Large-Scale Training Data Attribution for Music Generative Models via Unlearning

Woosung Choi[1*] **Junghyun Koo**[1*] **Kin Wai Cheuk**[1*]
Joan Serrà[1] **Marco A. Martínez-Ramírez**[1] **Yukara Ikemiya**[1]
Naoki Murata[1] **Yuhta Takida**[1] **Wei-Hsiang Liao**[1] **Yuki Mitsufuji**[1,2]

Sony AI[1] Sony Group Corporation[2]

Abstract

This paper explores the use of unlearning methods for training data attribution (TDA) in music generative models trained on large-scale datasets. TDA aims to identify which specific training data points contributed the most to the generation of a particular output from a specific model. This is crucial in the context of AI-generated music, where proper recognition and credit for original artists are generally overlooked. By enabling white-box attribution, our work supports a fairer system for acknowledging artistic contributions and addresses pressing concerns related to AI ethics and copyright. We apply unlearning-based attribution to a text-to-music diffusion model trained on a large-scale dataset and investigate its feasibility and behavior in this setting. To validate the method, we perform a grid search over different hyperparameter configurations and quantitatively evaluate the consistency of the unlearning approach. We then compare attribution patterns from unlearning with non-counterfactual approaches. Our findings suggest that unlearning-based approaches can be effectively adapted to music generative models, introducing large-scale TDA to this domain and paving the way for more ethical and accountable AI systems for music creation.

1 Introduction

Generative AI has demonstrated impressive capabilities across modalities, including text, images, audio, and video, reshaping artistic creation, as shown by [1, 16, 33]. While democratizing creative work, these advancements have also raised concerns regarding authorship, copyright, attribution, and ethics. Notably, generative models can unintentionally reproduce copyrighted material, posing risks of intellectual property violations [6, 28]. As highlighted by Deng et al. [7], these issues are especially pressing in music, where proper attribution and credit to original artists and creators are critical, yet often neglected. To address this, training data attribution (TDA) has emerged as a promising direction to identify which training data points contribute to a model's output, thereby enabling fair crediting.

TDA can be approached in two scenarios based on model access. In the black-box scenario, where the model is inaccessible, corroborative (similarity-based) attribution is typically performed by computing similarity between generated outputs and training data using external feature encoders [2, 4]. While practical, it relies entirely on each encoder's perspective, which does not necessarily align with the generative model's perspective or its inner workings. In contrast, the white-box scenario assumes access to the model's parameters, enabling attribution methods that directly reflect the model's internal behavior. An intuitive approach in this setting is based on counterfactual reasoning [20], asking how the model's prediction would change if a particular training data point was removed.

[*] Equal contributon. Email: {*first_name.last_name*}@sony.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Creative AI Track.

The straightforward solution then to measure the influence of a training data point $\mathbf{x}_i$ is to retrain the whole model without $\mathbf{x}_i$ (leave-one-out retraining). However, this method is computationally unfeasible for large-scale datasets. Instead of retraining, Koh and Liang [20] and Park et al. [24] approximate the change in loss using the influence function [13]. To our knowledge, Deng et al. [7] is the first work in music generation that explored influence function-based TDA methods, proposing an algorithmic solution to estimate the impact of individual training data items on the generated music. They applied these methods to a Music Transformer [15] trained on the MAESTRO dataset [14], which contains about 200 hours of virtuosic piano performances. The same experimental setup was adopted by [8] to further explore ensemble-based TDA approaches.

While the aforementioned methods approximate loss change via the influence function or the ensemble approach, machine unlearning emerges as a promising approach to emulate the counterfactual model. Machine unlearning was proposed by Cao and Yang [5] to forget a specific data item from a pretrained model. Recent studies on TDA have employed gradient ascent to maximize the loss on specific training samples, a process that can be interpreted as unlearning [19, 29]. To mitigate catastrophic forgetting, Wang et al. [29] introduced a regularization technique using the Fisher Information Matrix (FIM) when unlearning a sample $\mathbf{x}_i$ from the pretrained model, measuring the change in loss between two different checkpoints as the attribution score. Although unlearning-based TDA has been actively explored in other domains, its application to music generation remains unstudied.

In this paper, we adapt unlearning-based TDA to measure the attribution score of individual training examples on generated music. We train a latent DiT-based text-to-music generation model [10] on a private dataset comprising 115 k high-quality music tracks with diverse musical styles, totaling approximately 4,356 hours. To validate our unlearning-based TDA pipeline, we adopt a self-influence attribution setup to assess whether our unlearning method effectively approximates the counterfactual-based approach. In this setup, we evaluate the fidelity of our unlearning method to ensure accurate TDA by unlearning a training example and evaluating its influence on other training samples (this could be seen as the case where a model incidentally generated a direct duplicate of a training sample). Inspired by [11], we assess self-influence based on two criteria: (1) the influence of the removed sample must be effectively eliminated, and (2) the model's overall performance must remain stable. We detail these two metrics in 3.2. We then leverage these criteria to conduct a grid search and identify optimal configurations for TDA with unlearning.

Using the best configuration from the self-influence setup, we further analyze test-to-train attribution, comparing the unlearning approach with several other non-counterfactual TDA methods that treat the generative model either as a black-box (similarity-based) or as a white-box (similarity- and gradient-informed), and examine how their influence patterns differ. To our knowledge, this is the first work that explores TDA on a text-to-music DiT, trained on large dataset of diverse musical styles.

2 Methodology

Intuitively, the attribution score can be obtained by measuring the impact of removing a training sample on the model's ability to generate a target output $\hat{\mathbf{z}}$ given the condition $\mathbf{c}_i$. Let θ_0 be the model trained on the full dataset $D = \{\mathbf{z}_i\}_{i=1}^{N}$, where $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{c}_i)$ is an audio–caption pair, and $\theta_{\setminus \mathbf{z}_i}$ be the model trained without $\mathbf{z}_i$. While there are multiple ways to define the attribution scores $\tau\left(\hat{\mathbf{z}}, \mathbf{z}_i\right)$, we adopt the definition based on the changes in loss as in [29], which is defined as

$$\tau\left(\hat{\mathbf{z}}, \mathbf{z}_i\right) = \mathcal{L}(\hat{\mathbf{z}}, \theta_{\setminus \mathbf{z}_i}) - \mathcal{L}(\hat{\mathbf{z}}, \theta_0), \tag{1}$$

However, this leave-one-out method is computationally costly, especially when the dataset is huge and the model training is expensive. A slightly better alternative is to unlearn $\mathbf{z}_i$ from θ_0 to obtain $\theta_{\setminus \mathbf{z}_i}$. Nonetheless, this approach still requires unlearning θ_0 for N times to obtain the exhaustive attribution scores for the whole dataset D.

As proposed by Wang et al. [29], an even better solution is to approximate Eq. 1 with the mirrored influence hypothesis [19] by unlearning the generated sample $\hat{\mathbf{z}}$ from a pretrained model θ_0 to obtain an approximation to $\theta_{\setminus \hat{\mathbf{z}}}$

$$\tau\left(\hat{\mathbf{z}}, \mathbf{z}_i\right) = \mathcal{L}(\mathbf{z}_i, \theta_{\setminus \hat{\mathbf{z}}}) - \mathcal{L}(\mathbf{z}_i, \theta_0). \tag{2}$$

This approach requires only a single unlearning step per generated sample, further reducing the computation cost. In the following section, we detail the unlearning algorithm used in this work to obtain $\theta_{\setminus\hat{\mathbf{z}}}$.

2.1 Unlearning Algorithm

While the most intuitive way to unlearn $\hat{\mathbf{z}}$ is to directly maximize its loss, this approach can lead to catastrophic forgetting [18]. To unlearn the generated sample $\hat{\mathbf{z}}$ without catastrophic forgetting, the unlearned objective should be defined as

$$\mathcal{L}^{\hat{\mathbf{z}}}_{\text{unlearn}}(\theta) = -\mathcal{L}(\hat{\mathbf{z}}, \theta) + \sum_{\mathbf{z}_i \in D} \mathcal{L}(\mathbf{z}_i, \theta), \tag{3}$$

where the first term unlearns $\hat{\mathbf{z}}$ by maximizing the diffusion loss of the generated sample and the second term acts as a regularization to prevent the model from forgetting existing training data by minimizing the diffusion loss for the dataset D. Borrowing the idea from [22], the second term can be simplified by applying second-order Taylor expansion around θ_0:

$$\begin{aligned}\mathcal{L}(\mathbf{z}, \theta) &\approx \mathcal{L}(\mathbf{z}, \theta_0) + \nabla_\theta \mathcal{L}(\mathbf{z}, \theta_0)^\top (\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)^\top \mathbf{H} (\theta - \theta_0) \\ &\approx \frac{1}{2}(\theta - \theta_0)^\top \mathbf{F} (\theta - \theta_0),\end{aligned} \tag{4}$$

where $\mathcal{L}(\mathbf{z}, \theta_0)$ is a constant and the gradient $\nabla_\theta \mathcal{L}(\mathbf{z}, \theta_0)$ at θ_0 should be close to zero for a fully trained model, so both terms can be ignored, leaving behind only the last term. It has been proven in existing literature [3, 12, 21] that the Hessian $\mathbf{H}$ is equivalent to the Fisher information matrix (FIM) $\mathbf{F}$. Plugging Eq. 4 back to Eq. 3, we have the following unlearning objective:

$$\mathcal{L}^{\hat{\mathbf{z}}}_{\text{unlearn}}(\theta) = -\mathcal{L}(\hat{\mathbf{z}}, \theta) + \frac{N}{2}(\theta - \theta_0)^\top \mathbf{F} (\theta - \theta_0). \tag{5}$$

Note that $\nabla_\theta \mathcal{L}^{\hat{\mathbf{z}}}_{\text{unlearn}}(\theta) = 0$ when the loss attains its optimal point and $\nabla_\theta (\theta - \theta_0)^\top \mathbf{F} (\theta - \theta_0) = 2\mathbf{F}(\theta - \theta_0)$ (the gradient of quadratic form for symmetric $\mathbf{F}$). Taking the gradient of Eq. 5 w.r.t θ and rearranging the terms on both sides, we have the following update rule [29]:

$$\begin{aligned}0 &= -\nabla_\theta \mathcal{L}(\hat{\mathbf{z}}, \theta) + N\mathbf{F}(\theta - \theta_0) \\ \theta &= \theta_0 + \frac{1}{N}\mathbf{F}^{-1} \nabla \mathcal{L}(\hat{\mathbf{z}}, \theta).\end{aligned} \tag{6}$$

Note that for diffusion models, the loss depends on the denoising timestep t. So we calculate the average loss across multiple timesteps T, i.e. $\mathcal{L}_(\hat{\mathbf{z}}, \theta) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t(\hat{\mathbf{z}}, \theta)$.

2.2 Fisher Information Matrix

The FIM quantifies the amount of information an observation z carries about the model parameters θ, reflecting the curvature of the log-likelihood function. Mathematically, the FIM is defined as

$$\mathbf{F} = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \theta)} \left[\left(\nabla_\theta \log p(\mathbf{z} \mid \theta)\right) \left(\nabla_\theta \log p(\mathbf{z} \mid \theta)\right)^\top \right].$$

Computing the full FIM is often costly. Thus, a diagonal approximation is commonly used [18], where each diagonal element $(\mathbf{F}_{\text{diag}})_{jj}$ is estimated by averaging the squared gradients over the N data samples $\mathbf{z}_i$:

$$(\mathbf{F}_{\text{diag}})_{jj} \approx \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\partial \log p(\mathbf{z}_i \mid \theta)}{\partial \theta_j} \right)^2.$$

In the context of diffusion models where $\log p(\mathbf{z}_i \mid \theta) = \mathcal{L}_t(\mathbf{z}_i, \theta)$, this is further averaged across T timesteps t:

$$(\mathbf{F}_{\text{diag}})_{jj} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T} \sum_{t=1}^{T} \left(\frac{\partial \mathcal{L}_t(\mathbf{z}_i, \theta)}{\partial \theta_j} \right)^2.$$

Now, we have all the information required to unlearn our model via Eq. 6, and then calculate the attribution score using Eq. 2.

2.3 Masking Silence

For music generative modeling, we build upon the DiT model proposed by Evans et al. [9], which processes variable-length audio as input. Generally, zeros are padded to shorter clips to match the required length. Notably, one can choose to apply or omit masking for this padded section. Specifically, a mask M can be applied to exclude the padded section from loss computation. We assume the model was trained without masking, which is the default setting [2].

Despite this setting, we can apply a mask $\mathrm{M_U}$ when unlearning a target sample, and a mask $\mathrm{M_L}$ when computing the loss for measuring attribution. We also propose a "mixed" strategy: applying $\mathrm{M_U}$ but not $\mathrm{M_L}$. $\mathrm{M_U}$ ensures that the zero-padded section, not part of the actual content, does not interfere with unlearning. However, during loss computation, we omit $\mathrm{M_L}$ to remain consistent with the training setup. A different setup may result in the model's unpredictable behavior, leading to inaccurate attribution.

3 Experimental Results

3.1 Dataset and Model

We utilize an in-house dataset consisting of 115 k high-quality music tracks spanning a diverse range of genres and styles. This dataset is used exclusively to train our base music generative model. To evaluate unlearning-based attribution, we consider two experimental setups: (1) *Train-to-Train*, which estimates attribution by unlearning individual training samples, and (2) *Test-to-Train*, which attributes generated outputs to specific training data points. The Train-to-Train setup serves as a controlled way to test the efficacy of the unlearning method by unlearning a training instance and measuring attribution scores to verify whether the unlearned sample is correctly identified as highly influential. In this setup, we select 40 training samples using k-means clustering on CLAP audio embeddings to ensure diversity across the dataset, and use them for the grid search experiments described in Section 3.2. The Test-to-Train setup is employed in the context of a more qualitative evaluation: we generate 16 two-minute music tracks using distinct text prompts and examine which training data the model attributes to each generated output. This setup is used for comparison with other attribution results, as described in Section 3.3.

As mentioned in the previous section, we train a latent diffusion transformer [DiT; 25] building upon the methodology of Stable Audio [9, 10]. We first train a variational autoencoder [VAE; 17] using the Stable Audio configuration to encode 44.1 kHz stereo audio into a latent space with a dimensionality of 64 and a time downsampling ratio of 2048. We employed the v-objective diffusion process method [26] to train our latent DiT. The maximum length of audio that our DiT can process is approximately two minutes, corresponding to 2584 latent frames. The model is conditioned on CLAP embeddings to enable text-to-music generation, as described in Evans et al. [10]. Additionally, it incorporates timing conditions to support variable-length generation, following the methodology outlined in Evans et al. [9]. We computed $\mathrm{FD_{openl3}}$ on Song Describer reference data [23] to evaluate the overall quality of the generated music, following Evans et al. [10]. Our music generative model achieved an $\mathrm{FD_{openl3}}$ of 110.5, which falls between the performance of Stable Audio 1.0 (142.5) and 2.0 (71.3).

For each unlearning step following Eq. 6, we average gradients over 2048 random timesteps. A single step takes approximately 20 minutes on an NVIDIA H100 80GB GPU. Computing then the losses for all the training data points requires around 5 hours using 8 H100 GPUs.

3.2 Self-Influence Experiment and Tuning

The employed unlearning method features a number of hyperparameters, such as learning rate, target layers, and number of steps (number of model weights' updates following Eq. 6). To explore such different options and select the best combination, we performed a grid search. In this grid search, we computed Train-to-Train data attribution, where we unlearn a train data point from the model and compute the attribution score (Eq. 1) for each item in the training set. Specifically, we examined learning rates ranging from 10^{-7} to 10^{-1}, number of steps from 1 to 4, multiple groups of target layers, and different methods for masking silence (see section 2.3). Due to space constraints, we

[2] https://github.com/Stability-AI/stable-audio-tools

Table 1: Grid search results for optimal unlearning hyperparameters. $\mathrm{FD}_{\mathrm{openl3}}$ is 110.5 for the original checkpoint.

Target Layer	M_{U}	M_{L}	$\mathbf{R}(\mathbf{z}_{\mathbf{tar}})$	$\mathbf{CLAP}_{\mathbf{topk}}$	$\mathbf{CLAP}_{\mathbf{botk}}$	$\mathbf{FD}_{\mathbf{openl3}}$
Cross-Attention's *to_kv* weights	✓		103.2	0.38	0.35	110.5
Cross-Attention Layers	✓		1.4	0.60	0.32	110.4
Self-Attention Layers	✓		1.1	0.63	0.30	110.5
All the Transformer Layers	✓	✓	**1.0**	0.80	0.38	110.5
All the Transformer Layers			6615.7	**0.82**	0.42	110.5
All the Transformer Layers	✓		**1.0**	0.66	**0.26**	110.5

report only the effects of the target layers and the methods for masking silence, as summarized in Table 1, while fixing the learning rate to 10^{-6} and the number of steps to 1 (the best combination we found for these hyperparameters).

In Table 1, we report several metrics to evaluate whether we successfully unlearned the target data while preserving other information. The rank of the target sample, denoted as $\mathrm{R}(\mathbf{z}_{\mathrm{tar}})$, represents the position of the unlearned data point $\mathbf{z}_{\mathrm{tar}}$ in the sorted list of attribution scores $\{\tau(\mathbf{z}_{\mathrm{tar}}, \mathbf{z}_i)\}_{i=1}^{n}$, where a rank of 1 corresponds to the best attribution score. If $\mathrm{R}(\mathbf{z}_{\mathrm{tar}})$ is greater than 1, it indicates that some samples, which are not the target, were more affected than the target. We also report $\mathrm{CLAP}_{\mathrm{topk}}$, the mean CLAP cosine similarity of the top-k attribute scores. Specifically, we measure the mean CLAP cosine similarities of $\mathbf{z}_{\mathrm{tar}}$ and $\mathbf{z}_k$, where $\mathbf{z}_k$ belongs to the top k attribution scores. We hypothesize that unlearning a target track impacts tracks with similar musical components more than irrelevant tracks, making a higher $\mathrm{CLAP}_{\mathrm{topk}}$ preferable. We measure the cosine similarities of the top 100 tracks. Similarly, we report $\mathrm{CLAP}_{\mathrm{botk}}$, the mean CLAP cosine similarity of 100 tracks with the least significant attribution scores.

As shown in Table 1, omitting both masks did not result in a low $\mathrm{R}(\mathbf{z}_{\mathrm{tar}})$. The $\mathrm{R}(\mathbf{z}_{\mathrm{tar}})$ is particularly high for shorter tracks (typically less than 30 s) because unlearning these tracks involves unlearning the padded long silence, which may have a greater impact than the actual content. Enabling both masks prevents this, achieving an $\mathrm{R}(\mathbf{z}_{\mathrm{tar}})$ of 1.0. However, some short tracks appear frequently in different target tracks due to a mismatch between training loss and attribution loss. Extremely short tracks (less than 10 s) tend to have high losses for their actual (non-padded) frames because the losses are averaged over the entire frames (120 s). To mitigate this problem, we used a mixed strategy, applying only M_{U}, which results in the same $\mathrm{R}(\mathbf{z}_{\mathrm{tar}})$ of 1.0.

We also investigated the effect of the target layers. In our experiment, unlearning all the weights in the transformer blocks achieved the highest $\mathrm{CLAP}_{\mathrm{topk}}$ and the lowest $\mathrm{CLAP}_{\mathrm{botk}}$ among the mixed results. In contrast to Wang et al. [29], unlearning only self-attention layers, cross-attention layers, or to_kv layers in each cross-attention layer was found to be suboptimal (Table 1). $\mathrm{FD}_{\mathrm{openl3}}$ did not vary significantly, indicating that the model does not forget information unrelated to the target sample. We unlearned all the transformer layers with the mixed strategy in the subsequent Test-to-Train attribution experiment.

3.3 Comparison with Non-counterfactual Methods

We finally compare our unlearning-based attribution method with alternative approaches that do not incorporate counterfactual reasoning: CLAP [30], CLEWS [27], LPIPS [32], and RPS [31], under the *Test-to-Train* setup. These methods can all preserve the temporal dimension by windowing the input audio into overlapping segments. Attribution is computed in an *all-against-all* manner, where similarity is computed across all time-wise segments for both target and training tracks and the maximum value is taken as the attribution score. The **CLAP** model encodes 10-second audio segments, and we extract embeddings with a hop size of 1 second to preserve temporal resolution across the track. **CLEWS**, a contrastive embedding for capturing musical identity across different versions of the same musical piece, follows CLAP in treating the generative model as a black-box. In contrast, **LPIPS** utilizes intermediate activations from the generative model by computing similarity at each DiT layer's output and then averaging across all layers. **RPS** decomposes the model's pre-activation prediction $\Phi(\hat{z})$ into a weighted sum over training examples: $\Phi(\hat{z}) = \Sigma_{i=1}^{n} \tau(\hat{z}, z_i)$. The attribution score of i^{th} training sample is $\tau(\hat{z}, z_i) = \alpha_i f(z_i)^{\top} f(\hat{z})$, where $f(\cdot)$ is the feature

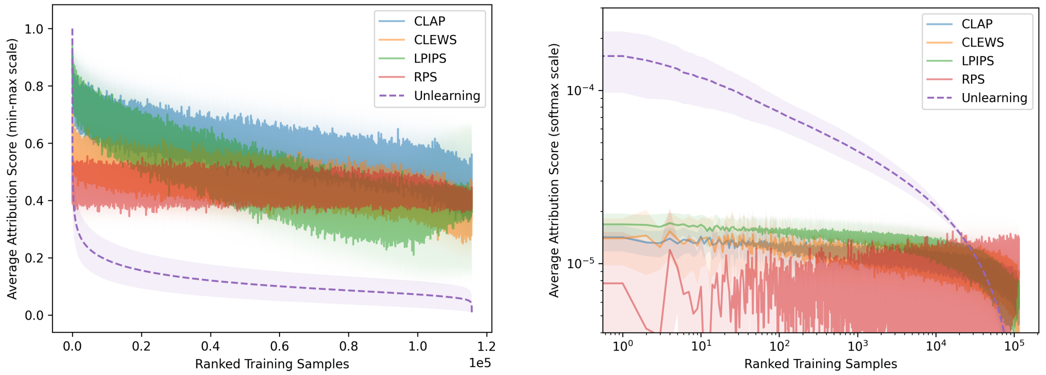

Figure 1: Comparison of attribution scores from unlearning- and similarity-based methods. Mean (line) and standard deviation (shading) over attribution scores from 16 generated test samples. Min-max (left) and softmax (right) normalizations are shown (notice the logarithmic axes in the later).

vector from the last layer of DiT. The representer value $\alpha_i = \frac{1}{-2\lambda n}\frac{\partial \mathcal{L}(z_i,\theta)}{\partial \Phi(z_i,\theta)}$ can be reformulated as $\alpha_i = \frac{1}{-\lambda n}(\Phi(z_i) - z_i)$ as the training objective of our generative model is mean squared error. The final attribution score is computed by taking the average of signed sum of its components.

Figure 1 visualizes the attribution scores' distribution by sorting all training samples in descending order, based on their unlearning scores, and retrieving the corresponding scores from each of the other methods for the same samples. We present two views: one using min-max normalization to account for scale differences across methods, and the other using softmax scaling (where each method's scores sum to 1) to highlight differences in the overall attribution distribution and sparsity patterns. From these plots, we observe: (1) the unlearning-based method exhibits a sharp concentration of influence in the top few samples, indicating high influence on a small subset of training examples; and (2) the similarity-based methods follow a similar trend but with higher variance and a more gradual decrease in attribution scores, suggesting a broader and less concentrated attribution pattern. These qualitative patterns are confirmed by the correlation analysis shown in Figure 2. Notably, the unlearning-based scores show the strongest Pearson correlation with other attribution methods in the order of LPIPS, CLAP, CLEWS, and RPS, with correlation coefficients of 0.56, 0.46, 0.32, and 0.11, respectively. This result aligns with methodological similarities: unlearning and LPIPS may exhibit the highest correlation as both leverage internal information from the generative model. Likewise, CLAP and CLEWS also show strong mutual correlation, reflecting their reliance on external embeddings. In contrast, RPS demonstrates low correlation with all other methods, suggesting it captures a distinct attribution pattern.

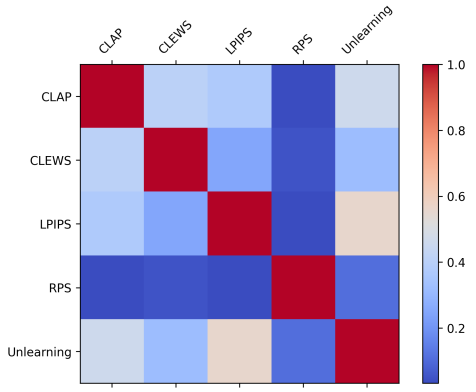

Figure 2: Correlation matrix between different attribution methods.

4 Conclusion

This paper presents a practical approach for training data attribution in music generative models using machine unlearning. We apply unlearning techniques to a text-to-music diffusion model trained on a large-scale in-house dataset, and conduct a grid search by unlearning training data itself to identify configurations suitable for attribution. We compare the unlearning-based results with other attribution methods on generated samples, finding that, while the unlearning and others show similar trends, their attribution patterns differ. This work provides a framework for applying unlearning-based attribution to music generation models at scale.

References

[1] Mehul Agarwal, Gauri Agarwal, Santiago Benoit, Andrew Lippman, and Jean Oh. Secure & personalized music-to-video generation via charcha. In *Neural Information Processing Systems (NeurIPS) Creative AI Track*, 2024.

[2] Julia Barnett, Hugo Flores-Garcia, and Bryan Pardo. Exploring musical roots: Applying audio embeddings to empower influence attribution for a generative music model. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2024. URL `https://arxiv.org/abs/2401.14542`.

[3] 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.

[4] Roser Batlle-Roca, Wei-Hsiang Liao, Xavier Serra, Yuki Mitsufuji, and Emilia Gómez. Towards assessing data replication in music generation with music similarity metrics on raw audio. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 1004–1011, 2024. doi: 10.5281/ZENODO.14877501. URL `https://doi.org/10.5281/zenodo.14877501`.

[5] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[6] 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.

[7] Junwei Deng, Shiyuan Zhang, and Jiaqi Ma. Computational copyright: Towards a royalty model for music generative ai. *arXiv preprint arXiv:2312.06646*, 2023.

[8] Junwei Deng, Ting-Wei Li, Shichang Zhang, and Jiaqi Ma. Efficient ensembles improve training data attribution. *arXiv preprint arXiv:2405.17293*, 2024.

[9] Zach Evans, CJ Carr, Josiah Taylor, Scott H Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In *International Conference on Machine Learning (ICML)*, 2024.

[10] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[11] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2426–2436, 2023.

[12] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

[13] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[14] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=r1lYRjC9F7`.

[15] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, and Douglas Eck. Music transformer: Generating music with long-term structure. *arXiv preprint arXiv:1809.04281*, 2018.

[16] Yi-Lin Jiang, Chia-Ho Hsiung, Yen-Tung Yeh, Lu-Rong Chen, and Bo-Yu Chen. AI track-mate: Finally, someone who will give your music more than just “sounds great!”. In *Neural Information Processing Systems (NeurIPS) Creative AI Track*, 2024.

[17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

[18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[19] Myeongseob Ko, Feiyang Kang, Weiyan Shi, Ming Jin, Zhou Yu, and Ruoxi Jia. The mirrored influence hypothesis: Efficient data influence estimation by harnessing forward passes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26286–26295, 2024.

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

[21] Byung-Kwan Lee, Junho Kim, and Yong Man Ro. Masking adversarial damage: Finding adversarial saliency for robust and sparse network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15126–15136, June 2022.

[22] Yijun Li, Richard Zhang, Jingwan (Cynthia) Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15885–15896. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b6d767d2f8ed5d21a44b0e5886680cb9-Paper.pdf.

[23] Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, Elio Quinton, György Fazekas, and Juhan Nam. The song describer dataset: a corpus of audio captions for music-and-language evaluation. In *Machine Learning for Audio Workshop at NeurIPS 2023*, 2023.

[24] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*, 2023.

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[26] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

[27] Joan Serrà, R Oguz Araz, Dmitry Bogdanov, and Yuki Mitsufuji. Supervised contrastive learning from weakly-labeled audio segments for musical version matching. In *International Conference on Machine Learning (ICML)*, 2025.

[28] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HtMXRGbUMt.

[29] Sheng-Yu Wang, Aaron Hertzmann, Alexei A Efros, Jun-Yan Zhu, and Richard Zhang. Data attribution for text-to-image models by unlearning synthesized images. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=kVr3L73pNH.

[30] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[31] 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.

[32] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[33] Yixiao Zhang, Akira Maezawa, Gus Xia, Kazuhiko Yamamoto, and Simon Dixon. Loop copilot: Conducting ai ensembles for music generation and iterative editing. *arXiv preprint arXiv:2310.12404*, 2023.