 samples. The distributions at checkpoints 100, 200, and 300 all demonstrate an upward trend. This suggests that samples whose training timesteps fall within the later timestep region tend to have a larger norm.



(a) Checkpoint 100 (b) Checkpoint 200 (c) Checkpoint 300

Figure 7: **Varying Timestep for a Single Sample.** The norms of sample #0 are computed at different timesteps. The distribution obtained at checkpoints 100, 200, and 300 all demonstrates a similar trend that peaks at the later timestep region.

Algorithm 1 Training Timesteps and Norm Ranking

$y_1 \leftarrow \{\}$
$y_2 \leftarrow \{\}$
for $k \leftarrow 0$ to N **do**:
  $s \leftarrow$ k-th training sample
  $y_1 \leftarrow y_1 \cup (T_{s_{\max}} - T_{s_{\text{train}}})$
  $y_2 \leftarrow y_2 \cup \text{Rank}(s)$
end for
return spearman-rank(y_1, y_2)

A.2 Varying Timestep for a Single Sample

We further show that for a fixed training sample, the gradient norm with respect to its loss computed at different timesteps varies significantly. This reinforces the effect of training timestep in the estimation of influence, indicating that each sample receives a varying degree of bias since the training timestep is stochastically sampled. An example norm distribution for a fixed sample at different checkpoints is shown in Figure 7.

A.3 Correlation

If a sample's training timestep falls within its timestep region that gives the larger norms, does this sample also have a relatively larger norm compared to the rest of the training dataset?

To analyze the relationship between the stochastically chosen training timestep and the sample's overall norm ranking among the rest, we obtain a correlation score by i). compute the distance between a sample's training timesteps (T_{train}) and the timestep that yields the maximum norm ($T_{x_{\max}}$), and ii). the ranking of this sample's gradient norm among all the training samples, and iii). calculate a Spearman Rank correlation score between distance and ranking (Algorithm 1).

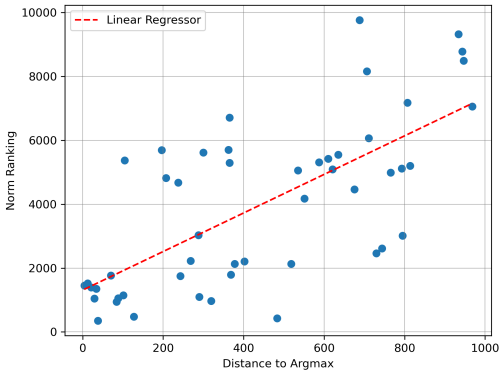


Figure 8: **Correlation of Training Timestep and Norm Ranking.** We plot the norm ranking and distance between training timestep to $T_{\max}$ for 50 randomly selected samples. The linear regressor (red) has a slope of 6.038.

A visualization of the correlation is shown in Figure 8. The linear regressor that fits the data exhibits a slope of 6.038, suggesting a strong correlation between the training timesteps and loss gradient norms.

B Generally Influential Samples

As additional motivation for renormalization, we observe that Diffusion-TracIn assigns dominantly high influence to samples with a large norm at only one checkpoint. We examine the loss gradient norms and training timesteps associated with "generally influential" training

samples (exerts high influence on various different test samples) retrieved by Diffusion-TracIn. It is shown that these generally influential samples all exhibit large norms and a timestep closer to its highest norm region relative to other training samples in the same checkpoint. Furthermore, such samples only emerge as influential when this checkpoint is utilized. This suggests that a substantial norm in a single checkpoint can significantly override prior attribution results, reinforcing the necessity of renormalization in Diffusion-ReTrac.

For illustration, we show the norm of an example generally influential sample in Figure **??** (first image on the last row). This specific sample emerges as influential across various test samples at checkpoint 80, which is also when its training timestep happens to fall within the $T_{\max}$ region (Figure 9).

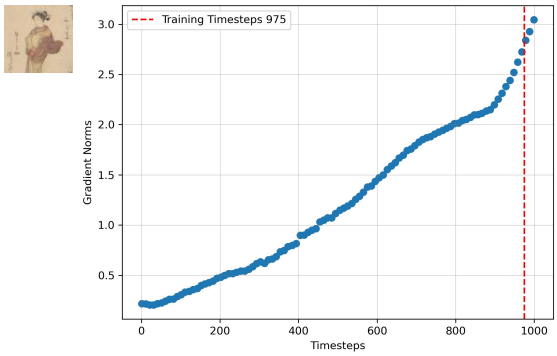


Figure 9: **Generally Influential and Timestep.** Example of a generally influential sample (left) whose training timestep falls exactly within the $T_{x_{\max}}$ region.

C CIFAR-Planes with High Self-influence

Self-influence is used to identify outliers in the training dataset. While Diffusion-TracIn and ReTrac assign high self-influence to most of the 200 MNIST samples, certain CIFAR-plane samples also received high self-influence scores and are ranked among the top 200. These plane samples tend to have dark backgrounds or high contrast, which are also visually distinct from typical samples in the CIFAR-plane subclass (Figure 10).

Figure 10: **CIFAR-planes with High Self-Influence.** These four samples are assigned high self-influence scores by both Diffusion-TracIn and ReTrac. They are visually distinct from the typical plane samples in the training dataset.

D Implementation Details

D.1 Model Details

We trained a DDIM Song et al. (2020) with 1000 training timesteps and 50 inference steps using an Adam optimizer. However, it should be noted that our method works with no difference for DDPM, as we are only working with the training stage. Although the derivation of TracIn assumed certain forms of training (e.g. SGD update rule), it may be modified to handle the variations in training. Specifically, the practical form of TracInCP is expected to remain the same across these variations Pruthi et al. (2020).

D.2 Checkpoint Selection

When estimating influences, it is ideal to select checkpoints with consistent learning and a steady decline in loss. Checkpoints that are early in the model's learning stage often yield fluctuating gradient information, while those near model convergence offer limited insights into the attribution. Influence estimation at these early/late epochs of the learning process can introduce noise and compromise the accuracy of attribution results.

Attribution methods that rely on loss gradient norm information are also particularly sensitive to checkpoint selection. We observe that certain samples may exhibit an unusually large norm at specific checkpoints. When this checkpoint is used in Diffusion-TracIn, such samples emerge as generally influential with notably high influence on various test samples, overshadowing attribution results from previous checkpoints. This effect is mitigated in Diffusion-ReTrac due to renormalization, reducing the method's susceptibility to dominant norms.

D.3 Timestep Selection

To approximate the expectation over timesteps in the attribution efficiently, 50 linearly spaced timesteps over the denoising trajectory are used. This provides similar results to estimating influences across the entire trajectory using T timesteps. It is also observed that the loss induced is relatively stable at neighboring timesteps, while significant variation persists among distant timesteps. This provides justification for reducing computational costs by employing an adequate number of evenly spaced timesteps to approximate the loss over the entire trajectory.