Understanding Diffusion Memorization via Complexity of the Analytical Score Target

Justin Jung[*]

Abstract

Diffusion models can learn to generalize and generate novel samples when trained on finite datasets, despite having a closed form analytic solution to the denoising score matching objective that would only return memorized samples in the training set. We show that the complexity of the supervision target—the analytical score—elucidates the diffusion models' failure to memorize. We highlight the role that the euclidean geometry of the training dataset and the diffusion objective plays in diffusion memorization and we exploit this understanding to intervene and control the diffusion model's memorization behavior.

1 Introduction

That diffusion models can learn to generalize and generate novel samples when being trained on a *finite* number of samples is quite mysterious.

The denoising score matching (DSM) training objective the model is trained under has a closed-form analytic solution. Specifically, if we write the forward corruption process $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ for $\epsilon \sim \mathcal{N}(0, I)$ where $\mathbf{x}$ is a clean training sample, then for a finite dataset of samples $D = \{\mathbf{x}^{(i)}\}$ we can write our denoising score matching training objective

$$L_{\text{DSM}}(t) = \mathbb{E}_{\mathbf{x}\sim\text{Unif}(D),\, \epsilon\sim\mathcal{N}(0,I)} \left[\| s_\theta(\mathbf{z}_t, t) - \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t) \|^2 \right], \tag{1}$$

and by Tweedie's formula and denoising score matching we can re-express our objective to:

$$L_{\text{DSM}}(t) = \mathbb{E}_{\mathbf{x}\sim\text{Unif}(D),\, \epsilon\sim\mathcal{N}(0,I)} \left[\left\| s_\theta(\mathbf{z}_t, t) - \left(\frac{-\epsilon}{\sigma_t} \right) \right\|^2 \right]. \tag{2}$$

Notably, our training objective is an L^2 loss over a finite number of training examples and we can write out the corresponding analytic solution, which is a vector pointing towards a weighted sum of the training samples:

$$s_\star(\mathbf{z}_t, t) = \frac{1}{\sigma_t^2} \left[-\mathbf{z}_t + \alpha_t \sum_{i=1}^{N} \text{softmax}\left(\frac{-\|\mathbf{z}_t - \alpha_t \mathbf{x}^{(i)}\|^2}{2\sigma_t^2} \right) \mathbf{x}^{(i)} \right]. \tag{3}$$

Now if our model s_θ learns to solve this L^2 regression loss perfectly and matches the analytic score $s_\star$, then reverse sampling with our model s_θ would only output memorized training samples $\mathbf{x}^{(i)}$.

[*] Email: justinsoljung@gmail.com.

However, trained diffusion models tend not to return training samples but generate novel samples, so clearly our model's learned score is diverging from the analytical score $s_\star$.

What is the cause for this divergence? What controls whether the model memorizes and outputs training samples or generalizes and generates novel samples?

Our contributions.

1. We aim to show that we can understand diffusion models' failure to memorize through the complexity of the analytic target score $s_\star$.
2. This simple lens allows us to see how the "euclidean geometry" of the training data distribution (i.e., thinking about the spacing of points in high-dimensional euclidean space) can control this memorization behavior.
3. Moreover, through the complexity metric, we can distinguish and predict which generation trajectories will lead to memorization behavior.
4. And finally, we can intervene on the generation process by modulating the complexity metric to promote or prevent our model from memorizing.

2 Related Work

There is an extensive body of work on why diffusion models memorize or generalize. Many of them present ideas related to ones presented here. Mentioning a select few:

Scarvelis et al. (2025) and Bertrand et al. (2025) highlight that diffusion/flow models have a closed-form analytic solution to the denoising training objective. While just sampling from the analytic solution would return training samples, Scarvelis et al. (2025) motivated by the smoothness inductive bias of neural networks smooth the analytical solution to generate interpolated barycenters of training samples. Bertrand et al. (2025) show that in high-dimensional space the target analytical score is mainly explained by a single training point, making the target non-stochastic. They characterize generalization as being determined early on in generation and show that it is possible and effective to train the denoising score matching objective with the analytic target defined by the training dataset.

Song et al. (2025) show that the distribution of the input noised data $\mathbf{z}_t$ that the model trains on is quite narrow (thin shells around training samples) and often non-overlapping, leading to a trivial analytical score target for most time steps t. They posit that generalization comes from querying the model on an "extrapolation region" beyond the thin shells that the model is trained on. They also show that this extrapolation region only exists for the first few sampling steps—partial denoising starting from a noised latent $\mathbf{z}_t$ at some intermediate time t can induce the model to just return the corresponding clean training sample $\mathbf{x}$ that the latent $\mathbf{z}_t$ was noised from.

Aithal et al. (2024) show that the learned score of diffusion models are smoother than the analytical score and also show that diffusion models have a behavior of interpolating between data modes. Chen (2025) investigate the behavior of two-layer MLPs and show that their learned scores are smoothed versions of the analytical score. Moreover, they show that it is possible to generate interpolated data via smoothing the analytic score. Buchanan et al. (2025) characterize when a trained denoiser memorizes based on the training losses of surrogate denoisers. Zhang et al. (2024) analyze memorization behavior as a function of model size and training dataset size

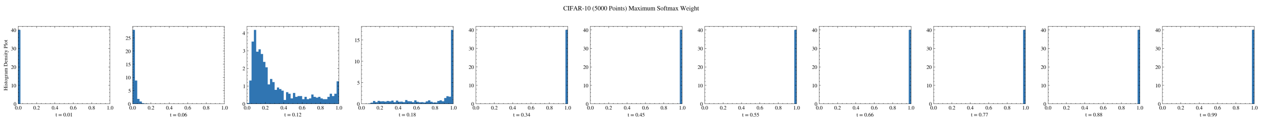

Figure 1: Maximum weight of our weighted sum in the target score definition (CIFAR-10, 5000 points).

and characterize a memorization and generalization regime based on the training dataset size relative to the model capacity.

3 Viewing Memorization from the Lens of "Euclidean Geometry"

When we look at the definition of our analytical score, we can understand how the "euclidean geometry" of our training dataset and the diffusion objective impacts model memorization.

Our target score $s_\star(\mathbf{z}_t, t)$ from Eq. (3) is a vector that points our noised $\mathbf{z}_t$ to a weighted average of training points $\mathbf{x}^{(i)}$, where the weight is defined by the distance to the scaled training point $\|\mathbf{z}_t - \alpha_t \mathbf{x}^{(i)}\|^2$.

Now the reality is that for most timesteps t, our noised $\mathbf{z}_t$ is a lot closer to one scaled training point $\alpha_t \mathbf{x}^{(i)}$ relative to the others, such that our weighted sum of training points collapses to effectively a single training point, making our target score a vector pointing towards the nearest training point. We can see this early collapse of the analytic target in Figure 1: early on with t around ≈ 0.2 our analytic score $s_\star$ has a sum that is composed of effectively one training point. Here we use flow matching convention (and run our experiments with flow models as in Bertrand et al. (2025)) so $t \to 1$ approaches the data distribution.

Now we can understand this collapse behavior both from the distribution of our corrupted $\mathbf{z}_t$ and also the euclidean distance spacing of our dataset $\mathbf{x}^{(i)}$.

In high-dimensional euclidean space, with high probability the points are far from each other (a result from the curse of dimensionality). Thus when thinking about the spacing of our training datapoints $\mathbf{x}^{(i)}$ (CIFAR-10 images $\mathbb{R}^{3\times 32\times 32} = \mathbb{R}^{3072}$), they end up being spaced far from each other in terms of the L^2 norm.

We can also consider the distribution of our corrupted input $\mathbf{z}_t$ which is sampled from a noised mixture of Gaussians centered around our scaled training points, $\mathbf{z}_t \sim \hat{p}_t(\cdot) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}^{(i)}, \sigma_t^2 I)$. Looking at a single Gaussian component $\mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}^{(i)}, \sigma_t^2 I)$, chi-squared tail bounds tell us that with high probability a point sampled from this Gaussian lies on a thin shell of radius approximately $\sigma_t \sqrt{d}$ where d is the data dimensionality. Previously we noted that in high-dimensional space our training points $\mathbf{x}^{(i)}$ are spaced far apart from each other. With the exception of small t, where $\alpha_t \to 0$ forcibly pushes our scaled data points $\alpha_t \mathbf{x}^{(i)}$ towards the origin, we have that these scaled data points are spaced far apart, notably further than $\sigma_t \sqrt{d}$ such that our mixture of Gaussians is a mixture of non-overlapping thin shells in high-dimensional space. Then our corrupted points $\mathbf{z}_t$ that we train on are close to a single Gaussian $\mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}^{(i)}, \sigma_t^2 I)$ while apart from the others.

Now looking again at the analytic target score definition we can see why our target collapses to a simple vector pointing to the nearest training point. For most t, with high probability $\mathbf{z}_t$—lying on a thin shell around one Gaussian—is much closer to a single $\alpha_t \mathbf{x}^{(i)}$, causing the weighted sum to basically be a single training point $\mathbf{x}^{(i)}$ and thus reducing our target score to a trivial vector pointing towards the nearest training point. This makes the denoising score matching objective a

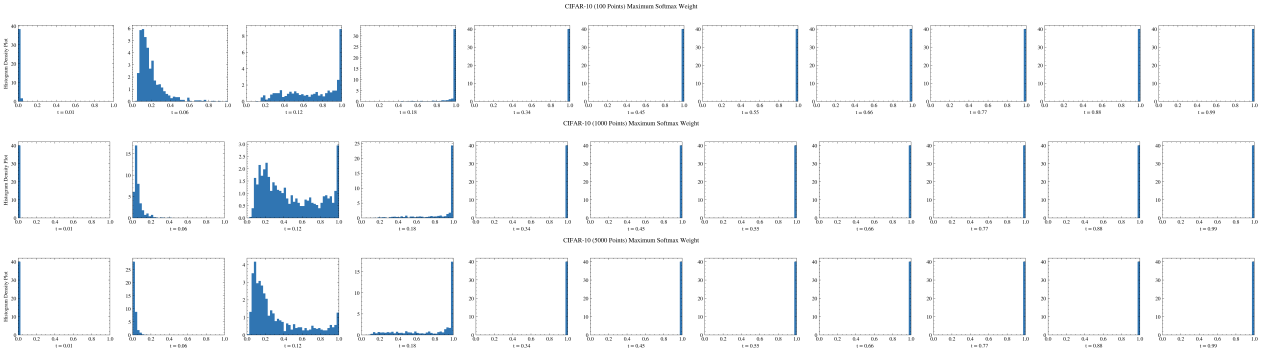

Figure 2: Maximum weight of the weighted sum in the target score definition across CIFAR-10 dataset sizes.

trivial supervision problem of mapping $\mathbf{z}_t$ to the nearest training point, something that the model can easily memorize and overfit to.

Understanding the connection between training set size and memorization behavior. Previous work have characterized the relationship between training dataset size and memorization behavior; they show memorization behavior decreases as the training dataset size increases relative to the model capacity. We can understand why memorization decreases with more training points from the euclidean geometry viewpoint; as we increase the number of training samples (keeping the dimensionality of the space fixed), the probability of scaled training points $\alpha_t \mathbf{x}^{(i)}$ having a nearby neighbor increases, making our analytical target score composed of multiple training points, rather than a single training point, thus preventing the target from collapsing to a trivial vector pointing towards the nearest training point. In fact in Figure 2 we can directly see the impact of training dataset size on the complexity of the analytic target. With more training samples, the analytic score tends to have smaller maximum weight in its weighted sum, meaning that the sum is composed of more terms.

4 Predicting Memorization Behavior via Complexity of Our Analytic Target

So far we have shown that our score target during training can collapse to a trivial vector pointing towards the nearest training point, making it easy for our model to memorize the supervision target. However, our samples aren't generated by querying our model $s_\theta(\mathbf{z}_t, t)$ for some noised training point $\mathbf{z}_t \sim \hat{p}_t(\cdot)$; rather they are generated by inputting some reverse sampled latent $\mathbf{z}_t$ which starts from the base distribution, typically isotropic Gaussian $\mathcal{N}(0, \mathbf{I})$. Indeed we see that we can use the complexity of the analytic target score during generation to analyze generated trajectories and distinguish memorizing vs non-memorizing trajectories.

Previous work such as Bertrand et al. (2025) have shown that generalization behavior is determined early on in generation. In Figure 3 we see a similar phenomenon. We plot on the y-axis the score divergence between our model score and the analytical target score $\|s_\theta - s_\star\|^2$ (normalized across time by multiplying by σ_t^2). We partition our generated trajectories based on whether they memorize or not (we adopt the strict memorization criteria as in Buchanan et al. (2025) and say that a sample $\hat{\mathbf{x}}$ is memorized if the distance to the nearest training point is smaller

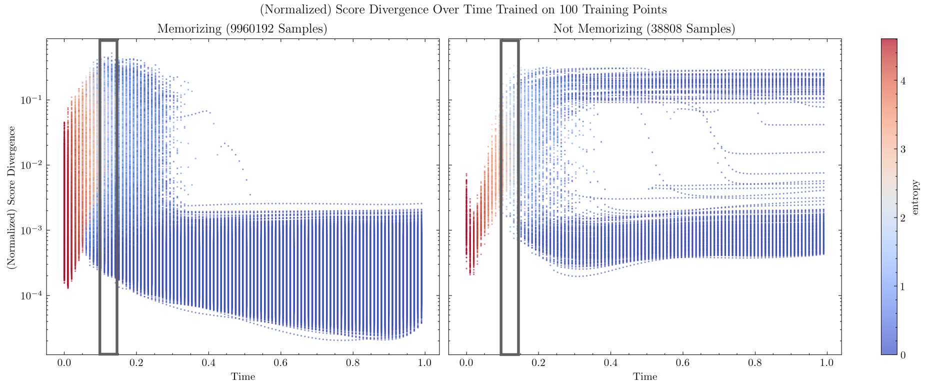

Figure 3: Entropy of analytic target sum weights for memorizing and non-memorizing trajectories.

than $c = \frac{1}{9}$ the distance to the second nearest training point, i.e., $\|\hat{\mathbf{x}} - \mathbf{x}^{(1)}\| < \frac{1}{9}\|\hat{\mathbf{x}} - \mathbf{x}^{(2)}\|$). We can see that memorized trajectories have a different behavior than non-memorized trajectories early on, $t < 0.2$. We highlight the $t \in [0.1, 0.15]$ region as an example and can see that memorized trajectories typically have a much lower complexity of the analytic score target evaluated at the current generation $s_\star(\mathbf{z}_t, t)$. This is color-coded such that cooler colored dots correspond to lower complexity of the analytic score target (as measured by the entropy of the weights distribution); we can see that the memorized trajectories have a lot more cooler colored dots in the $t \in [0.1, 0.15]$ region versus the non-memorized trajectories and also have a much lower divergence from the analytic target score.

5 Intervening on Memorization Behavior

In fact we can intervene on our generated trajectories and modulate the model's memorization behavior, even when starting from a partially noised $\mathbf{z}_t \sim \hat{p}_t(\cdot)$. Notably, this experiment cannot be readily explained by the perspective of Song et al. (2025), which posits that generalization arises from early reverse generation steps that are outside the training supervision region $\hat{p}_t$. We choose different starting timesteps $t \in [0.1, 0.2, 0.3, 0.5, 0.9]$ to start partial denoising from. Here we use a model trained under 100 CIFAR-10 training samples. Now looking back at the distribution of the maximum weight in the sum of the analytical score target, we see that for $t > 0.18$ the sampled $\mathbf{z}_t \sim \hat{p}_t(\cdot)$ have an analytic target $s_\star(\mathbf{z}_t, t)$ effectively defined by a single training point, i.e., a trivial supervision target. As expected, if we run reverse generation from $t = 0.2$ we have that 97.6% of $n = 9875$ generated samples are memorizing.

However, we can intervene on the starting point and select for a starting point of a larger or smaller complexity of the analytic target. Specifically, we select a starting $\mathbf{z}_t \sim \hat{p}_t(\cdot)$ that has the maximum or minimum entropy of the analytic score sum weights across a sampled set of 10,000 $\mathbf{z}_t$ and do reverse generation from that extrema starting point $\mathbf{z}_t$. As shown in Figure 4, by intervening on the target complexity of the starting point we can induce either more or less memorization. Notably, even at $t = 0.3$, by starting from a point with high analytic score target complexity, we end up having some fraction of samples that don't memorize (6.4% of samples versus 0% of samples with normal generation).

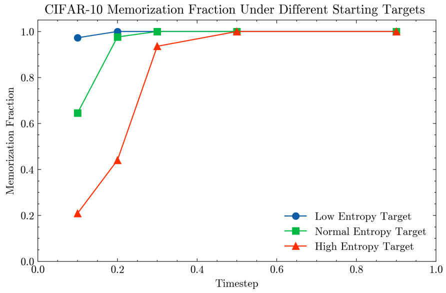

Figure 4: Intervening on starting target complexity to induce/reduce memorization behavior.

6 Conclusion and Future Work

In this work we hope to highlight how properties stemming from the euclidean geometry of our training dataset and our diffusion formulation impact diffusion memorization.

From this, we think that there are some interesting follow-up investigations. First, the literature has characterized the role of increasing training dataset sizes on diffusion memorization. However, we believe that the number of training points is just a general yardstick and more importantly the spacing of the training dataset should be considered. For example, for the same training dataset size, a dataset with points spaced further apart will induce the supervision analytic score target to be more trivial, while a dataset of points on a compact subspace would make the supervision target more complex. This may motivate understanding the multiple roles a compressing autoencoder (such as a VAE) in latent diffusion plays—not just on speeding up computation, but also on regulating memorization via a compact subspace.

Moreover, we have seen that generalization behavior is determined early on. This may inform training or sampling strategies that induce a more complex supervision target early on in generation, such as in coarse-to-fine generation diffusion models.

References

Aithal, S. K., Maini, P., Lipton, Z. C., and Kolter, J. Z. (2024). Understanding hallucinations in diffusion models through mode interpolation.

Bertrand, Q., Gagneux, A., Massias, M., and Emonet, R. (2025). On the closed-form of flow matching: Generalization does not arise from target stochasticity.

Buchanan, S., Pai, D., Ma, Y., and Bortoli, V. D. (2025). On the edge of memorization in diffusion models.

Chen, Z. (2025). On the interpolation effect of score smoothing in diffusion models.

Scarvelis, C., de Ocáriz Borde, H. S., and Solomon, J. (2025). Closed-form diffusion models.

Song, K., Kim, J., Chen, S., Du, Y., Kakade, S., and Sitzmann, V. (2025). Selective underfitting in diffusion models.

Zhang, H., Zhou, J., Lu, Y., Guo, M., Wang, P., Shen, L., and Qu, Q. (2024). The emergence of reproducibility and generalizability in diffusion models.