

NONPARAMETRIC DATA ATTRIBUTION FOR DIFFUSION MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Data attribution for generative models seeks to quantify the influence of individual training examples on model outputs. Existing methods for diffusion models typically require access to model gradients or retraining, limiting their applicability in proprietary or large-scale settings. We propose a *nonparametric* attribution method that operates entirely on data, measuring influence via patch-level similarity between generated and training images. Our approach is grounded in the analytical form of the optimal score function and naturally extends to multiscale representations, while remaining computationally efficient through convolution-based acceleration. In addition to producing spatially interpretable attributions, our framework uncovers patterns that reflect intrinsic relationships between training data and outputs, independent of any specific model. Experiments demonstrate that our method achieves strong attribution performance, closely matching gradient-based approaches and substantially outperforming existing nonparametric baselines.

1 INTRODUCTION

Recent advances in generative models, particularly diffusion models (Ho et al., 2020; Song et al., 2021), have led to significant progress in image synthesis and editing (Rombach et al., 2022; Ramesh et al., 2022; Meng et al., 2022). As these powerful models are trained on increasingly large-scale datasets that often contain private, copyrighted, or low-quality content, concerns around data transparency, accountability, and ethical use are growing (Carlini et al., 2023; Saveri & Butterick, 2023). These concerns motivate the study of *data attribution*: identifying the influence of individual training examples on a given generation. Effective data attribution not only supports the responsible deployment of generative models but also enables various downstream applications, including interpreting model behavior (Koh & Liang, 2017; Sui et al., 2021; Ilyas et al., 2022), detecting mislabeled or poisoned data (Jia et al., 2021), guiding data valuation (Nohyun et al., 2023), and improving dataset quality through informed curation (Khanna et al., 2018; Jia et al., 2021; Liu et al., 2021).

Much progress has been made in attributing image generations to training data. Retraining-based methods (Ghorbani & Zou, 2019; Ilyas et al., 2022) assess how generations change when specific training data are removed. While effective, these methods typically require retraining the model tens of thousands of times on different data subsets (Ghorbani & Zou, 2019), making them computationally expensive. To improve efficiency, recent works (Zheng et al., 2024; Lin et al., 2025; Mlodozieniec et al., 2025) adopt approximations based on additive attribution scores (Park et al., 2023a), enabling scalable attribution on large datasets. A common assumption in these methods is access to model gradients, i.e., full access to the generative model. However, this assumption is not always practical. For example, in scenarios where users seek copyright protection against infringement by proprietary models (Somepalli et al., 2023; Zhao et al., 2024), the model gradients may not be accessible. Furthermore, when attribution is intended to support tasks such as data selection (Gu et al., 2025), training a generative model solely for attribution may be prohibitively costly or infeasible.

These challenges call for an attribution method that does not require access to the generative model, either to handle proprietary or black-box models or to serve as a fast surrogate without the cost of training or accessing the model. Existing methods (Zheng et al., 2024; Mlodozieniec et al., 2025) typically measure the similarity between generated images and training data in feature spaces, but often perform poorly as they disregard the behavior of the generative model. Effective attribution for diffusion models in this restricted-access setting remains an open challenge.

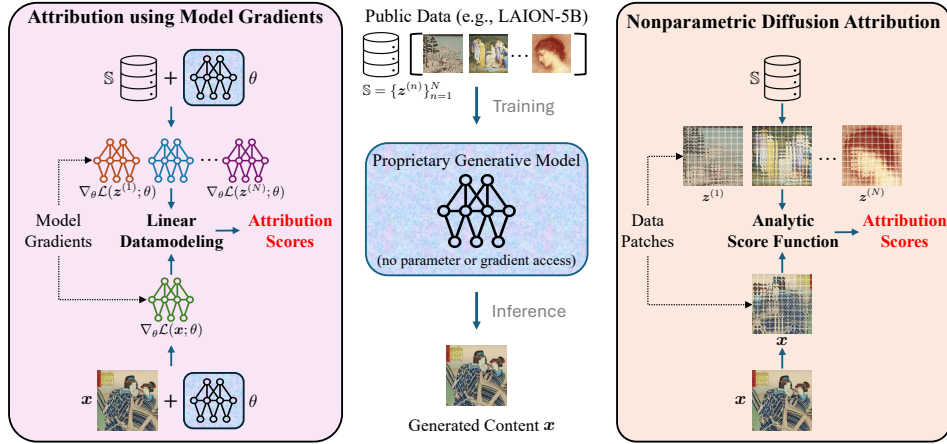


Figure 1: Schematic illustration of attribution methods. **Left:** Model-based attribution relies on gradients and requires parameter access. **Right:** Our *Nonparametric Diffusion Attribution (NDA)* compares local image patches via an analytic score function, enabling attribution without model access.

In this paper, we present a *nonparametric* approach to image data attribution in diffusion models, which directly quantifies the influence of training samples on generated outputs through patch-level comparisons. Drawing inspiration from analytical expressions of the score function in diffusion models (Gu et al., 2024; Kamb & Ganguli, 2024), we extract local patches from both generated images and training samples and compute attribution scores based on pairwise distances, as illustrated in Figure 1. Unlike prior methods that require retraining or gradient access, our method operates entirely on data, without any reliance on model architecture or training. Moreover, it provides fine-grained, patch-level attribution, enabling the localization of influential training regions and offering interpretable insights into the generation behaviors. Due to space constraints, related work is deferred to Appendix C.

2 PRELIMINARIES

Our approach is grounded in the optimal score function of diffusion models. In this section, we begin with a brief overview of diffusion models and their optimal empirical score functions, followed by a review of data attribution techniques and evaluation protocols.

2.1 DIFFUSION MODELS

Diffusion models (Ho et al., 2020; Song et al., 2021) are a class of probabilistic generative models that learn to approximate a data distribution $q(\mathbf{x})$ by modeling a parametrized Markovian process $p_\theta(\mathbf{x})$. Specifically, it defines a forward process that transforms a clean data sample $\mathbf{x} \sim q(\mathbf{x})$ into a noisy sequence $\mathbf{x}_{1:T} = \mathbf{x}_1, \dots, \mathbf{x}_T$ by gradually adding Gaussian noise. The transition probability is given by $q(\mathbf{x}_t | \mathbf{x}_{t-1}) \triangleq \mathcal{N}(\mathbf{x}_t | \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$, where $\{\beta_t\}_{t=1}^T$ denotes a predefined variance schedule. A key property of the forward process is that \mathbf{x}_t can be sampled in closed form at any timestep t :

$$q(\mathbf{x}_t | \mathbf{x}) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \mathbf{x}, (1 - \bar{\alpha}_t) \mathbf{I}), \quad (1)$$

where $\bar{\alpha}_t \triangleq \prod_{s=1}^t \alpha_s$ and $\alpha_t \triangleq 1 - \beta_t$. As t increases from 0 to T , the noise level increases and $\bar{\alpha}_t$ decays from 1 to 0. Consequently, the marginal distribution $q_t(\mathbf{x}_t)$ approaches a standard Gaussian, i.e., $q_T(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$. The reverse process is learned by approximating the reverse conditionals $q(\mathbf{x}_{t-1} | \mathbf{x}_t)$ using a neural network model $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) \triangleq \mathcal{N}(\mathbf{x}_{t-1} | \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$, where the variance σ_t^2 are typically chosen as hyperparameters (Bao et al., 2022). In practice, diffusion models are usually trained to predict the added noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, which relates to the mean via $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t))$ (Ho et al., 2020). The model is learned by minimizing a variational bound on the negative log-likelihood:

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}; \theta) = \mathbb{E}_{\boldsymbol{\epsilon}, t} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x} + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t) \right\|_2^2 \right], \quad (2)$$

where $\epsilon \sim \mathcal{N}(\epsilon|\mathbf{0}, \mathbf{I})$. Let $\mathbb{S} \triangleq \{\mathbf{z}^{(n)} | \mathbf{z}^{(n)} \sim q\}_{n=1}^N$ be the training dataset. The empirical training objective is then $\mathcal{L}_{\text{ELBO}}(\mathbb{S}; \theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}_{\text{ELBO}}(\mathbf{z}^{(n)}; \theta)$. To enhance sample quality, a simplified training objective is often used (Ho et al., 2020):

$$\mathcal{L}_{\text{Simple}}(\mathbf{x}; \theta) = \mathbb{E}_{\epsilon, t} \left[\|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon\|^2 \right], \quad (3)$$

with the empirical objective on \mathbb{S} given by $\mathcal{L}_{\text{Simple}}(\mathbb{S}; \theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}_{\text{Simple}}(\mathbf{z}^{(n)}; \theta)$.

2.2 OPTIMAL SCORE FUNCTIONS

A key property (Song et al., 2021; Kingma & Gao, 2023) of diffusion models is that the optimal noise prediction function $\epsilon^*(\mathbf{x}_t, t)$ is closely related to the *score function* $\mathbf{s}(\mathbf{x}_t, t) \triangleq \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ via:

$$\mathbf{s}(\mathbf{x}_t, t) = -\frac{\epsilon^*(\mathbf{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}}. \quad (4)$$

Intuitively, for a finite dataset $\mathbb{S} = \{\mathbf{z}^{(n)}\}_{n=1}^N$, the marginal distribution $q_t(\mathbf{x}_t)$ becomes a Gaussian mixture centered at the scaled data points $\sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}$: $q_t(\mathbf{x}_t) = \frac{1}{N} \sum_{n=1}^N \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I})$. The score function $\mathbf{s}(\mathbf{x}_t, t)$ therefore admits an analytical form (see Appendix B for details):

$$\mathbf{s}(\mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) = \frac{1}{1 - \bar{\alpha}_t} \sum_{n=1}^N (\sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)} - \mathbf{x}_t) W_t(\mathbf{z}^{(n)} | \mathbf{x}_t), \quad (5)$$

where $W_t(\mathbf{z}^{(n)} | \mathbf{x}_t)$ is a weighting term defined as:

$$W_t(\mathbf{z}^{(n)} | \mathbf{x}_t) = \frac{\mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I})}{\sum_{n'=1}^N \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n')}, (1 - \bar{\alpha}_t) \mathbf{I})}. \quad (6)$$

This score function $\mathbf{s}(\mathbf{x}_t, t)$ can be interpreted as a conditional average over the added noise, where the residual term $\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}$ is averaged across training examples, weighted by the posterior probability $W_t(\mathbf{z}^{(n)} | \mathbf{x}_t)$ that \mathbf{x}_t was transformed from $\mathbf{z}^{(n)}$ at $t = 0$ under the forward process.

2.3 DATA ATTRIBUTION AND EVALUATION METRICS

Given a training dataset \mathbb{S} and a generated sample \mathbf{x} , the goal of data attribution is to quantify the influence of each training example in \mathbb{S} on the generation of \mathbf{x} . Formally, this involves assigning an attribution score $\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S})$ to each training example, reflecting its relative importance in generating \mathbf{x} .

We follow Zheng et al. (2024) and adopt the linear datamodeling score (LDS; Park et al., 2023a) as our evaluation metric, which quantifies how well an attribution method aligns with the ground-truth influence of training data on model outputs. Specifically, for a training dataset $\mathbb{S} = \{\mathbf{z}^{(n)}\}_{n=1}^N$ of size N , LDS evaluates an attribution method τ by first sampling multiple random subsets $\{\mathbb{S}_m \subset \mathbb{S}\}_{m=1}^M$. Let $\theta^*(\mathbb{S}_m)$ denote the generative model trained on subset \mathbb{S}_m , and let $\mathcal{F}(\mathbf{x}; \theta^*(\mathbb{S}_m))$ denote the model output on a test input \mathbf{x} .¹ Then, for each subset \mathbb{S}_m , the attribution scores $\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S})$ assigned to training examples in \mathbb{S}_m are summed to form an attribution-based prediction $g_\tau(\mathbf{x}, \mathbb{S}_m; \mathbb{S})$:

$$g_\tau(\mathbf{x}, \mathbb{S}_m; \mathbb{S}) \triangleq \sum_{\mathbf{z}^{(n)} \in \mathbb{S}_m} \tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}). \quad (7)$$

Finally, the LDS score for attribution method τ on test input \mathbf{x} is calculated as the Spearman rank correlation $\rho(\cdot, \cdot)$ between the ground-truth outputs from the M retrained models (each trained on a different subset \mathbb{S}_m) and the corresponding attribution-based predictions:

$$\text{LDS}(\tau, \mathbf{x}) \triangleq \rho\left(\{\mathcal{F}(\mathbf{x}; \theta^*(\mathbb{S}_m)) : m \in [M]\}, \{g_\tau(\mathbf{x}, \mathbb{S}_m; \mathbb{S}) : m \in [M]\}\right). \quad (8)$$

¹For diffusion models, we set $\mathcal{F} = \mathcal{L}_{\text{Simple}}$, representing the model output used in LDS evaluation.

3 METHODOLOGY

Our approach is motivated by the weighting term in the optimal score function of diffusion models, which naturally encodes the relative importance of each training example during generation. Leveraging this insight, we build upon the recent theoretical framework of Kamb & Ganguli (2024), which extends the analysis of optimal score functions to capture inductive biases that promote generalization in diffusion models. By bridging this framework with the problem of data attribution, we develop a nonparametric data attribution method that does not require access to model gradients or retraining.

3.1 EQUIVARIANT AND LOCAL SCORE MACHINES

While Eq. (5) defines the optimal score function, it relies solely on distance-based weighting in image space and does not generalize beyond the training set. Kamb & Ganguli (2024) derive an analytical form under inductive biases of *locality* and *equivariance*. This formulation preserves the structure of the empirical score but yields meaningful similarity by incorporating spatial structure and symmetry.

Let $\mathbf{x}_t \in \mathbb{R}^{C \times L \times L}$ denote a noisy image at diffusion time t , with C channels and spatial resolution L . For a pixel location $\ell \in [L] \times [L]$, let $\mathbf{x}_{t,\ell} \in \mathbb{R}^C$ denote its pixel value. Define Ω_ℓ as the $P \times P$ neighborhood centered at ℓ , and $\mathbf{x}_{t,\Omega_\ell} \in \mathbb{R}^{C \times P \times P}$ as the corresponding local patch. Let $\mathbb{P}_\Omega(\mathbb{S})$ be the set of all such patches extracted from the training dataset \mathbb{S} . Each patch $\mathbf{u} \in \mathbb{P}_\Omega(\mathbb{S})$ is thus a local crop of some training image $\mathbf{z} \in \mathbb{S}$, and we denote its center pixel as \mathbf{u}_0 . Under the locality and equivariance assumptions, Kamb & Ganguli (2024) show that the optimal MMSE estimator of the score function at pixel location ℓ , denoted by $\mathbf{s}(\mathbf{x}_t, t, \ell) \in \mathbb{R}^C$, takes the form:

$$\mathbf{s}(\mathbf{x}_t, t, \ell) = \sum_{\mathbf{u} \in \mathbb{P}_\Omega(\mathbb{S})} \frac{\sqrt{\bar{\alpha}_t} \mathbf{u}_0 - \mathbf{x}_{t,\ell}}{1 - \bar{\alpha}_t} W_t(\mathbf{u} | \mathbf{x}_{t,\Omega_\ell}), \quad (9)$$

where the weighting term $W_t(\mathbf{u} | \mathbf{x}_{t,\Omega_\ell})$ is defined as:

$$W_t(\mathbf{u} | \mathbf{x}_{t,\Omega_\ell}) = \frac{\mathcal{N}(\mathbf{x}_{t,\Omega_\ell} | \sqrt{\bar{\alpha}_t} \mathbf{u}, (1 - \bar{\alpha}_t) \mathbf{I})}{\sum_{\mathbf{v} \in \mathbb{P}_\Omega(\mathbb{S})} \mathcal{N}(\mathbf{x}_{t,\Omega_\ell} | \sqrt{\bar{\alpha}_t} \mathbf{v}, (1 - \bar{\alpha}_t) \mathbf{I})}. \quad (10)$$

This formulation generalizes Eq. (5) by measuring similarity at the patch level rather than over full images, allowing it to exploit fine-grained local structure.

3.2 PATCH-BASED DATA ATTRIBUTION SCORES

The weighting term in Eq. (10) can be expressed as a softmax over quadratic distances $\frac{\|\mathbf{x}_{t,\Omega_\ell} - \sqrt{\bar{\alpha}_t} \mathbf{u}\|^2}{2(1 - \bar{\alpha}_t)}$, which naturally reflects the contribution of each training patch \mathbf{u} to the generation of pixel $\mathbf{x}_{t,\ell}$. We reinterpret this term as a *patch-wise influence* score that quantifies the influence of a training patch \mathbf{u} on the local region of a generated image. By aggregating these local scores across spatial locations, we obtain a nonparametric, spatially interpretable attribution measure.

Patch-wise influence. For a noisy patch $\mathbf{x}_{t,\Omega_\ell}$ centered at ℓ in a generated image \mathbf{x} at timestep t , we adopt the local weighting from Eq. (10) and define the patch-wise influence score as:

$$\tau(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})) \triangleq \exp\left(-\frac{\|\mathbf{x}_{t,\Omega_\ell} - \sqrt{\bar{\alpha}_t} \mathbf{u}\|^2}{2(1 - \bar{\alpha}_t)}\right) \cdot \left(\sum_{\mathbf{v} \in \mathbb{P}_\Omega(\mathbb{S})} \exp\left(-\frac{\|\mathbf{x}_{t,\Omega_\ell} - \sqrt{\bar{\alpha}_t} \mathbf{v}\|^2}{2(1 - \bar{\alpha}_t)}\right)\right)^{-1}, \quad (11)$$

which is a normalized similarity score over all patches in $\mathbb{P}_\Omega(\mathbb{S})$ extracted from the training set.

Image-level attribution. Given a generated image \mathbf{x} , we apply the forward diffusion process to obtain noisy samples \mathbf{x}_t . For each pixel location ℓ , we extract the $P \times P$ patch $\mathbf{x}_{t,\Omega_\ell}$ from \mathbf{x}_t , using zero-padding near boundaries to handle incomplete patches. The patch-wise influence scores in Eq. (11) is then computed for all training patches $\mathbf{u} \in \mathbb{P}_\Omega(\mathbb{S})$.

To aggregate into an image-level attribution score $\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S})$, we proceed as follows: (1) For each patch $\mathbf{x}_{t,\Omega_\ell}$ of \mathbf{x}_t , we select the k most influential patches from each training image $\mathbf{z}^{(n)}$, denoted by

$\mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)})$.² (2) We sum the influence scores of these top- k patches to estimate the contribution of the training image $\mathbf{z}^{(n)}$ to generating the local region around ℓ . (3) Finally, we aggregate across all spatial locations ℓ and average over a set of timesteps \mathcal{T} :

$$\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) \triangleq \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \sum_{\ell} \sum_{\mathbf{u} \in \mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)})} \tau(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})). \quad (12)$$

This attribution score is both *spatially interpretable*, as it aggregates patch-wise influence with local meaning, and *nonparametric*, as it operates entirely on training data without relying on the model.

3.3 ATTRIBUTION WITH MULTISCALE PATCH-WISE INFLUENCE

The attribution method described above relies on quadratic Euclidean distances between fixed-size patches in the original image space. However, images may have varying resolutions, and a single patch size may fail to capture different levels of information: from fine-grained textures to higher-level structures (Adelson et al., 1984; Lin et al., 2017). Moreover, diffusion models are known to generate different levels of detail across timesteps: early (high-noise) stages capture coarse structures, while later (low-noise) stages refine local details (Jing et al., 2022; Park et al., 2023b).

To account for these effects, we introduce a *multiscale* extension that computes patch-wise influence across multiple resolutions. Specifically, we downsample both generated and training patches and evaluate distances in the lower-resolution space. Let $D(\cdot)$ denote a downsampling operator, and define $\hat{\mathbf{x}}_{t,\Omega_\ell} \triangleq D(\mathbf{x}_{t,\Omega_\ell})$, $\hat{\mathbf{u}} \triangleq D(\mathbf{u})$, and $\hat{\mathbf{v}} \triangleq D(\mathbf{v})$. The low-resolution patch-wise influence score is then:

$$\hat{\tau}(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})) \triangleq \exp\left(-\frac{\|\hat{\mathbf{x}}_{t,\Omega_\ell} - \sqrt{\bar{\alpha}_t} \hat{\mathbf{u}}\|^2}{2(1 - \bar{\alpha}_t)}\right) \cdot \left(\sum_{\mathbf{v} \in \mathbb{P}_\Omega(\mathbb{S})} \exp\left(-\frac{\|\hat{\mathbf{x}}_{t,\Omega_\ell} - \sqrt{\bar{\alpha}_t} \hat{\mathbf{v}}\|^2}{2(1 - \bar{\alpha}_t)}\right)\right)^{-1}. \quad (13)$$

We then combine the original and low-resolution influence measures into a multiscale score:

$$\tau^{\text{ms}}(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})) \triangleq \gamma_t \tau(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})) + (1 - \gamma_t) \hat{\tau}(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})), \quad (14)$$

where $\gamma_t \in [0, 1]$ is a timestep-dependent weighting factor that balances the contribution of the original and low-resolution influence. Finally, we extend this to multiscale image-level attribution by aggregating over timesteps \mathcal{T} and spatial locations ℓ :

$$\tau^{\text{ms}}(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) \triangleq \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \sum_{\ell} \sum_{\mathbf{u} \in \mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)})} \tau^{\text{ms}}(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}; \mathbb{P}_\Omega(\mathbb{S})). \quad (15)$$

3.4 CONVOLUTION-BASED ACCELERATION

Directly evaluating Eq. (11 & 13) on large-scale datasets is computationally challenging. In a naive implementation, each test patch $\mathbf{x}_{t,\Omega_\ell}$ is broadcast against all unfolded training patches $\mathbf{u} \in \mathbb{P}_\Omega(\mathbb{S})$, leading to peak memory consumption of $\mathcal{O}(NL^2CP^2)$, which is P^2 times larger than the dataset itself.

To address this, we propose a memory-efficient implementation that avoids explicit patch unfolding by leveraging convolutional operators. Specifically, to compute $\|\mathbf{x}_{t,\Omega_\ell} - \sqrt{\bar{\alpha}_t} \mathbf{u}\|^2$ for all $L \times L$ patches \mathbf{u} from a training image $\mathbf{z}^{(n)}$, we treat the patch $\mathbf{x}_{t,\Omega_\ell} \in \mathbb{R}^{C \times P \times P}$ as a convolutional kernel. Applying this kernel to the training image yields the inner-products $\langle \mathbf{x}_{t,\Omega_\ell}, \mathbf{u} \rangle$ over all spatial locations in a single convolution pass. The quadratic distance is then obtained as $\|\mathbf{x}_{t,\Omega_\ell}\|^2 - 2\sqrt{\bar{\alpha}_t} \langle \mathbf{x}_{t,\Omega_\ell}, \mathbf{u} \rangle + \bar{\alpha}_t \|\mathbf{u}\|^2$.

This approach leverages GPU-optimized convolutions to reduce memory usage. We further parallelize across the L^2 test patches by batching B patches into a convolutional kernel with B output channels. This yields peak memory $\mathcal{O}(BNL^2)$, which does not explicitly scale with patch size P , thereby avoiding the prohibitive P^2 factor of naive unfolding and enabling scalable attribution on large datasets.

²Formally, $\mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)}) \subset \mathbb{P}_\Omega(\{\mathbf{z}^{(n)}\})$ with $|\mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)})| = k$, and for all $\mathbf{u}' \in \mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)})$ and $\mathbf{u}'' \in \mathbb{P}_\Omega(\{\mathbf{z}^{(n)}\}) \setminus \mathbb{P}_\Omega^k(\mathbf{x}_{t,\Omega_\ell}, \mathbf{z}^{(n)})$, we have $\tau(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}'; \mathbb{P}_\Omega(\mathbb{S})) \geq \tau(\mathbf{x}_{t,\Omega_\ell}, \mathbf{u}''; \mathbb{P}_\Omega(\mathbb{S}))$.

3.5 DISCUSSIONS

Data attribution is often regarded as a lens for understanding model behavior, quantifying the causal influence of individual training examples on model predictions through the learning process. From this perspective, it might seem paradoxical to speak of attribution without specifying a model: if no model is given, what exactly is being attributed?

Our work invites a broader interpretation. In a narrow sense, our nonparametric method can be understood as a principled guess of how training data might have influenced a generated sample, assuming it was produced by some model trained on the same data. Since our method does not access model parameters, it produces the same attribution for all possible models trained on the dataset. At first glance this may appear counterintuitive, because different models can generalize in different ways and might be expected to yield different attribution patterns. However, our empirical results show that the attribution scores remain consistent across a variety of architectures and training regimes. This suggests that there exists a shared, model-agnostic structure to generalization, which reflects intrinsic relationships between training data and outputs, independent of any specific model.

Seen from an even broader perspective, this idea resonates with how humans attribute provenance. People are often able to determine which training images most likely inspired a generated image even without any knowledge of the mechanism that produced it. This form of attribution is grounded in *perceived similarity* rather than *parametric causality*, and it applies whether the image was produced by a neural network, a human artist, or some natural process. In this light, nonparametric data attribution can be viewed as an attempt to formalize this intuitive, model-independent notion of influence: some training examples are simply more responsible for a given generation than others.

4 EXPERIMENTS

We evaluate our method, *Nonparametric Diffusion Attribution (NDA)*, against both nonparametric and gradient-based attribution approaches using two complementary protocols: the linear datamodeling score (LDS), which quantifies alignment with ground-truth influence, and counterfactual evaluation, which assesses the effect of removing influential training data on generated outputs. Our results and ablations show that NDA achieves strong attribution performance without accessing model parameters or gradients, while qualitative visualizations highlight its spatial interpretability and visual consistency.³

4.1 EXPERIMENTAL SETUP

Datasets. We conduct experiments on CIFAR-10 (Krizhevsky, 2009) and CelebA (Liu et al., 2015). For efficient ablation studies, we follow Zheng et al. (2024) and construct a CIFAR-2 subset consisting of 5,000 images from CIFAR-10. Additional dataset details are provided in Appendix D.1.

Target models. For each dataset, we train a diffusion model to serve as the target model for data attribution. On CIFAR-2 and CIFAR-10, we adopt the original DDPM implementation (Ho et al., 2020) to train an unconditional diffusion model with a U-Net backbone containing 35.7M parameters. For CelebA, we use the same implementation but with a modified architecture of 118.8M parameters to accommodate the 64×64 resolution. The number of diffusion steps is fixed to $T=1,000$. Further training details are provided in Appendix D.2. Note that gradient-based attribution methods directly use the target model parameters, whereas our NDA does not access model parameters or architectures.

LDS evaluation. Following Zheng et al. (2024), we sample $M=64$ random subsets of the training set \mathbb{S} , each containing 50% of the samples. For each subset \mathbb{S}_m , we train three models with different random seeds and average their $\mathcal{L}_{\text{Simple}}$ losses as the model output $\mathcal{F}(\mathbf{x}; \theta^*(\mathbb{S}_m))$ for a test input \mathbf{x} . Additional implementation details are provided in Appendix D.3. To ensure a fair comparison, we use the same 1,000-image held-out validation set and 1,000-image generation set as Zheng et al. (2024).

NDA setup. For patch-wise influence, we use patch sizes $P \in [3, 21]$ for different timesteps $t \in \mathcal{T}$ and resolutions, as determined by ablation studies in Sec. 4.5. To compute low-resolution patch-wise influence, we apply an average-pooling downsampling operator $D(\cdot)$ with a window size of 2, reducing patch resolution by half. In all experiments, we select the top- k most influential patches per training image with $k=100$ to obtain image-level attribution scores. Due to low signal-to-noise ratios at large timesteps, we restrict the set of timesteps to $\mathcal{T}=\{100, 200, 300, 400, 500\}$ in our experiments.

³Code is available in Supplementary Material.

Table 1: LDS (%) of different attribution methods on CIFAR-2, CIFAR-10, and CelebA.

Method	CIFAR-2		CIFAR-10		Celeb-A	
	Validation	Generation	Validation	Generation	Validation	Generation
<i>Without Model Access</i>						
Raw pixel (dot prod.)	7.77±0.57	4.89±0.58	2.50±0.42	2.25±0.39	5.58±0.73	4.94±1.58
Raw pixel (cosine)	7.87±0.57	5.44±0.57	2.71±0.41	2.61±0.38	6.16±0.75	4.38±1.63
CLIP similarity (dot prod.)	6.51±1.06	3.00±0.95	2.39±0.41	1.11±0.47	8.87±1.14	2.51±1.13
CLIP similarity (cosine)	8.54±1.01	4.01±0.85	3.39±0.38	1.69±0.49	10.92±0.87	3.03±1.13
NDA (Ours)	24.88±0.42	15.91±0.49	11.81±0.30	7.41±0.45	16.89±0.59	13.92±0.68
<i>Using Model Gradients</i>						
Gradient (dot prod.)	5.14±0.60	2.80±0.55	0.79±0.43	0.74±0.45	3.82±0.50	3.83±1.06
Gradient (cosine)	5.08±0.59	2.78±0.54	0.66±0.43	0.58±0.42	3.65±0.52	3.86±0.96
TracInCP	6.26±0.84	3.76±0.61	0.98±0.44	0.96±0.40	5.14±0.75	5.18±1.05
GAS	5.78±0.82	3.34±0.56	0.89±0.48	0.90±0.41	5.44±0.68	4.69±0.97
TRAK	11.42±0.49	5.78±0.48	2.93±0.46	2.20±0.38	11.28±0.47	7.02±0.89
D-TRAK	26.79±0.33	18.82±0.43	14.69±0.46	11.05±0.43	22.83±0.51	16.84±0.54

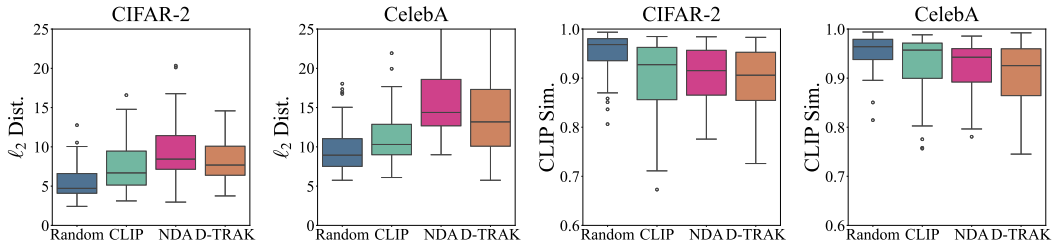


Figure 2: Counterfactual evaluation of ℓ_2 distance (**Left**) and CLIP similarity (**Right**) between original and regenerated images on CIFAR-2 and CelebA after removing the most influential training samples identified by different attribution methods and retraining the model.

4.2 MAIN RESULTS: EVALUATING LDS FOR ATTRIBUTION METHODS

We compare NDA against a range of attribution baselines; detailed descriptions are provided in Appendix D.4. Our primary comparison focuses on representative approaches that do not require access to model parameters, with gradient-based methods included as references. Table 1 reports the results on CIFAR-2, CIFAR-10, and CelebA.

Compared to the strongest baseline based on CLIP cosine similarity, NDA achieves consistent and substantial gains across both validation and generation sets. On CIFAR-2, NDA improves over CLIP by +16.32 (validation) and +11.90 (generation); on CIFAR-10, by +8.42 and +5.72; and on CelebA, by +5.97 and +10.89, respectively. When compared to strong gradient-based methods tailored for diffusion models, such as D-TRAK, NDA substantially closes the gap while accessing no model information. For example, on CelebA, the gap is reduced to 5.94/2.92 (validation/generation), much smaller than the CLIP gap of 11.91/13.81. Overall, NDA approaches the performance of strong parametric baselines while markedly outperforming CLIP similarity on LDS.

4.3 COUNTERFACTUAL EVALUATION

To evaluate the faithfulness and practical effectiveness of NDA, we conduct a counterfactual influence experiment on CIFAR-2 and CelebA. For each generated test image, we first identify the top-1,000 most positively influential training samples according to each attribution method, remove these samples from the training set, and retrain the diffusion model from scratch. The test image is then regenerated using the retrained models under the same random seed, and the impact of removal is quantified using pixel-wise ℓ_2 distance and CLIP cosine similarity. We repeat this procedure for 60 randomly generated test images and report the average results.

We compare NDA against three baselines: Random removal, CLIP similarity (cosine), and D-TRAK. As shown in Figure 2, NDA achieves median ℓ_2 distances of 8.43 and 14.38 on CIFAR-2 and CelebA, respectively, outperforming CLIP (6.68, 10.30) and D-TRAK (7.68, 13.18). For CLIP similarity, NDA attains 0.92 and 0.94, approaching D-TRAK (0.91, 0.93). These results indicate that NDA effectively identifies and removes training samples with strong influence over the generated output, providing a compelling nonparametric alternative to gradient-based attribution. Qualitative visualizations in Figure 3 further show that the generated images exhibit significant distortions after removing the training samples deemed most influential by NDA.



Figure 3: Counterfactual visualization on CIFAR-2 (Top) and CelebA (Bottom). Images are compared to those generated by retrained models using the same seed. See Appendix F.1 for more cases.

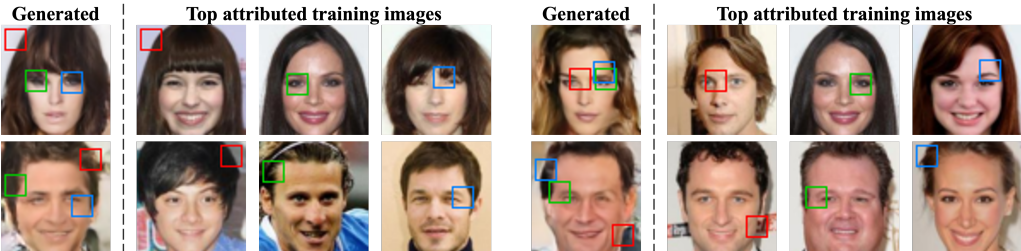


Figure 4: Spatial interpretability of NDA. The patches with the highest patch-wise influence scores (w.r.t. patches in the generated image) are highlighted in the top attributed training images.

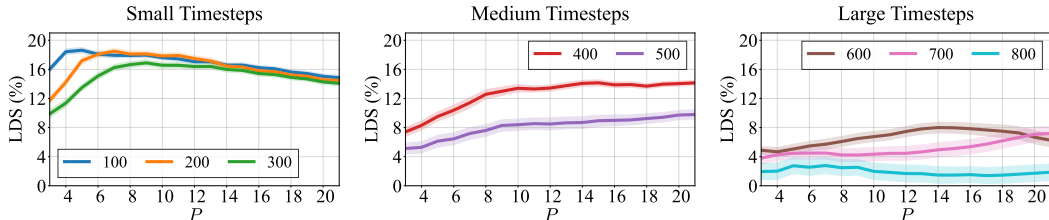


Figure 5: LDS (%) on the CIFAR-2 validation set across different patch sizes and timesteps at the original resolution. **Left:** small timesteps ($t \leq 300$), where moderate patch sizes achieve the highest scores. **Middle:** medium timesteps ($t = 400, 500$), where larger patches outperform smaller ones due to higher noise. **Right:** large timesteps, where noise dominates and informative signal is minimal.

4.4 SPATIAL INTERPRETABILITY

Since NDA computes attribution by aggregating patch-wise influence, it naturally offers an additional spatial level of interpretability. Intuitively, a training image with a high attribution score must contain patches that strongly align with influential regions of the test image. To visualize this correspondence, we highlight the most influential patches from each of the top attributed training images in Figure 4. We observe that top-ranked training images consistently contain local patches that are visually similar to those in the test image, offering a natural explanation for their high attribution scores.

4.5 ABLATION STUDIES

We conduct ablation studies on key hyperparameters of our method. All studies are performed on the validation sets, which are i.i.d. samples drawn from the same distribution as the training sets. The selected hyperparameters are then applied to the generation sets without further tuning to produce the results reported in Sec. 4.2. In all figures where applicable, solid lines show the mean, and shaded regions indicate ± 1 standard deviation.

Patch size selection. We study the effect of patch size by evaluating LDS performance across $P \in [3, 21]$ at different timesteps. To isolate the impact of timestep, we fix \mathcal{T} to contain a single timestep (e.g., $\mathcal{T} = \{100\}, \dots, \{900\}$) and vary P . Figure 5 shows LDS curves as a function of P for each \mathcal{T} on CIFAR-2 (see Appendix E for CelebA). We observe that the optimal patch size varies with timestep and generally increases as t grows. At early timesteps ($t \leq 300$), small to moderate patches ($P = 5, 7, 9$) yield the highest LDS scores, suggesting that local patterns dominate when noise levels are relatively low during reverse diffusion. For mid-range timesteps ($400 \leq t \leq 500$), larger patches perform better, due to their ability to aggregate more contextual information under higher noise levels. In the high-noise regime ($t \geq 600$), LDS scores drop significantly across all patch sizes. Based on these results, we adopt a timestep-dependent patch size selection strategy: using smaller patches for early timesteps and progressively larger patches for later timesteps, choosing the P that maximizes LDS for each t .

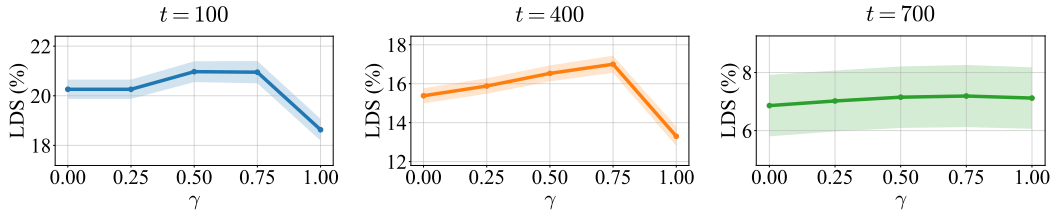


Figure 6: LDS (%) on CIFAR-2 across different weighting factor γ at varying timesteps.

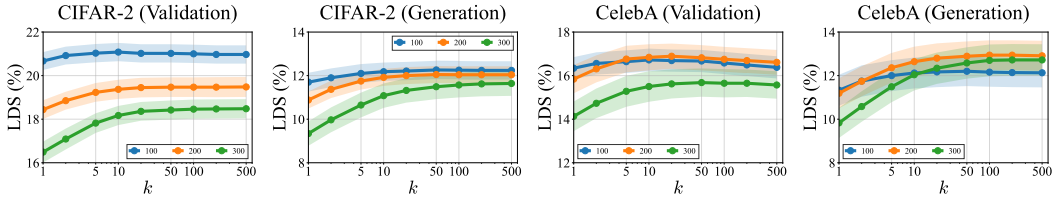


Figure 7: LDS (%) as a function of the number k of top influential patches selected for aggregation on CIFAR-2 (Left) and CelebA (Right). Setting $k=100$ generally provides strong performance.

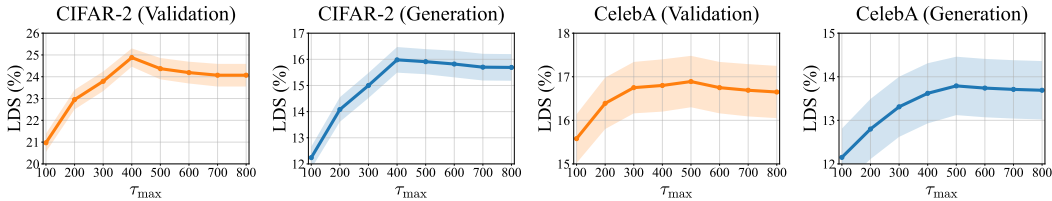


Figure 8: LDS (%) as a function of the max aggregation timestep τ_{\max} on CIFAR-2 (Left) and CelebA (Right). Choosing $\tau_{\max}=500$ generally yields good results.

Multiscale influence. We first find the optimal patch size for the low-resolution patch-wise influence in Eq. (13) using the same procedure as above and observe a similar trend, where early timesteps favor smaller patches. Interestingly, the optimal patch size differs between the original and low-resolution cases (see Appendix E for details). We then evaluate the proposed multiscale attribution in Eq. (15) by varying the weighting factor $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$ across timesteps. As shown in Figure 6, combining multiple scales (e.g., $\gamma=0.75$) consistently improves LDS, particularly in the low-noise regime, suggesting that multiscale features provide complementary information that enhances attribution. Further improvements may be possible by incorporating more scales, which we leave for future work.

Top- k patch selection. We examine how the number of top- k patches affects attribution. Figure 7 shows LDS as a function of k on lower-noise timesteps, which contribute more significantly to the final attribution. We test k ranging from 1 to 500 and find that an intermediate value of $k=100$ generally achieves optimal performance across timesteps and datasets and generalizes well to generation sets.

Timestep selection. Since lower-noise timesteps provide stronger attribution signal, we define $\mathcal{T} = \{100, 200, \dots, \tau_{\max}\}$ and study the effect of varying the max aggregation timestep τ_{\max} . At each $t \in \mathcal{T}$ we use the optimal patch sizes and weighting factors identified above. Figure 8 shows that enlarging \mathcal{T} improves LDS up to mid-range timesteps, after which gains saturate or degrade. On CIFAR-2, performance rises from $t=100$ to a peak near $\tau_{\max}=400$ before dropping by ~ 0.5 at $\tau_{\max} \geq 600$; On CelebA, performance saturates near $\tau_{\max}=400 \sim 500$ with negligible gains beyond. These results suggest that late (high-noise) timesteps contribute little useful signal and may instead inject noise. We therefore fix $\tau_{\max}=500$ for all subsequent experiments, which achieves near-optimal performance while avoiding unnecessary computation and variability.

5 CONCLUSION

We present a nonparametric method for data attribution in diffusion models that measures the influence of training examples via patch-level similarity. Our approach leverages the analytical form of the optimal score function, extends to multiscale representations, and remains efficient through convolution-based operations. It provides spatially interpretable attributions and uncovers intrinsic relationships between training data and generated outputs, independent of any specific model. Experiments on CIFAR-2, CIFAR-10, and CelebA show that our method closely matches gradient-based approaches and outperforms existing nonparametric baselines, demonstrating that model-agnostic, data-driven attribution is a practical tool for understanding generative models.

REFERENCES

- 486
487
488 Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid
489 methods in image processing. *RCA engineer*, 29(6):33–41, 1984.
- 490
491 Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal
492 reverse variance in diffusion probabilistic models. In *International Conference on Learning
493 Representations*, 2022.
- 494
495 Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramèr, Borja
496 Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd
497 USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.
- 498
499 Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input similarity from the
500 neural network perspective. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox,
501 and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran
502 Associates, Inc., 2019.
- 503
504 Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long
505 tail via influence estimation. In *Advances in Neural Information Processing Systems*, 2020.
- 506
507 Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning.
508 In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.
- 509
510 Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in
511 diffusion models, 2024.
- 512
513 Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data
514 selection via optimal control for language models. In *The Thirteenth International Conference on
515 Learning Representations*, 2025.
- 516
517 Zayd Hammoudeh and Daniel Lowd. Identifying a training-set attack’s target using renormalized
518 influence estimation. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and
519 Communications Security, CCS ’22*, pp. 1367–1381, New York, NY, USA, 2022. Association for
520 Computing Machinery. ISBN 9781450394505.
- 521
522 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in
523 neural information processing systems*, 33:6840–6851, 2020.
- 524
525 Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Data-
526 models: Predicting predictions from training data. In *ICML*, 2022.
- 527
528 Ruoxi Jia, Fan Wu, Xuehui Sun, Jiachen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and
529 Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance
530 quantification? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
531 Recognition (CVPR)*, 2021.
- 532
533 Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative
534 models. In *European conference on computer vision*, pp. 274–289. Springer, 2022.
- 535
536 Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models,
537 2024.
- 538
539 Rajiv Khanna, Been Kim, Joydeep Ghosh, and Oluwasanmi Koyejo. Interpreting black box predic-
540 tions using fisher kernels. In *International Conference on Artificial Intelligence and Statistics*,
541 2018.
- 542
543 Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data
544 augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- 545
546 Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In
547 *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- 548
549 Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

- 540 Jinxu Lin, Linwei Tao, Minjing Dong, and Chang Xu. Diffusion attribution score: Evaluating
541 training data influence in diffusion model. In *The Thirteenth International Conference on Learning*
542 *Representations*, 2025.
- 543
544 Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie.
545 Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer*
546 *vision and pattern recognition*, pp. 2117–2125, 2017.
- 547
548 Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, and Conghui He. Influence selection
549 for active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*
(*ICCV*), pp. 9274–9283, October 2021.
- 550
551 Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In
552 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- 553
554 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.
- 555
556 Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon.
557 SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International*
Conference on Learning Representations, 2022.
- 558
559 Bruno Kacper Mlodozieniec, Runa Eschenhagen, Juhan Bae, Alexander Immer, David Krueger, and
560 Richard E. Turner. Influence functions for scalable data attribution in diffusion models. In *The*
561 *Thirteenth International Conference on Learning Representations*, 2025.
- 562
563 Ki Nohyun, Hoyong Choi, and Hye Won Chung. Data valuation without training of a model. In *The*
564 *Eleventh International Conference on Learning Representations*, 2023.
- 565
566 Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK:
567 Attributing model behavior at scale. In *Proceedings of the 40th International Conference on*
Machine Learning. PMLR, 2023a.
- 568
569 Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding
570 the latent space of diffusion models through the lens of riemannian geometry. In *Thirty-seventh*
Conference on Neural Information Processing Systems, 2023b.
- 571
572 Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data
573 influence by tracing gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and
574 H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930.
Curran Associates, Inc., 2020.
- 575
576 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
577 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
578 models from natural language supervision. In *International conference on machine learning*, pp.
579 8748–8763. PmLR, 2021.
- 580
581 Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-
582 conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- 583
584 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
585 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Confer-*
ence on Computer Vision and Pattern Recognition, pp. 10684–10695, 2022.
- 586
587 Joseph Saveri and Matthew Butterick. Image generator litigation. [https://](https://imagegeneratorlitigation.com/)
588 imagegeneratorlitigation.com/, 2023. Accessed: 2024-07-06.
- 589
590 Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Under-
591 standing and mitigating copying in diffusion models. In *Thirty-seventh Conference on Neural*
Information Processing Systems, 2023.
- 592
593 Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben
Poole. Score-based generative modeling through stochastic differential equations. In *International*
Conference on Learning Representations, 2021.

594 Yi Sui, Ga Wu, and Scott Sanner. Representer point selection via local jacobian expansion for
595 post-hoc classifier explanation of deep neural networks and ensemble models. In *Advances in*
596 *Neural Information Processing Systems*, 2021.

597
598 Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai man Cheung, and Min Lin. A recipe for
599 watermarking diffusion models, 2024.

600 Xiaosen Zheng, Tianyu Pang, Chao Du, Jing Jiang, and Min Lin. Intriguing properties of data attribu-
601 tion on diffusion models. In *The Twelfth International Conference on Learning Representations*,
602 2024.

603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647

A LLM USAGE

We used an OpenAI LLM (GPT-5) as a writing and formatting assistant. In particular, it helped refine grammar and phrasing, improve clarity, and suggest edits to figure/table captions and layout (e.g., column alignment, caption length, placement). The LLM did not contribute to research ideation, experimental design, implementation, data analysis, or technical content beyond surface-level edits. All outputs were reviewed and edited by the authors, who take full responsibility for the final text and visuals.

B PROOF OF SECTION 3.2

The empirical DDPM objective is

$$\begin{aligned}
\mathcal{L}_{\text{Simple}}(\mathbb{S}; \theta) &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{t \sim [0, T]} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} [\|\epsilon_{\theta}(\mathbf{x}_t, t) - \epsilon\|^2] \\
&= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{t \sim [0, T]} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} [\|\sqrt{1 - \bar{\alpha}_t} s_{\theta}(\mathbf{x}_t, t) + \epsilon\|^2] \\
&= \mathbb{E}_{t \sim [0, T]} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\frac{1}{N} \sum_{n=1}^N \|\sqrt{1 - \bar{\alpha}_t} s_{\theta}(\mathbf{x}_t, t) + \epsilon\|^2 \right] \\
&= \mathbb{E}_{t \sim [0, T]} \left[\int \frac{1}{N} \sum_{n=1}^N \|\sqrt{1 - \bar{\alpha}_t} s_{\theta}(\mathbf{x}_t, t) + \epsilon\|^2 \mathcal{N}(\epsilon; 0, \mathbf{I}) d\epsilon \right].
\end{aligned} \tag{16}$$

Since $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)} + \sqrt{1 - \bar{\alpha}_t} \epsilon$, we have $\epsilon = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}}{\sqrt{1 - \bar{\alpha}_t}}$. Thus $d\epsilon = \frac{d\mathbf{x}_t}{\sqrt{1 - \bar{\alpha}_t}}$. Then

$$\mathcal{L}_{\text{Simple}}(\mathbb{S}; \theta) = \mathbb{E}_{t \sim [0, T]} \left[\int \frac{1}{N} \sum_{n=1}^N \|\sqrt{1 - \bar{\alpha}_t} s_{\theta}(\mathbf{x}_t, t) + \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}}{\sqrt{1 - \bar{\alpha}_t}}\|^2 \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I}) \sqrt{1 - \bar{\alpha}_t} d\mathbf{x} \right]. \tag{17}$$

The minimization of the objective $\mathcal{L}_{\text{Simple}}(\mathbb{S}; \theta)$ can be solved by taking the gradient w.r.t. $s_{\theta}(\mathbf{x}_t, t)$:

$$\begin{aligned}
\mathbf{0} &= \nabla_{s_{\theta}(\mathbf{x}_t, t)} \left[\frac{1}{N} \sum_{n=1}^N \|\sqrt{1 - \bar{\alpha}_t} s_{\theta}(\mathbf{x}_t, t) + \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}}{\sqrt{1 - \bar{\alpha}_t}}\|^2 \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I}) \right] \\
&= \sum_{n=1}^N 2 \left[s_{\theta}(\mathbf{x}_t, t) + \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}}{1 - \bar{\alpha}_t} \right] \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I}) \\
&= \left[\sum_{n=1}^N \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I}) \right] s_{\theta}(\mathbf{x}_t, t) + \sum_{n=1}^N \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I}) \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}}{1 - \bar{\alpha}_t}.
\end{aligned}$$

Then the optimal diffusion model can be written as:

$$\begin{aligned}
s^*(\mathbf{x}_t; t) &= \frac{\sum_{n=1}^N \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I}) \frac{\sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)} - \mathbf{x}_t}{1 - \bar{\alpha}_t}}{\sum_{n=1}^N \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}, (1 - \bar{\alpha}_t) \mathbf{I})} = \frac{\sum_{n=1}^N \exp\left(-\frac{\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}\|^2}{2(1 - \bar{\alpha}_t)}\right) \frac{\sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)} - \mathbf{x}_t}{1 - \bar{\alpha}_t}}{\sum_{n=1}^N \exp\left(-\frac{\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}\|^2}{2(1 - \bar{\alpha}_t)}\right)} \\
&= \sum_{n=1}^N \mathbb{S}\left(-\frac{\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)}\|^2}{2(1 - \bar{\alpha}_t)}\right) \frac{\sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)} - \mathbf{x}_t}{1 - \bar{\alpha}_t} = \sum_{n=1}^N \frac{\sqrt{\bar{\alpha}_t} \mathbf{z}^{(n)} - \mathbf{x}_t}{1 - \bar{\alpha}_t} W_t(\mathbf{z}^{(n)} | \mathbf{x}_t).
\end{aligned}$$

C RELATED WORKS

C.1 DATA ATTRIBUTION

Training data plays a critical role in shaping the behavior of machine learning models. Data attribution aims to measure the contribution of individual training samples to model predictions. Existing methods can be broadly categorized into two classes: retraining-based and retraining-free approaches.

Retraining-based methods, such as empirical influence functions (Feldman & Zhang, 2020), Shapley value estimators (Jia et al., 2021), and datamodels (Ilyas et al., 2022), provide high-fidelity attributions by measuring the effect of removing or modifying each training example. However, these methods typically require retraining the model tens of thousands of times on different data subsets to achieve reliable results, making them computationally infeasible for large-scale settings. Retraining-free methods aim to approximate influence scores without additional retraining, offering a more scalable alternative. These approaches fall into two main categories. The first comprises gradient-based methods without kernelization, which rely solely on first-order gradient signals and avoid computing second-order derivatives. Representative approaches include Gradient (Charpiat et al., 2019), TracInCP (Pruthi et al., 2020) and GAS (Hammoudeh & Lowd, 2022). The second category includes gradient-based methods with kernelization, which incorporate curvature information by constructing kernels, typically using the inverse of the Hessian matrix. However, Hessian inversion is often numerically unstable. To mitigate this, recent work approximates the Hessian with the Fisher information matrix (FIM). For instance, TRAK (Park et al., 2023a) introduces a kernel-based approximation that efficiently estimates influence scores via random projections, which is both accurate and computationally tractable for large-scale models. D-TRAK (Zheng et al., 2024) further improves attribution performance by modifying the output function and training loss in TRAK, resulting in more effective LDS scores. Based on empirical analysis, D-TRAK recommends a specific configuration that achieves the best overall performance.

C.2 NONPARAMETRIC DATA ATTRIBUTION METHODS

While most data attribution methods rely on parametric models and require gradient computations or model retraining, an alternative line of research explores nonparametric approaches that operate directly on training data without fitting parametric functions. These are typically similarity-based approaches, which estimate the influence of a training sample based on its similarity to the target using a predefined metric. For example, one can leverage pretrained vision-language embeddings such as CLIP (Radford et al., 2021) features to measure high-level semantic similarity of images (Zheng et al., 2024; Mlodozieniec et al., 2025), or compare raw pixel values directly. However, these methods overlook the internal dynamics of the generative model. As a result, effective data attribution for diffusion models in settings without model access remains an open and challenging problem.

D IMPLEMENTATION DETAILS

D.1 DATASETS

CIFAR-10 (32×32). The CIFAR-10 dataset contains 5,000 training samples. For LDS evaluation, we randomly sample 1,000 validation images from the CIFAR-10 test set. To reduce computational overhead, we also construct a subset called CIFAR-2, consisting of 5,000 training images randomly selected from the “automobile” and “horse” in CIFAR-10’s training set, along with 1,000 validation images from the corresponding classes in the test set.

CelebA (64×64). We constructed our dataset by selecting 5,000 training samples and 1,000 validation samples from the original training and test splits of CelebA (Liu et al., 2015). Following the preprocessing steps outlined by (Song et al., 2021), we first center crop the images to 140×140 pixels, and then resize them to 64×64 pixels.

D.2 MODELS

CIFAR. We follow the original implementation of the unconditional DDPM (Ho et al., 2020), which uses a U-Net backbone comprising approximately 35.7M parameters. The model is trained for 200 epochs with a batch size of 128, using a cosine annealing learning rate schedule. Additional model configurations include a dropout rate of 0.1 and the use of the AdamW optimizer (Loshchilov & Hutter, 2019) with a weight decay of 10^{-6} . To enhance robustness, random horizontal flips are applied as a data augmentation strategy.

CelebA. We adopted an unconditional DDPM framework similar to that used for CIFAR-10, but modified the architecture to handle 64×64 inputs. To better capture the higher complexity of CelebA, we scaled up the U-Net model to 118.8M parameters. All training settings, including the noise

756 variance schedule, optimizer configurations, and overall training procedure, were kept consistent with
 757 those used in the CIFAR-10 setup.

759 D.3 LDS EVALUATION SETUP

760
 761 For the LDS evaluation, we sample $M = 64$ random subsets \mathbb{S}_m from the training set, with each
 762 subset comprising 50% of the data (i.e. $\alpha = 0.5$). For each subset, three models are trained using
 763 different random seeds to improve robustness. We then compute the linear datamodeling score
 764 for each sample of interest as the Spearman rank correlation between the model’s output and the
 765 attribution score as described in Eq. (8). In particular, to compute the simple loss $\mathcal{L}_{\text{simple}}(\mathbf{x}, \theta)$ as
 766 defined in Eq. (3) for any sample of interest, we evaluate it over 1000 timesteps uniformly spaced in
 767 the interval $[1, T]$. At each timestep, we further approximate the expectation \mathbb{E}_ϵ by sampling three
 768 standard Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Finally, we average the LDS scores across the validation and
 769 generation samples to obtain the overall performance.

771 D.4 BASELINES

772
 773 We compare our proposed method against two major categories of the attribution methods: (1)
 774 nonparametric similarity-based method, which do not rely on model parameters and instead operate
 775 directly on the image features and representations, and (2) post-hoc retraining-free methods, which
 776 leverage gradient-based representations of a trained model to estimate the attribution score without
 777 requiring model retraining. Within the first category, we include two similarity-based methods: Raw
 778 Pixel and CLIP (Radford et al., 2021). Within the second category, we further divide the methods
 779 into (gradient-based methods without kernels) and (gradient-based methods with kernels). For the
 780 gradient-based methods without kernels, we include the techniques such as Gradient (Charpiat et al.,
 781 2019), TracInCP (Pruthi et al., 2020) and GAS (Hammoudeh & Lowd, 2022). As for the gradient-
 782 based methods with kernel, we compare with some representative methods including TRAK (Park
 783 et al., 2023a) and D-TRAK (Zheng et al., 2024).

784 **Raw Pixel.** This is a naive similarity-based attribution method that directly uses the raw image as the
 785 feature representation. The attribution score is then computed by measuring the similarity—such
 786 as the dot product or cosine similarity—between the query sample and each training sample in the
 787 dataset.

788 **CLIP Similarity.** We adopt CLIP (Radford et al., 2021) as a baseline attribution method. Each target
 789 sample and training sample is embedded using CLIP, and the cosine similarity or the dot product of
 790 normalized embeddings between the two sample is taken as the attribution score. This provides a
 791 simple way to estimate the influence of training data, independent of the diffusion model or the LDS
 792 measurement function.

793 **Gradient.** This method is a gradient-based influence estimator from (Charpiat et al., 2019), where
 794 the attribution score for each training sample is computed using either the dot product or cosine
 795 similarity between its gradient(with respect to the model parameters) and that of a given test sample.

$$796 \tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) = \mathcal{P}^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^*)^\top \cdot \mathcal{P}^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{z}^{(n)}, \theta^*)$$

$$797$$

$$798 \tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) = \frac{\mathcal{P}^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^*)^\top \cdot \mathcal{P}^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{z}^{(n)}, \theta^*)}{\|\mathcal{P}^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^*)\| \|\mathcal{P}^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{z}^{(n)}, \theta^*)\|}.$$

801 where \mathcal{P} is the Gaussian random projection matrix that projects the gradient into low-dimensional
 802 subspace.

803 **TracInCP.** We adopt the TracInCP estimator introduced by (Pruthi et al., 2020) formulated as
 804 $\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) = \frac{1}{C} \sum_{c=1}^C (\mathcal{P}_c^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^c))^\top \cdot (\mathcal{P}_c^\top \nabla_\theta \mathcal{L}_{\text{Simple}}(\mathbf{z}^{(n)}, \theta^c))$ where C denotes the
 805 number of model checkpoints uniformly sampled from the training trajectory, and θ^c represents the
 806 parameters at the c -th checkpoint. We use four checkpoints for each experiment. For example, on
 807 CIFAR-2, we select checkpoints at epochs $\{50, 100, 150, 200\}$.

808 **GAS.** This is a "renormalized" version of the TracInCP that leverages cosine similarity in place of
 809 raw dot products (Hammoudeh & Lowd, 2022).

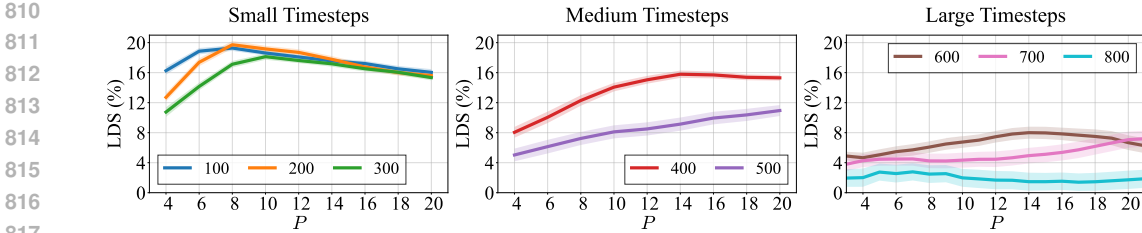


Figure 9: The LDS (%) on the validation set of CIFAR-2 across different patch sizes and timesteps on the low-resolution scale. The left panel shows LDS scores at early (small) timesteps ($t = 100, 200, 300$), where moderate patch sizes yield the highest scores. The middle and right panels show results at late (large) timesteps ($t = 600, 700, 800$), where smaller patches tend to underperform due to overwhelming noise, and larger patches offer relatively more stable performance.

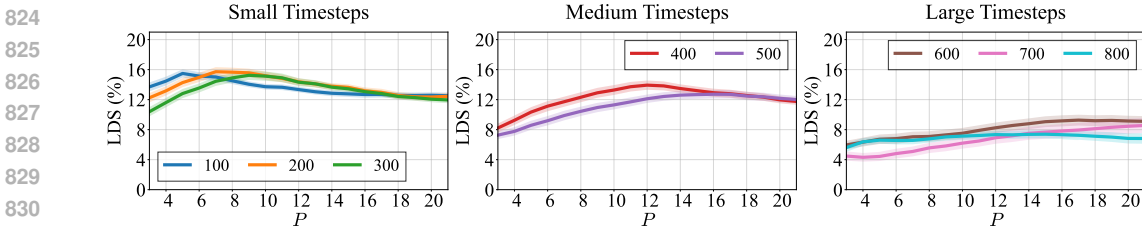


Figure 10: LDS (%) on the CelebA validation set across different patch sizes and timesteps at the original resolution. **Left:** small timesteps ($t \leq 300$), where moderate patch sizes achieve the highest scores. **Middle:** medium timesteps ($t=400, 500$), where larger patches outperform smaller ones due to higher noise. **Right:** large timesteps, where noise dominates and informative signal is minimal.

TRAK. As discussed in (Zheng et al., 2024), TRAK (Park et al., 2023a) can be extended to the diffusion setting and the retraining-free TRAK can be implemented as:

$$\Phi_{\text{TRAK}} = [\phi(\mathbf{x}^1), \dots, \phi(\mathbf{x}^N)]^\top, \text{ where } \phi(\mathbf{x}) = \mathcal{P}^\top \nabla_{\theta} \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^*)$$

$$\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) = \mathcal{P}^\top \nabla_{\theta}^* \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^*)^\top \cdot (\Phi_{\text{TRAK}}^\top \Phi_{\text{TRAK}} + \lambda I)^{-1} \cdot \mathcal{P}^\top \nabla_{\theta} \mathcal{L}_{\text{Simple}}(\mathbf{z}^{(n)}, \theta^*)$$

where the λI is included for numerical stability and regularization effect.

D-TRAK. Similar to TRAK,

$$\Phi_{\text{D-TRAK}} = [\phi(\mathbf{x}^1), \dots, \phi(\mathbf{x}^N)]^\top, \text{ where } \phi(\mathbf{x}) = \mathcal{P}^\top \nabla_{\theta} \mathcal{L}_{\text{Square}}(\mathbf{x}, \theta^*)$$

$$\tau(\mathbf{x}, \mathbf{z}^{(n)}; \mathbb{S}) = (\mathcal{P}^\top \nabla_{\theta} \mathcal{L}_{\text{Simple}}(\mathbf{x}, \theta^*))^\top \cdot (\Phi_{\text{TRAK}}^\top \Phi_{\text{TRAK}} + \lambda I)^{-1} \cdot \mathcal{P}^\top \nabla_{\theta} \mathcal{L}_{\text{Square}}(\mathbf{z}^{(n)}, \theta^*)$$

where λI is also included for numerical stability and regularization, as in TRAK and here $\mathcal{L}_{\text{Square}} = \|\epsilon_{\theta}(\mathbf{x}_t, t)\|^2$. Additionally, the output function $\mathcal{L}_{\text{Square}}$ could be replaced to other functions.

E ABLATION STUDIES

Patch size selection. For completeness, we also study the effect of patch size by evaluating LDS across $P \in [3, 21]$ at different timesteps on CelebA. To isolate timestep effects, we fix \mathcal{T} to a single t and vary P . Figure 10 shows LDS as a function of P for each t . The trend mirrors CIFAR-2: the optimal patch size generally grows with t . At early timesteps ($t \leq 300$), small to moderate patches (e.g., $P = 5, 7, 9$) yield the highest scores, indicating local patterns dominate when noise is low. At mid-range timesteps ($400 \leq t \leq 500$), larger patches perform better (peaks shift to $P = 11, 13$), likely due to the need for more contextual information under higher noise. In the high-noise regime ($t \geq 600$), LDS drops and curves flatten across all patch sizes, suggesting informative signal is limited.

Multiscale influence. We further compare the optimal patch size between the original and the low-resolution settings on CIFAR-2. As shown in Figure 5 and Figure 9, at the early timestep $t=100$,

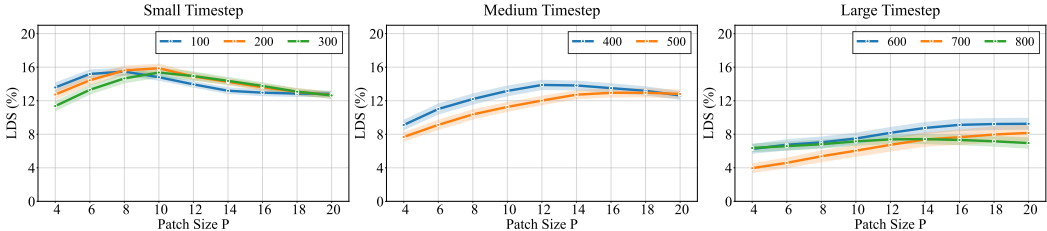


Figure 11: LDS (%) on the CelebA validation set across different patch sizes and timesteps at the low resolution. **Left:** small timesteps ($t \leq 300$), where moderate patch sizes achieve the highest scores. **Middle:** medium timesteps ($t = 400, 500$), where larger patches outperform smaller ones due to higher noise. **Right:** large timesteps, where noise dominates and informative signal is minimal.

the optima differ notably: the original resolution peaks at $P=5$, whereas the low-resolution curve peaks at $P=8$, indicating that coarser inputs benefit from slightly larger spatial context. By contrast, at $t = 200$ and $t = 300$, the two resolutions behave very similarly: on the original scale it peaks at $P = 7$ and $P = 9$ while low-resolution peaks at $P = 8$ and $P = 10$, with near-overlapping curves around their maxima. At mid timesteps ($t = 400, 500$), both resolution favors large patch size with $P = 21$. These results show that downscaling mainly shifts optimal to large patch size P at small timestep, while mid/large timesteps show qualitatively similar trends across resolutions.

F VISUALIZATION

We provide additional visualizations, including counterfactual examples and proponent–opponent analyses.

F.1 COUNTERFACTUAL VISULIZATION

We include more counterfactual visualizations in Figures 12 and 13. As shown, NDA identifies training examples whose removal produces a marked change in the output of a model retrained with the same seed. In Figure 12 (right column, third row), models retrained after removing images selected by Random and CLIP still synthesize an “automobile”. In contrast, removing images selected by NDA or D-TRAK yields an image that resembles a mixture of “automobile” and “horse”. Notably, the image after NDA’s removal shows clearer horse morphology (e.g., head and leg outlines) than that after D-TRAK. These results suggest that NDA is an effective nonparametric approach for identifying training images with strong influence on a given target.

F.2 PROPONENTS AND OPPONENTS VISULIZATION

Following Pruthi et al. (2020), we call training examples with positive influence scores proponents and those with negative scores opponents. For each target, we retrieve and visualize the top-5 proponents and the top-3 opponents. Qualitative results on CIFAR-2, CIFAR-10, and CelebA are shown in Figures 14, 15, and 16. Across datasets, NDA consistently retrieves proponents that are visually closer to the target than those selected by CLIP, and its opponents are correspondingly more dissimilar.

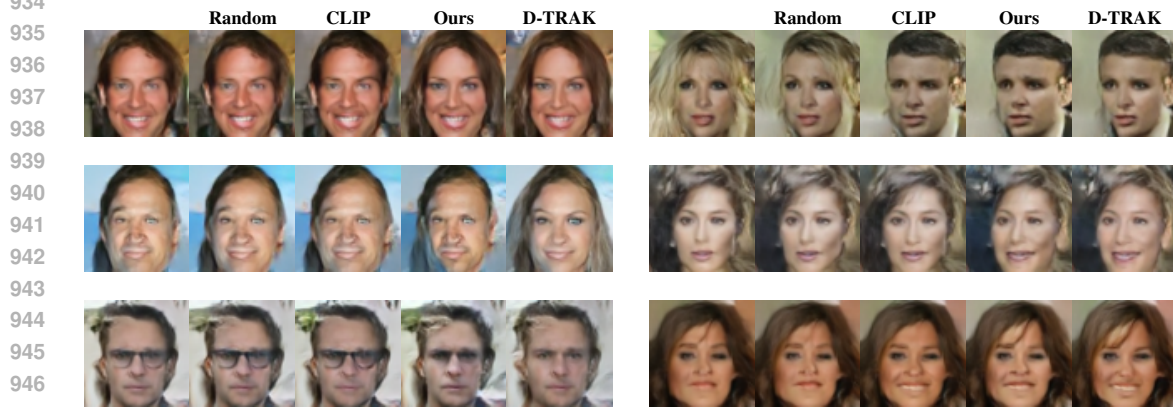
G ADDITIONAL ARCHITECTURES AND TRAINING OBJECTIVES

We further evaluate NDA under two new settings, varying both model architectures and training objectives:

1. **DiT-DDPM:** The target model uses a DiT architecture and is trained with the standard DDPM loss $\mathcal{L}_{\text{Simple}}$ in Eq. (3).
2. **DiT-Flow Matching:** The target model uses a DiT architecture and is trained with the rectified-flow loss \mathcal{L}_{FM} .



931 Figure 12: Counterfactual visualization on CIFAR-2 dataset. We compare samples generated by
 932 retrained models with different attribution methods using the same seed.
 933



948 Figure 13: Counterfactual visualization on CelebA dataset. We compare samples generated by
 949 retrained models with different attribution methods using the same seed.
 950

951

952 Table 2 reports LDS results for NDA and the baselines under the **DiT-DDPM** and **DiT-Flow Matching**
 953 settings described above. NDA consistently outperforms the Raw pixel and CLIP baselines and
 954 remains close to D-TRAK in both cases, indicating that NDA produces attributions that remain
 955 consistent across different models and training objectives.
 956

957 H ADDITIONAL PATCH-WISE BASELINES

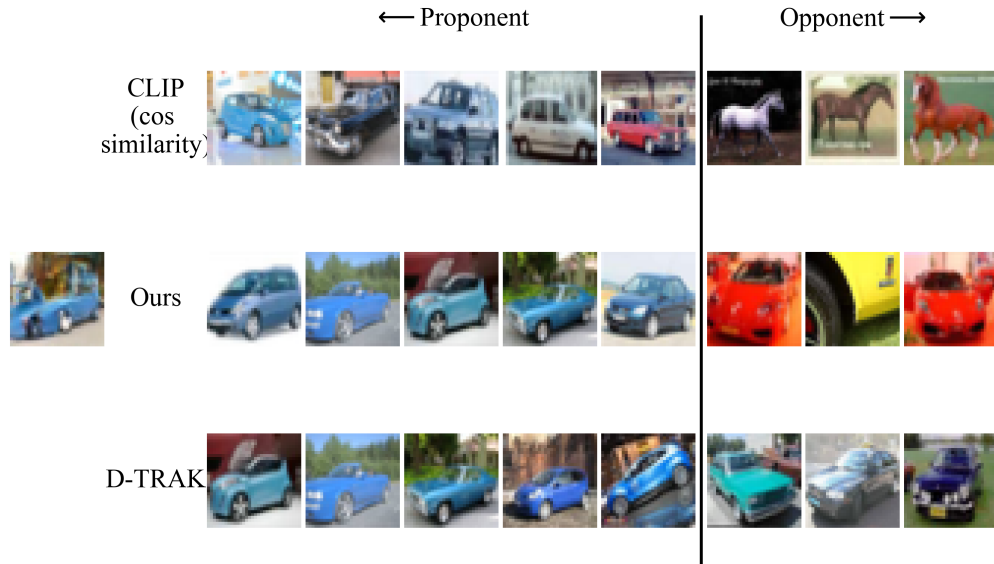
958

959 We include additional results on two patch-wise baselines. For both baselines, we extract local
 960 patches from the query image x and each training image z using the same setup as in Sec. 3.1. Given
 961 these patches:
 962

- 963 1. **Patch-wise Raw pixel:** We compute cosine similarity between every patch of x and every
 964 patch of z . For each patch of x , we select its top- k most similar patches in z and sum their
 965 scores, then aggregate these per-patch sums over all patches of x to obtain an image-level
 966 similarity score.
- 967 2. **Patch-wise CLIP:** We encode each patch with CLIP, and then compute cosine similarities
 968 between patch embeddings of x and z , and apply the same procedure as above to get an
 969 image-level score.
 970

971 We tune the patch size $P \in \{5, 7, 9, 11\}$ and the top- k parameter separately for each patch-wise
 baseline on the CIFAR-2 validation dataset. For the patch-wise Raw pixel baseline, $P = 9$ and

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025



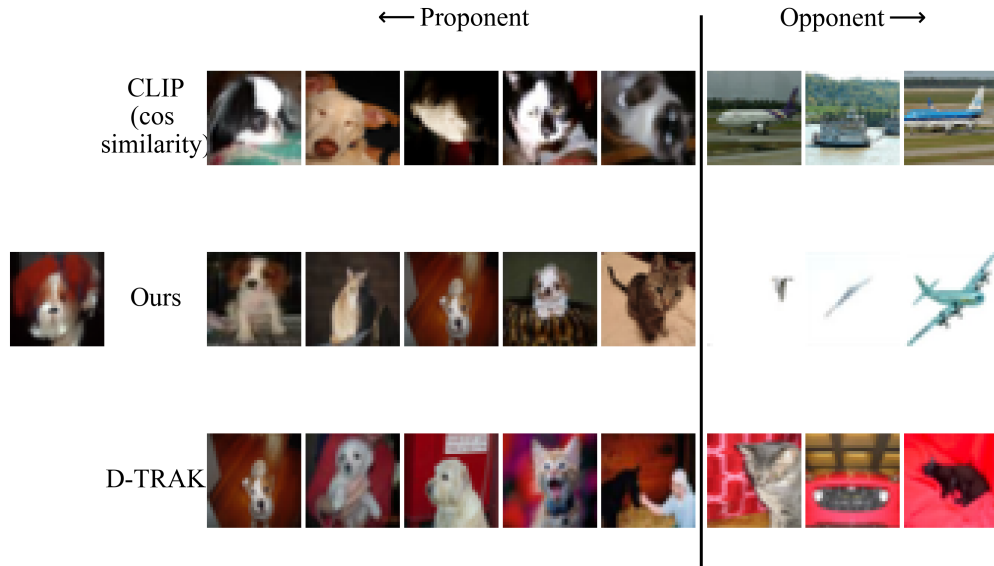
(a) Generation



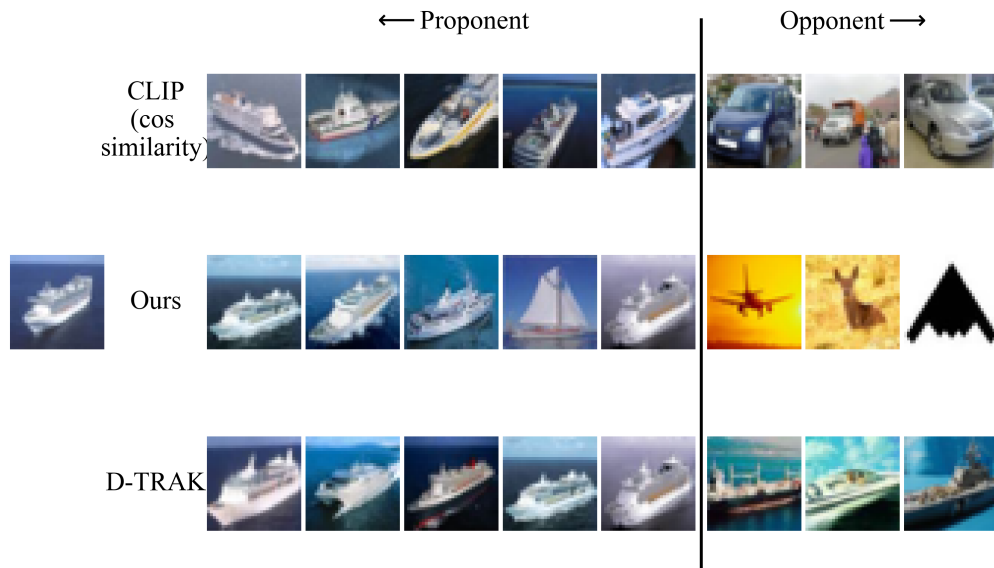
(b) Validation

Figure 14: Proponents and opponents visualization on CIFAR-2 dataset using CLIP, NDA and D-TRAK using averaged timestep. For each sample of interest, we retrieve the most influential training samples based on attribution scores. 5 most positive influential training samples(left) and 3 most negative influential training samples(right) are shown.

1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079



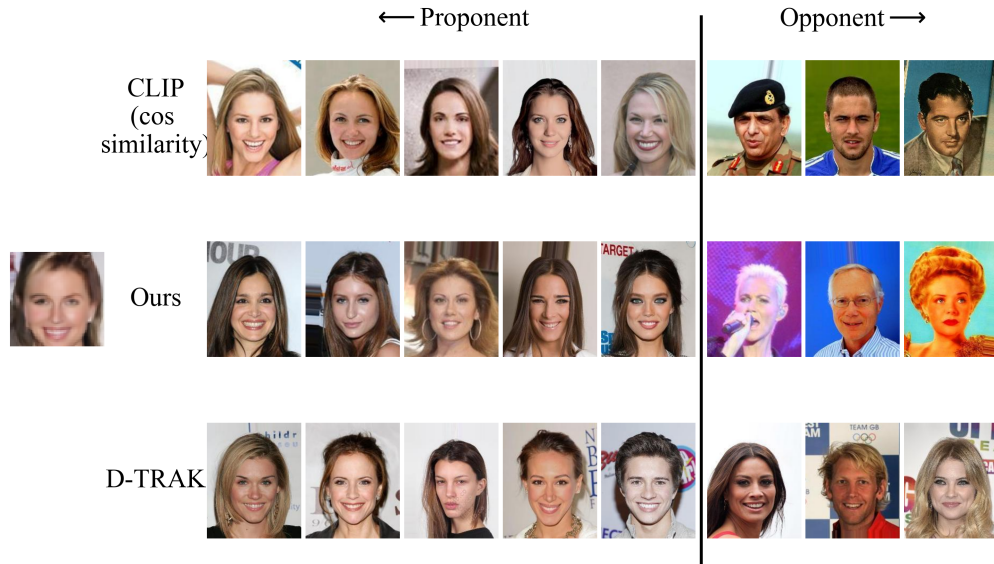
(a) Generation



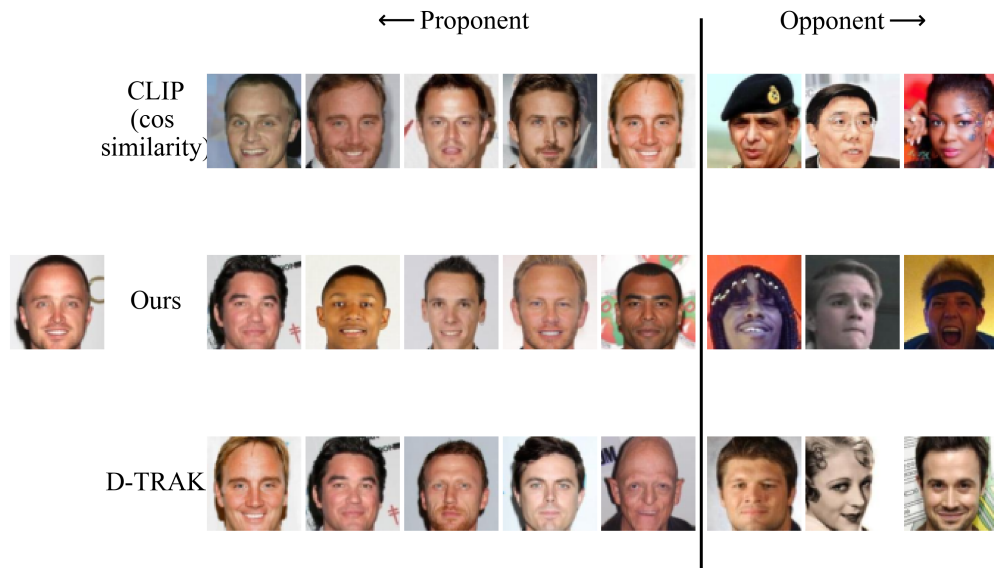
(b) Validation

Figure 15: Proponents and opponents visualization on CIFAR-10 dataset using CLIP, NDA and D-TRAK using averaged timestep. For each sample of interest, we retrieve the most influential training samples based on attribution scores. 5 most positive influential training samples(left) and 3 most negative influential training samples(right) are shown.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133



(a) Generation



(b) Validation

Figure 16: Proponents and opponents visualization on CelebA dataset using CLIP, NDA and D-TRAK using averaged timestep. For each sample of interest, we retrieve the most influential training samples based on attribution scores. 5 most positive influential training samples(left) and 3 most negative influential training samples(right) are shown.

Setting	(Dataset)	Raw pixel (cos.)	CLIP (cos.)	NDA	D-TRAK
DiT-DDPM	(CIFAR-2)	10.81	4.32	17.13	21.17
DiT-DDPM	(CIFAR-10)	1.73	5.13	6.07	6.62
DiT-DDPM	(CelebA)	8.84	6.92	11.62	15.18
DiT-Flow	(CIFAR-2)	3.95	5.30	15.61	15.20
DiT-Flow	(CIFAR-10)	1.71	2.82	9.10	8.41

Table 2: LDS (%) on CIFAR-2, CIFAR-10, and CelebA validation datasets for DiT target models under DDPM and Flow Matching objectives.

Method	CIFAR-2		CIFAR-10		CelebA	
	Validation	Generation	Validation	Generation	Validation	Generation
Patch-wise Raw pixel	4.16	2.42	1.18	2.03	3.26	3.77
Patch-wise CLIP	6.61	4.73	1.20	1.61	6.35	5.76
NDA	24.88	15.91	11.81	7.41	16.89	13.92

Table 3: LDS (%) comparison between patch-wise Raw pixel / CLIP baselines and NDA on CIFAR-2, CIFAR-10, and CelebA.

$k = 100$ perform best; for the patch-wise CLIP baseline, $P = 5$ and $k = 100$ work best. We fix these choices for all datasets and report the corresponding results in Table 3.

As shown in Table 3, NDA consistently outperforms both patch-wise Raw pixel and patch-wise CLIP baselines across all datasets, suggesting that simply patchifying the images does not improve LDS and that NDA’s gains arise from insights theoretically motivated by the analytical score, rather than from patchification alone.