Understanding Hallucinations in Diffusion Models through Mode Interpolation

Sumukh K Aithal [1] **Pratyush Maini** [1 2] **Zachary C Lipton** [1]

Abstract

Colloquially speaking, image generation models based upon diffusion processes are frequently said to exhibit "hallucinations," samples that could never occur in the training data. But where do such hallucinations come from? In this paper, we study a particular failure mode in diffusion models, which we term *mode interpolation.* Specifically, we find that diffusion models smoothly "interpolate" between nearby data modes in the training set, to generate samples that are completely outside the support of the original training distribution; this phenomenon leads diffusion models to generate artifacts that never existed in real data (i.e., hallucinations). We systematically study the reasons for, and the manifestation of this phenomenon. Through experiments on 1D and 2D Gaussians, we show how a discontinuous loss landscape in the diffusion model's decoder leads to a region where any smooth approximation will cause such hallucinations. Through experiments on artificial datasets with various shapes, we show how hallucination leads to the generation of combinations of shapes that never existed. Finally, we show that diffusion models in fact *know* when they go out of support and hallucinate. This is captured by the high variance in the trajectory of the generated sample towards the final few backward sampling process. Using a simple metric to capture this variance, we can remove over 95% of hallucinations at generation time while retaining 96% of in-support samples. We conclude our exploration by showing the implications of such hallucination (and its removal) on the collapse (and stabilization) of recursive training on synthetic data with experiments on MNIST dataset.

[1]Carnegie Mellon University [2]DatologyAI. Correspondence to: Sumukh K Aithal <saithal@cs.cmu.edu>, Pratyush Maini <pratyushmaini@cmu.edu>. We release our code at https://github.com/locuslab/diffusion-model-hallucination
Accepted as an extended abstract for the Geometry-grounded Representation Learning and Generative Modeling Workshop at the 41^{st} International Conference on Machine Learning, ICML 2024, Vienna, Austria.

1. Introduction

The high quality and diversity of images generated by diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have made them the de facto standard generative models across various tasks including video generation (Brooks et al., 2024), image inpainting (Lugmayr et al., 2022), image super-resolution (Gao et al., 2023), data augmentation (Trabucco et al., 2023), and others. As a result of their uptake, large volumes of synthetic data are rapidly proliferating on the internet. The next generation of generative models will likely be exposed to many machine-generated instances during their training, making it crucial to understand ways in which diffusion models fail to model the true underlying data distribution. Like other generative model families, much research has been done to understand the failure modes of diffusion models as well. Past works have identified, and attempted to explain and remedy failures such as, training instabilities (Huang et al., 2023), memorization (Carlini et al., 2023; Somepalli et al., 2023) and inaccurate modeling of objects such as hands and legs (Narasimhaswamy et al., 2024; Borji, 2023).

In this work, we formalize and study a particular failure mode of diffusion models that we call hallucination — a phenomenon where diffusion models generate samples that lie completely out of the support of the training distribution of the model. We observe hallucinations even in large generative models like StableDiffusion (Rombach et al., 2021) in the form of hands with extra (or missing) fingers or limbs. We begin our investigation with a surprising observation that an unconditional diffusion model trained on a distribution of simple shapes, generates images with combinations of shapes (or artifacts) that never existed in the original training distribution (Figure 1). While a lot of research in generative models has studied the phenomenon of 'mode collapse' (Zhang et al., 2018) which leads to a loss in diversity in the sampled distribution, in our work we explain hallucinations through the lens of a phenomenon we call 'mode interpolation', which has not been studied in past work to the best of our knowledge.

To understand the cause of these hallucinations and their relationship to mode interpolation, we construct simplified 1-d and 2-d mixture of Gaussian setups and train diffusion models on them (§ 2). We observe that when the true data distribution occurs in disjoint modes, diffusion models are unable to model a true approximation of the underlying distribution. This is because there exist 'step functions' between different modes, but the score function learned by the DDPM is a smooth approximation of the same, leading to interpolation between the nearest modes, even when these interpolated values are entirely absent from the training data. Our investigation on the occurrence of mode interpolation leads us to the observation that hallucinated samples usually have very high variance towards the end of their trajectory during the reverse diffusion process. Based on this observation, we use the trajectory variance during sampling as a metric to detect hallucinations (§ 3), and show that diffusion models usually 'know' when they hallucinate, allowing detection with sensitivity and specificity > 0.92.

Finally, we study the implications of this phenomenon in recursive generative model retraining where we train generative models on their own output (§ C). Recently, recursive training and its downsides in model collapse have garnered a lot of attention in both language and diffusion modeling literature (Alemohammad et al., 2023; Bertrand et al., 2023; Dohmatob et al., 2024; Briesch et al., 2023). We observe that the proposed detection mechanism is able to mitigate the model collapse during recursive training on 2D Grid of Gaussians, Shapes and MNIST dataset.

Hallucination in Diffusion Models. Before formalizing our notions and definitions in § B, let us first consolidate the observation that has been loosely labeled as 'hallucination' until now. To illustrate this phenomenon, we design a synthetic dataset called SIMPLE SHAPES, and train a diffusion model to learn its distribution.

SIMPLE SHAPES Setup. Consider a dataset consisting of black and white images that contain three shapes: triangle, square, and pentagon. Each image in the dataset is 64x64 pixels in size and divided into three (implied) columns. The first, second, and third columns contain a triangle, square, and pentagon, respectively. Each column has a 0.5 probability of containing the corresponding shape. A representation of this setup is shown in Fig 1. It is important to note that in this data generation pipeline, each shape is present at most once in each image.

Observation. We train an unconditional Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) on this toy dataset with $T = 1000$ timesteps. We observe that the DDPM generates a small fraction of images that are never observed in the training dataset, nor a part of the 'support' of the data generation pipeline. Specifically, the model generates some images that contain two occurrences of the same shape, as shown in Fig 1. Furthermore, when the model is iteratively trained on its own sampled data, the fraction of these occurrences increases significantly as the generation process progresses.

Inspired by these observations and their implications, we will perform experiments through the rest of this work to formalize what we mean by hallucinations (§ B), why do they occur (§ 2), how can we mitigate them (§ 3), and what are their implications for real-world datasets (§ C).

2. Understanding Mode Interpolation and Hallucination

In this section, we provide initial investigations into the central phenomenon of hallucinations in diffusion models. Formally, we consider a hallucination to be a generation from the model that lies entirely outside the support of the real data distribution (or, for distributions that theoretically have full support, in a region with negligible probability). That is, the ϵ-Hallucination set $H_\epsilon(q)$

$$H_\epsilon(q) = \{x : q(x) \leq \epsilon\}, \tag{1}$$

where we typically take $\epsilon = 0$ or take ϵ to be vanishingly small (well beyond numerical precision). We similarly define the ϵ-support set $S_\epsilon(q)$ to simply be the complement of the ϵ-Hallucination set.

Mode interpolation occurs when a model generates samples that directly *interpolate* (in input space) between two samples in the ϵ-support set, such that the interpolation lies in the ϵ-Hallucination set. That, is for $x, y \in S_\epsilon(q)$ the model generates $\theta x + (1-\theta) y \in H_\epsilon(q)$. The main argument of this paper, shown through examples and numerical analysis of special cases, is that diffusion models frequently exhibit mode interpolation between "nearby" modes in the data distributions, and such interpolation leads to the generation of artifacts that did not exist in the original data (hallucinations).

1D GAUSSIAN Setup. We have already seen how hallucinations manifest in the SIMPLE SHAPES set-up (§ 1). To investigate hallucinations via mode interpolation, we begin with a synthetic toy dataset characterized by a mixture of 1D Gaussians given by: $p(x) = \frac{1}{3}\mathcal{N}(\mu_1, \sigma^2) + \frac{1}{3}\mathcal{N}(\mu_2, \sigma^2) + \frac{1}{3}\mathcal{N}(\mu_3, \sigma^2)$. For our initial experiments, we set $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3$ and $\sigma = 0.05$. We sample 50k training points from this true distribution and train an unconditional DDPM using these samples with $T = 1000$ timesteps for $10,000$ epochs. Additional experimental details are present in the App. E.

We observe that diffusion models can generate samples that interpolate between the two nearest modes of the mixture of Gaussians (Fig. 2). To clearly observe the distribution of these interpolated samples, we generated 100 million samples from the diffusion models. The probability of sampling from the interpolated regions (regions outside the support of real data density, outlined in red) is non-zero, and decays with the distance from the modes. This region has nearly 0

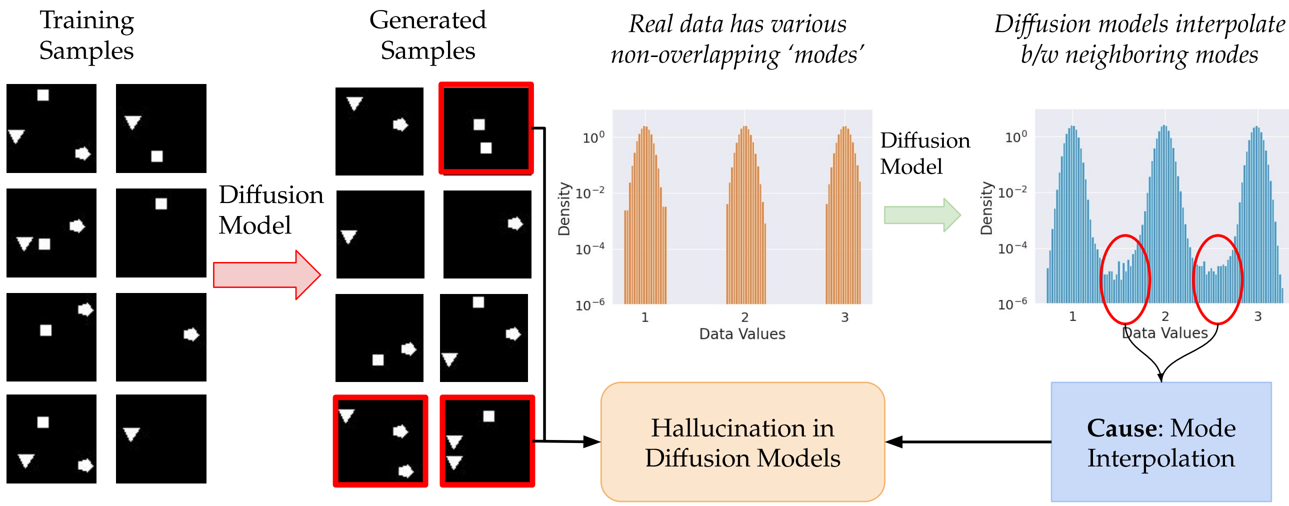

Figure 1: **Hallucinations in Diffusion Models**: Original Dataset (Left) & Generated Dataset (Right). The original dataset consists of 64x64 images divided into three columns, each containing a triangle, square, or pentagon with a 0.5 probability of the shape being present. Each shape appears at most once per image. The generated dataset created using an unconditional DDPM includes some samples (*hallucinations*) with multiple occurrences of the same shape, unseen in the original dataset.

probability mass of the true distribution, and no samples in this region occurred in the data used to train the DDPM.

The rate of mode interpolation depends primarily on three factors: **(i)** Number of training data points, **(ii)** variance of (and distance between) the distributions, and **(iii)** the number of sampling timesteps (T). As the number of training samples increases, we observe that the proportion of interpolated samples decreases. In this setup, the variance of $p(x)$ not only depends on σ but also the distance between the modes i.e, $|\mu_1 - \mu_2|$ and $|\mu_2 - \mu_3|$. We run another experiment with $\mu_1 = 1, \mu_2 = 2$ and $\mu_3 = 4$. In this case, we observe that the frequency of samples between μ_2 and μ_3 is much lower than μ_1 and μ_2. The number of interpolated samples also decreases as the distance from the modes increases. The frequency of interpolated samples is also proportional to timesteps T. Additional experiments with different numbers of Gaussians are presented in Appendix G.

2.1. 2D Gaussian Grid

The reduction in density of mode interpolation as two modes with $\mu = [2, 3]$ are moved apart calls for closer inspection into when and how diffusion models choose to interpolate between nearby modes. To investigate this, we make a toy dataset with a mixture of 25 Gaussians arranged in a two-dimensional square grid. A total of 100,000 samples are present in the training set. Similar to the 1D case, we observe interpolated samples between the two nearest modes of the Gaussian. Again, these samples have close to zero probability if sampled from the original distribution (Fig 8).

We note that mode interpolation only happens between the nearest neighbors. To demonstrate this occurrence, we also train a DDPM on the rotated version of the dataset where the modes are arranged in the shape of a diamond (Figure 8.c,d). The mode interpolation can be more clearly observed in this setting. Interestingly, there appears to be no interpolation between modes along the x-axis, indicating that only the nearest modes are being interpolated. We believe this empirical observation of mode interpolation being confined to nearby modes will spark further investigation.

What causes mode interpolation?. To understand the reason behind the observed mode interpolation, we analyze the score function learned by the model. The model learns to predict ϵ_θ which is related to the score function as $s_\theta(x_t, t) = -\frac{-\epsilon_\theta(x_t,t)}{\sqrt{1-\bar{\alpha}_t}}$. We know the true score function for the given mixture of Gaussians, and we can estimate the learned score function using the model's output. In Figure 4, we plot the ground truth score (left) and the learned score (right) across various timesteps. We observe that the neural network learns a smooth approximation of the true score function, particularly around the regions between disjoint modes of the distribution from timesteps $t = 0$ to $t = 20$. Notice that the true score function has sharp jumps that separate two modes, however, the neural network can not learn such sharp functions and smoothly approximates a tempered version of the same. We also plot the estimated $\hat{x_0}$ and observe a smooth approximation of the step function instead of the exact step function. There is a region of uncertainty in the region between the two modes which leads to mode interpolation i.e sampling in the regions between the two modes. As another sanity check, we used the true

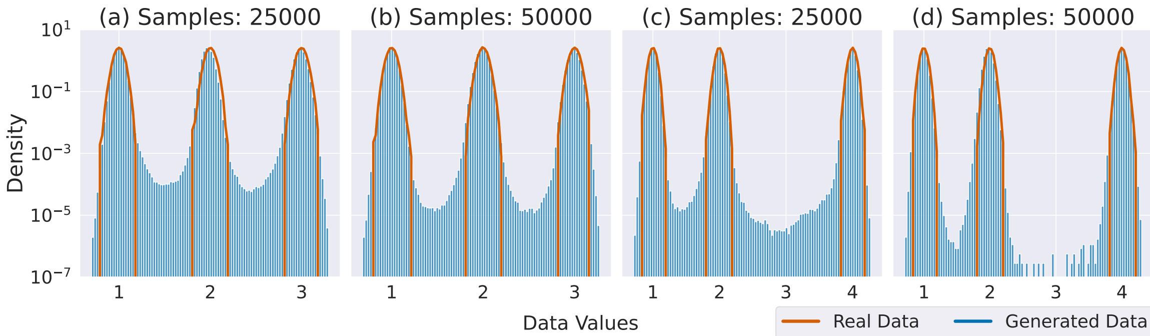

Figure 2: **Mode Interpolation in 1D GAUSSIAN**. The red curve indicates the PDF of the true data distribution $q(x)$, which is a mixture of 3 Gaussians (notice that the y-axis is in log-scale). In blue, we show a density histogram of the samples generated by a DDPM trained on varying number of samples from the true data distribution. For each histogram, we sampled 100 million examples from the diffusion model to observe the interpolated distribution. **(a,b)** show how the density of samples generated in the interpolated region reduces with an increase in the number of samples from the real distribution (used for training the DDPM). **(c,d)** show the impact of moving one of the modes (originally at $\mu = 3$) to $\mu = 4$. The density of samples in the region between distant (but neighboring) modes is significantly lesser than that between nearby modes.

score function in the reverse diffusion process for sampling (instead of the learned network). In this case, we did not see any instance of mode interpolation. This explains why the diffusion model generates samples between two modes of a Gaussian when it was never in the training distribution.

3. Diffusion Models know when they Hallucinate

Our previous sections established that hallucinations in diffusion models arise during sampling. More specifically, intermediate samples land in regions between different modes where the score function has high uncertainty. Since neural networks find it hard to learn discrete 'jumps' between different modes (or a perfect step function), they end up interpolating between different modes of the distribution. This understanding suggests that the trajectory of the samples that generate hallucinations must have high variance due to the highly steep score function in the region of uncertainty. We will build upon this intuition to identify hallucinations in diffusion models.

Variance in the trajectory of prediction. We revisit the hallucinated samples in the 1D GAUSSIAN setup, and examine the trajectory of the predicted value of $\hat{x_0}$ during the reverse diffusion process. Figure 14 depicts the variance of trajectories leading to hallucinations (red shades) and those generating samples within the original data distribution (blue shades). For trajectories in shades of blue (non-hallucinations), the variance remains low beyond $t = 20$. This indicates there is a minimal change in the predicted $\hat{x_0}$ during the final stages of reverse diffusion, signifying convergence. Conversely, the red trajectories (hallucinations) exhibit instability in the value of $\hat{x_0}$ in the same region suggesting a high overall variance in these trajectories.

Metric for detecting hallucination. Based on the above observation about high variance in predicted values of x_0 in the reverse diffusion process, we use the same observation as a metric to distinguish hallucinated and non-hallucinated (in-support) samples. Mathematically, the metric can be defined as follows: $\mathtt{Hal}(x) = \frac{1}{T}\sum_{i=0}^{T}(\hat{x_0}^{(t)} - \overline{\hat{x_0}^{(t)}})^2$ where $\hat{x_0}^{(t)}$ represents the predicted values of the final image at different time steps (t), and $\overline{\hat{x_0}^{(t)}}$ is the mean of these predictions over the same time steps. We now utilize this metric to analyze the histogram values of each sample from the three experimental setups studied thus far.

SIMPLE SHAPES. In the SIMPLE SHAPES setup, a sample is labeled as hallucinated if more than one shape of the same type occurs in the generated image. We generate 7500 images using a DDPM and study the separation between hallucinated and non-hallucinated images. We find that the reverse diffusion process of $T = 1000$ steps is rather long. Generally, the image stabilizes around $T = 700$ (as shown in Appendix 17). Therefore, we use the time range between $T = 850$ and $T = 700$ in the reverse diffusion process to compute the variance of the predicted sample value. Using this process, we can filter out 95% of the hallucinated samples while retaining 95% of the in-support samples. The histogram for the values is presented in Figure 5.

1D GAUSSIAN. In the 1D-Gaussian setup, we label any examples as a hallucination if they have negligible probability (for instance values greater than 6σ from the mean under normal) under the real data distribution (refer to Figure 2). We measure the variance of the last 15 steps of the $\hat{x_0}$ during the reverse diffusion process, and plot the histogram of values of the same in Figure 5. We can filter out 95% hallucinated samples while retaining 98% of in-support samples.

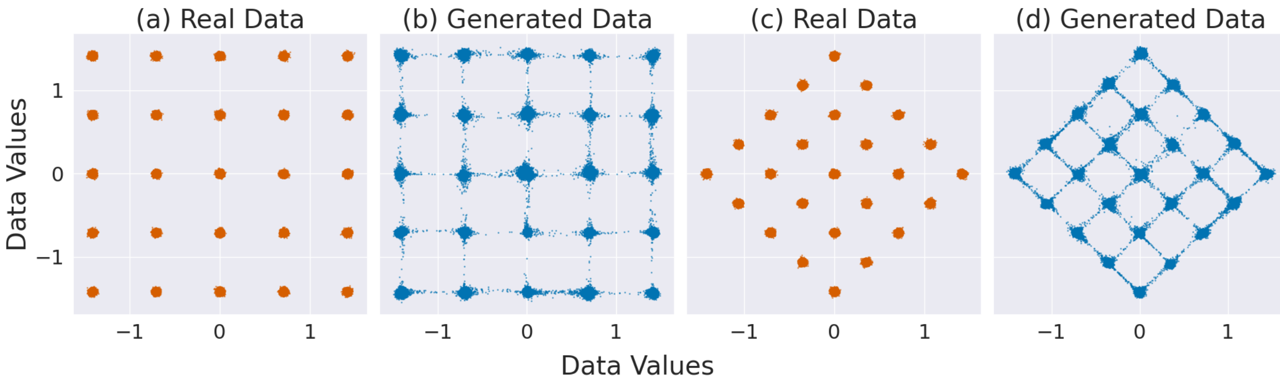

Figure 3: **Mode Interpolation in 2D GAUSSIAN**. The dataset consists of a mixture of 25 Gaussians arranged in a square grid, with a training set containing 100,000 samples. **(a,b)** The blue points represent samples generated by a DDPM, with visible density between the nearest modes of the original Gaussian mixture (in orange). These interpolated samples have near-zero probability in the original distribution. **(c,d)** We trained a DDPM on a rotated version of the dataset where the modes form a diamond shape. In this configuration, we see no interpolation along the x-axis, illustrating that diffusion models interpolate between nearest modes.

Figure 4: **Explaining Mode Interpolation via Learned Score Function.** The left panel shows the ground truth score function for a mixture of Gaussians across various timesteps, while the right panel illustrates the score function learned by the neural network. While the true score function exhibits sharp jumps that separate distinct modes (particularly in the initial time steps), the neural network approximates a smoother version.

We investigate the recursive generative model training setup in App C and the 2D Gaussians in App. D.

4. Discussion

We performed an in-depth study to formulate and understand hallucination in diffusion models, focusing on the phenomenon of mode interpolation. We saw how diffusion models learn smoothed approximations of disjoint score functions, leading to mode interpolation. Our analysis led to an effective metric for identifying hallucinated samples. We also explored the implications of hallucination in the context of recursive generative model training. Past works (Samuel et al., 2024) have shown that rare concepts/classes in large diffusion models like StableDiffusion (Rombach et al., 2021) are poorly modeled. This is evident from the distortion of hands commonly observed in samples generated by these models. The occurrence of 6-8 fingers is potentially analogous to the occurrence of 2 squares in the toy experiment, an exciting follow-up for future research.

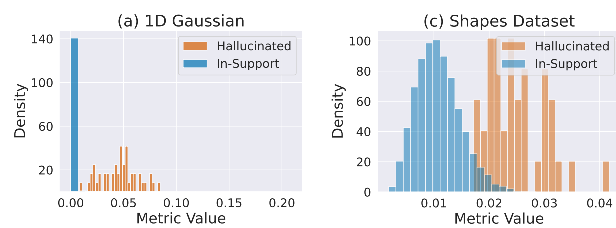

Figure 5: **Histogram of Hallucination Metric**. We depict the hallucination metric values for (a) 1D GAUSSIAN, (b) SIMPLE SHAPES setups. The histograms show that trajectory variance can capture a separation between hallucinated (orange) and non-hallucinated (blue) samples.

Acknowledgements

We thank J. Zico Kolter for his guidance and mentorship throughout this project. PM is supported by funding from the DARPA GARD program. ZL gratefully acknowledges the NSF (FAI 2040929 and IIS2211955), UPMC, Highmark Health, Abridge, Ford Research, Mozilla, the PwC Center, Amazon AI, JP Morgan Chase, the Block Center, the Center for Machine Learning and Health, and the CMU Software Engineering Institute (SEI) via Department of Defense contract FA8702-15-D-0002, for their generous support of ACMI Lab's research.

References

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*, 2023.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arXiv:2310.00429*, 2023.

Ali Borji. Qualitative failures of image generation models and their application in detecting deepfakes. *Image and Vision Computing*, 137:104771, 2023.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

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.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043*, 2024.

Shi Fu, Sen Zhang, Yingjie Wang, Xinmei Tian, and Dacheng Tao. Towards theoretical understandings of self-consuming generative models. *arXiv preprint arXiv:2402.11778*, 2024.

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10021–10030, 2023.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Zhongzhan Huang, Pan Zhou, Shuicheng Yan, and Liang Lin. Scalelong: Towards more stable training of diffusion model via scaling network long skip connection. *Advances in Neural Information Processing Systems*, 36: 70376–70401, 2023.

Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding ddpm latent codes through optimal transport. *arXiv preprint arXiv:2202.07477*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Qihao Liu, Adam Kortylewski, Yutong Bai, Song Bai, and Alan Yuille. Intriguing properties of text-guided diffusion models. *arXiv preprint arXiv:2306.00974*, 2023.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation? *arXiv preprint arXiv:2303.01255*, 2023a.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Towards understanding the interplay of generative artificial intelligence and the internet. In *International Workshop on Epistemic Uncertainty in Artificial Intelligence*, pages 59–73. Springer, 2023b.

Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. Handiffuser: Text-to-image generation with realistic hand appearances. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4695–4703, 2024.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European conference on computer vision (ECCV)*, pages 213–229, 2018.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021b.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023.

Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021. URL https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*, 2023.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

Zhaoyu Zhang, Mengyan Li, and Jun Yu. On the convergence and mode collapse of gan. In *SIGGRAPH Asia 2018 Technical Briefs*, pages 1–4. 2018.

A. Related Work

Diffusion Models. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) are a class of generative models characterized by a forward process and a reverse process. In the forward process, noise is incrementally added to an image over time steps, ultimately converting the data into noise. The reverse process learns to denoise the image using a neural network essentially learning to convert noise to data. Diffusion models have various interpretations. Score-based generative modeling (Song and Ermon, 2019; Song et al., 2021b) and DDPMs (Ho et al., 2020) are closely related, with (Song et al., 2020) proposing a unified framework using stochastic differential equations (SDEs) that generalizes both Score Matching with Langevin Dynamics (SMLD) (Song et al., 2020) and DDPM. In this framework, the forward process is a SDE with a continuous-time generalization instead of discrete timesteps and the reverse process is also an SDE that can be solved using a numerical solver. Another perspective is to view diffusion models as hierarchical Variational Autoencoders (VAEs) (Luo, 2022). Recent research (Khrulkov et al., 2022) suggests that diffusion models learn the optimal transport map between Gaussian distribution and data distribution. In this paper, we discover a surprising phenomenon in diffusion which we coin mode interpolation.

Recursive Generative Model Training. Recent works (Alemohammad et al., 2023; Shumailov et al., 2023; Martínez et al., 2023a;b; Bertrand et al., 2023) demonstrated that iteratively training the generative models on their own output (i.e recursive training) leads to model collapse. The model collapse can happen in two ways: either all samples collapse to a single mode (low diversity) or the model generates very low fidelity, unrealistic images (low sample quality). This has been shown in the visual domain with StyleGAN2 and diffusion models (Bertrand et al., 2023; Alemohammad et al., 2023), as well as in the text domain with Large Language Models (LLMs) (Shumailov et al., 2023; Briesch et al., 2023; Dohmatob et al., 2024). The current solution to mitigate this collapse is to include a fraction of real data in the training loop at all the generations (Bertrand et al., 2023; Alemohammad et al., 2023). Theoretical results have also proved that super-quadratic number of synthetic samples are necessary to prevent model collapse (Fu et al., 2024) in the absence of support from real data. A concurrent work (Gerstgrasser et al., 2024) studied the setup of data accumulation in recursive training where data from previous iterations of generative models together with real data are accumulated over time. The authors conclude that data accumulation (including real data) can avoid model collapse in various settings including language modeling and image data.

Past works have only studied the collapse of the generative model to the mode of the existing distribution. Through some controlled experiments, we study the interaction between different modes (a mode can be a class) or novel modes being developed in the generative models. This provides novel insights into the reasons behind the collapse of generative models during recursive training.

Failure Modes of Diffusion Models. One of the common failure modes of diffusion models is the generation of images where the hands and legs appear distorted or deformed which is commonly observed in Stable Diffusion (Rombach et al., 2021) and Sora (Brooks et al., 2024). Diffusion models also fail to learn rare concepts (Samuel et al., 2024) which have less than 10k samples in the training set. Various other failure modes including ignoring spatial relationships or confusing attributes have been discussed in (Liu et al., 2023; Borji, 2023).

Hallucination in Language Models. Hallucination in LLMs (Zhang et al., 2023; Ye et al., 2023) is a huge barrier to the deployment of LLMs in safety-critical systems. The LLMs may provide a factually incorrect output or incorrectly follow the instructions or be logically wrong. A simple example is that LLMs can generate new facts when asked to summarize a block of text (input-conflicting hallucination) (Zhang et al., 2023). Current hallucination mitigation techniques in LLMs include factual data enhancement (Gunasekar et al., 2023), retrieval augmentation (Ram et al., 2023) among other methods. Given the widespread adoption of image generation models, we argue that hallucination in diffusion models must also be studied carefully to identify its causes and mitigate it.

B. Definitions and Preliminaries

Let $q(x)$ be the real data distribution. We define a forward process where Gaussian noise is iteratively added at each timestep for a total of T timesteps. Let $x_0 \sim q(x)$, and x_t be the perturbed (noisy) sample after adding t timesteps of noise. The noise schedule is defined by $\beta_t \in (0, 1)$, which represents the variance of Gaussian (added noise) at time t. For large enough T, $x_T \sim \mathcal{N}(0, \mathbf{I})$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}); \tag{2}$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3}$$

In the forward diffusion process, we can directly sample x_t at any time step using the closed form $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$ where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{j=1}^{t} \alpha_j$.

The reverse diffusion process aims to learn the process of denoising i.e, learning $p_\theta(x_{t-1}|x_t)$ using a model (such as a neural network) with θ as the learnable parameters.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t); \tag{4}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{5}$$

The mean can be derived as $\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right)$ where $\epsilon_\theta(\mathbf{x}_t, t)$ is the predict noise at timestep t using the neural network. The original DDPM is trained to predict the noise ϵ_t instead of x_t and the variance $\Sigma_\theta(\mathbf{x}_t, t)$ is fixed and time-dependent. Since then, improved methods have learned the variance (Nichol and Dhariwal, 2021). We define predicted x_0 as $\hat{x_0} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t)\right)$

Connections to Score Based Generative Models. The score function $s(x)$ of a distribution $p(x)$ is the gradient of the log probability density function i.e, $\nabla_x \log p(x)$. The main premise of score-based generative modeling is to learn the score function of the data distribution given the samples from the same distribution. Once this score function is learned, annealed Langevin dynamics can be used to sample from the distribution using the formula $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \eta\nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\eta}\mathbf{z}_t$, where η is the step size and z_t is sampled from standard normal. The score function can be obtained from the diffusion model using the equation $s_\theta(x_t, t) = -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$ (Weng, 2021).

C. Implications on Recursive Model Training

The internet is increasingly populated by more and more synthetic data (data synthesized from generative models). It is likely that future generative models will be exposed to large volumes of machine-generated data during their training (Martínez et al., 2023a;b). Recursive training on synthetic data leads to mode collapse (Alemohammad et al., 2023; Dohmatob et al., 2024) and exacerbates data biases. In this section, we study the impact of hallucinations within the context of recursive generative model training. We adopt the standard synthetic-only setup similar to (Alemohammad et al., 2023) where we only use synthetic data from the current generative model in training the next generation of generative models. The first generation of generative model is trained on real data and samples from this generative model is used to train the second generation (and so on).

Most past work (Bertrand et al., 2023) studied model collapse to a single mode. This work emphasizes that interaction between modes and mode interpolation plays a massive role when training generative models on their output.

2D Gaussian. When we recursively train a DDPM on its own generated data using a square grid of 2D Gaussians (with $T = 500$), the hallucinated samples significantly influence the learning of the next generation's distribution (see Figure 6). The frequency of the interpolated samples increases as we further train on the learned distribution that consists of interpolated samples. Figure 6d shows samples from Generation 20, where it is evident that the modes have almost collapsed into a single mode, differing greatly from the original data distribution.

Simple Shapes. We define a hallucinated sample as one that contains at least two shapes of the same type (which is never seen in the training distribution). We observe the presence of around 5% hallucinated samples when trained on the real data. We note that the ratio of hallucinated samples increases exponentially as the we iteratively train the diffusion model on its own data. This is expected as the diffusion model progressively learns from a distribution increasingly dominated by hallucinated images, compounding the effect in subsequent generations.

MNIST. We also run the recursive model training on the MNIST dataset (LeCun et al., 1998). At every generation, we generate 65k images and sample 60k images using the filtering mechanism. For each generation, we train a class conditional DDPM with Classifier-Free Guidance (Ho and Salimans, 2022) with $T = 500$ for 50 epochs. To evaluate the quality of the generated images, we compute the FID (Heusel et al., 2017) using a LeNet (LeCun et al., 1998) trained on MNIST instead of Inception backbone as MNIST is not a natural image dataset. In Figure 7, we clearly see that the proposed metric based on the variance of the trajectory outperforms the random filtering method across all generations (lower FID is better). We also plot the Precision and Recall (Shmelkov et al., 2018) curves (in the Appendix Figure 17) where we observe that our filtering mechanism selects high quality samples without much loss in diversity.

Mitigating the curse of recursion with pre-emptive detection of hallucinations. Based on the metric developed in § 3, we analyze the efficacy of the proposed metric in filtering out the hallucinated samples for the next generation of training. After training each generation of the generative model, we sample k images more than size of the training data and then filter out hallucinated samples based on the metric. Figure 7 shows the results on 2D Grid of Gaussians, Simple Shapes and MNIST dataset. We also compare with random filtering where we randomly sample points for the next generation. The variance-based filtering method easily outperforms the random sampling method in all the generations. We see the effectiveness of the proposed metric in minimizing the rate of hallucinations across generations and thus model collapse to a certain extent. This holds true for all the three datasets we have studied in this work.

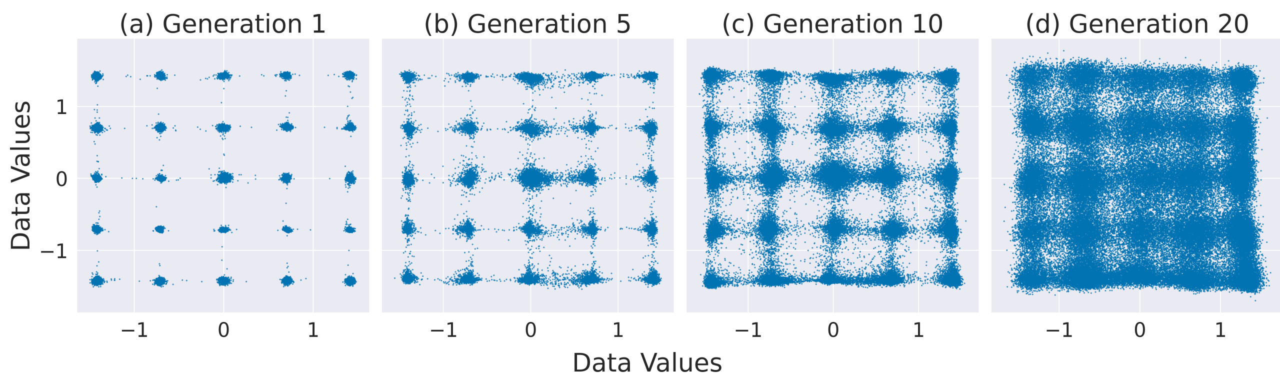

Figure 6: **Recursive Training on 2D GAUSSIAN**. We investigate the impact of recursively training a DDPM on its own generated data using a square grid of 2D Gaussians with $T = 500$ diffusion steps. In each generation, we sample 100k examples, and train the subsequent generation on these data points. As the training progresses through multiple generations, the hallucinated (interpolated) samples significantly influence the learning of the next generation's distribution.

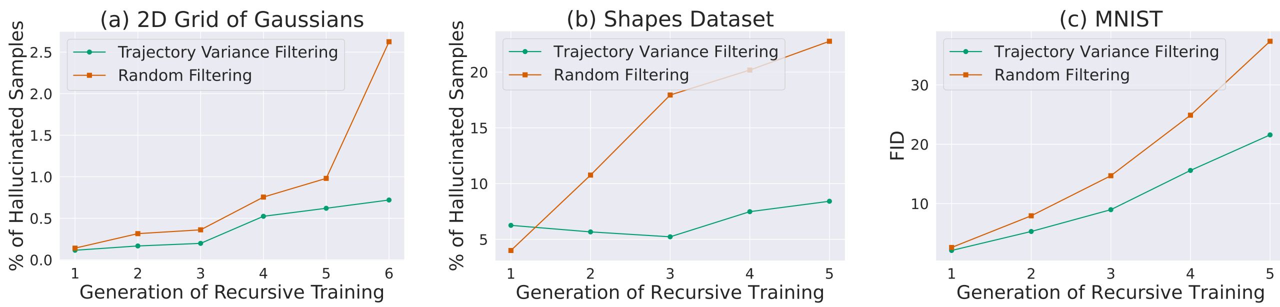

Figure 7: **Mitigating Hallucinations with Pre-emptive Detection**. We filter out hallucinated samples using the metric from § 3 before training on samples from the previous generation of the diffusion model. In the case of (a) 2D GAUSSIAN, (b) SIMPLE SHAPES, where we have clear definitions of hallucination (mode interpolation, and new shape combinations) we see the effectiveness of our variance-based filtering method in minimizing hallucinations across generations compared to random filtering. In the case of (c) MNIST dataset, we measure the FID of subsequent generations and notice that pre-emptive filtering of hallucinated samples makes the recursive model collapse slower.

D. 2D GAUSSIAN

D.1. 2D GAUSSIAN Grid

The reduction in density of mode interpolation as two modes with $\mu = [2, 3]$ are moved apart calls for closer inspection into when and how diffusion models choose to interpolate between nearby modes. To investigate this, we make a toy dataset with a mixture of 25 Gaussians arranged in a two-dimensional square grid. A total of 100,000 samples are present in the training set. Similar to the 1D case, we observe interpolated samples between the two nearest modes of the Gaussian. Again, these samples have close to zero probability if sampled from the original distribution (Figure 8).

We note that mode interpolation only happens between the nearest neighbors. To demonstrate this occurrence, we also train a DDPM on the rotated version of the dataset where the modes are arranged in the shape of a diamond (Figure 8.c,d). The mode interpolation can be more clearly observed in this setting. Interestingly, there appears to be no interpolation between modes along the x-axis, indicating that only the nearest modes are being interpolated. We believe this empirical observation of mode interpolation being confined to nearby modes will spark further investigation.

D.2. Detection

2D GAUSSIAN. Finally, we conclude our investigation on synthetic datasets with experiments on the 2D GAUSSIAN dataset. Similar to the 1D GAUSSIAN setup, we once again measure the prediction variance of the last 20 steps of the reverse diffusion process. We compute the variance per dimension and then take the mean across dimensions to .

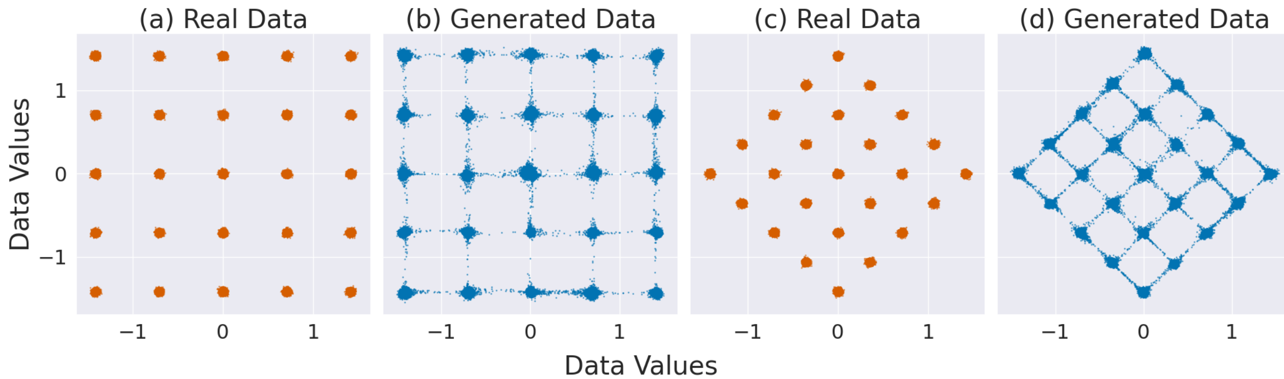

Figure 8: **Mode Interpolation in 2D GAUSSIAN**. The dataset consists of a mixture of 25 Gaussians arranged in a square grid, with a training set containing 100,000 samples. **(a,b)** The blue points represent samples generated by a DDPM, with visible density between the nearest modes of the original Gaussian mixture (in orange). These interpolated samples have near-zero probability in the original distribution. **(c,d)** We trained a DDPM on a rotated version of the dataset where the modes form a diamond shape. In this configuration, we see no interpolation along the x-axis, illustrating that diffusion models interpolate between nearest modes.

With this metric, we can filter out 96% of the hallucinated samples while retaining 95% of the in-support samples.

E. Additional Experimental Details

E.1. Gaussian experiments

We run all our experiments for $10,000$ epochs with batch size of $10,000$. A linear noise schedule is used with starting noise $\beta_0 = 0.001$ and the final noise $\beta_1 = 0.2$. We use $T = 1000$ by default in our experiments (unless specified otherwise). The neural network (NN) is trained to predict the noise (similar to the original DDPM (Ho et al., 2020) implementation) and we use a Mean Squared Error loss to train the model. The input and output of the NN have the same shape (in this case, 1 for 1D Gaussian and 2 for the 2D Gaussian). The NN architecture starts with an initial fully connected layer, followed by three blocks and then output fully connected layer. Each block includes normalization, a LeakyReLU activation, and two fully connected layers. Finally, the output is normalized and transformed back to the input dimension with a fully connected layer. Adam (Kingma and Ba, 2014) with learning rate of 0.001 is used as the optimizer. We build our codebase on top of [1] for the synthetic toy experiments.

Metric: We use $t = 0$ to $t = 15$ (last 15 steps in the reverse diffusion process) to compute the variance of the trajectory in the case of Gaussian 1D and $t = 0$ to $t = 8$ in the case of 2D Gaussian Grid.

[1] https://github.com/tqch/ddpm-torch

E.2. Shapes

The generated images are grayscale images of size 64×64. A total of 5000 images is generated for training the diffusion model. We use a U-Net (Ronneberger et al., 2015) architecture to model the reverse diffusion process. We use a cosine noise scheduler similar to ADM (Nichol and Dhariwal, 2021). We derive our implementation based on [2] for training the DDPM. We train an unconditonal DDPM on the dataset with $T = 1000$ while training and 250 steps during sampling to reduce computation cost (Song et al., 2021a).

E.3. MNIST

MNIST (LeCun et al., 1998) consists of 60,000 grayscale images of size (28, 28). We use classifier-free guidance (Ho and Salimans, 2022) to train a conditional DDPM on MNIST with $T = 500$. For each generation, we train for a total of 50 epochs with a batch size of 512 shared across 4 GPUs. Adam (Kingma and Ba, 2014) optimizer with learning rate of 1e-4 is used to train the network. We use a U-Net (Ronneberger et al., 2015) with 256 feature dimension to model the reverse diffusion process. For the variance filtering mechanism in Section C, we use 10 timesteps between $t = 100$ to $t = 150$ to compute the variance of the trajectory. In the case of MNIST, we do post-hoc filtering just using the samples. This means that we add t timesteps of noise, then compute $\hat{x_0}$ and then use this to compute variance.

Our implementations of the DDPM model is based on PyTorch (Paszke et al., 2019).

[2] https://github.com/VSehwag/minimal-diffusion

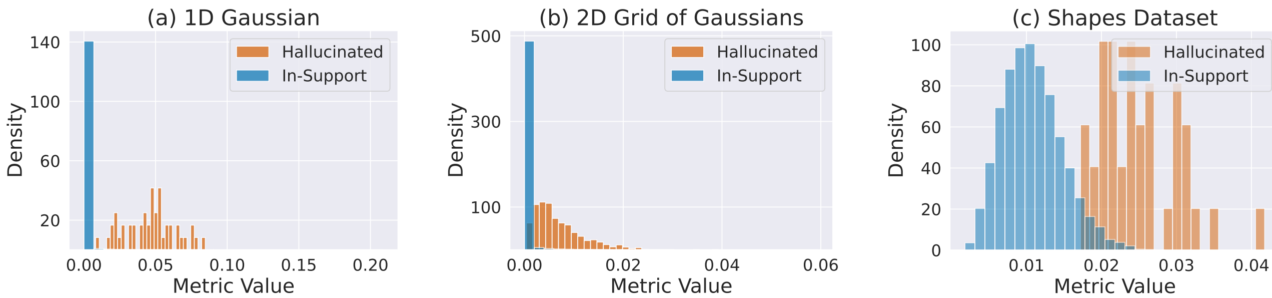

Figure 9: **Histogram of Hallucination Metric**. We depict the hallucination metric values for (a) 1D GAUSSIAN, (b) 2D GAUSSIAN, and (c) SIMPLE SHAPES setups. The histograms show that trajectory variance can capture a separation between hallucinated (orange) and non-hallucinated (blue) samples.

Compute: We run all our experiments of Nvidia RTX 2080 Ti and Nvidia A6000 GPUs. The training and sampling for the Gaussian experiments takes less than 3 hours on single 2080Ti GPU. Sampling 100 million datapoints takes around 3-4 hours. Running DDPM on the shapes dataset takes around 6-7 hours with 4 2080Ti GPUs. The recursive generative training on MNIST takes about 16 hours with 4 A6000 GPUs for 5 generations.

Code: We provide code to run all experiments in the supplementary material.

F. Limitations and Broader Impact

Most of our evidence for mode interpolation comes from the 1D/2D synthetic toy setups. We do not clearly demonstrate what mode interpolation and hallucination look like in real-world natural images. This is a challenging problem because natural images have a high-dimensional, complex distribution.

Hallucinations in LLMs have been studied extensively (Zhang et al., 2023; Ye et al., 2023) given the widespread use of these systems in various contexts. This work investigates hallucinations in diffusion models. In current generative models, these hallucinations could be used to more easily identify machine-generated images. Developing a metric to identify these hallucinations and remove them could make the detection of generated images much harder. However, we argue that understanding hallucinations in diffusion models is crucial as it can help shed light on their failure modes and thereby enable better control in practical applications.

G. Additional Experiments and Figures

The frequency of mode interpolation is inversely proportional to the number of training samples. We train the unconditional diffusion model with 25k, 50k, 100k and 500k samples from the true distribution.

1. Figure 10 shows the histogram of samples generated by the diffusion model (with 10 million samples) when the model is trained on the distribution with $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3$.

2. Figure 11 shows the histogram of samples generated by the diffusion model (with 10 million samples) when the model is trained on the distribution with $\mu_1 = 1, \mu_2 = 2, \mu_3 = 4$.

3. We also experiment with mixture of 2 Gaussians in Figure 12 and 4 Gaussians in Figure 13.

4. Figure 15 shows the FID, precision and recall curves for MNIST across generations.

5. Figure 16 shows additional examples of hallucinated images generated by the diffusion model.

6. Figure 17 shows the $\hat{x_0}$ across various timesteps for a hallucinated image. The number on top of the image indicates the timestep.

7. Figure 18 shows the $\hat{x_0}$ across various timesteps for a image in-support of the distribution. The number on top of the image indicates the timestep.

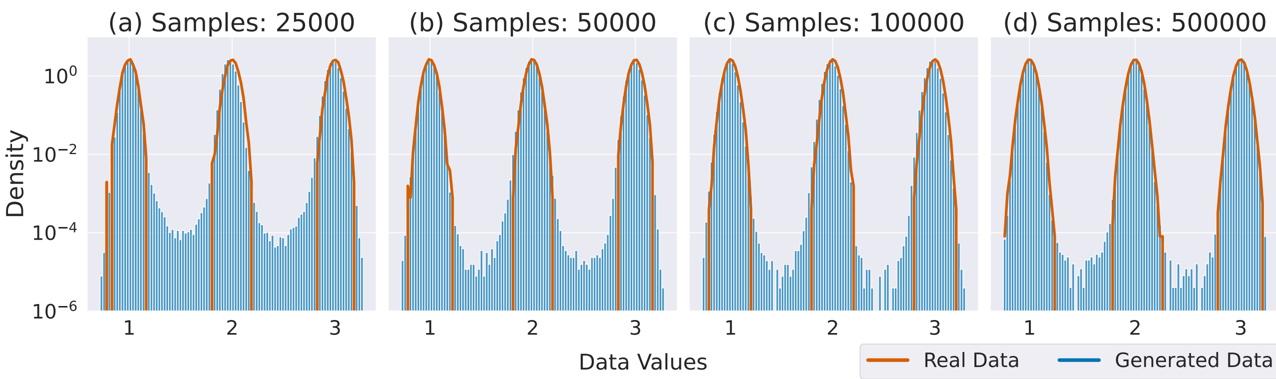

Figure 10: Mixture of 3 Gaussians with $\mu = [1, 2, 3]$. We vary the number of training samples and observe that mode interpolation decreases with increase in the size of training data

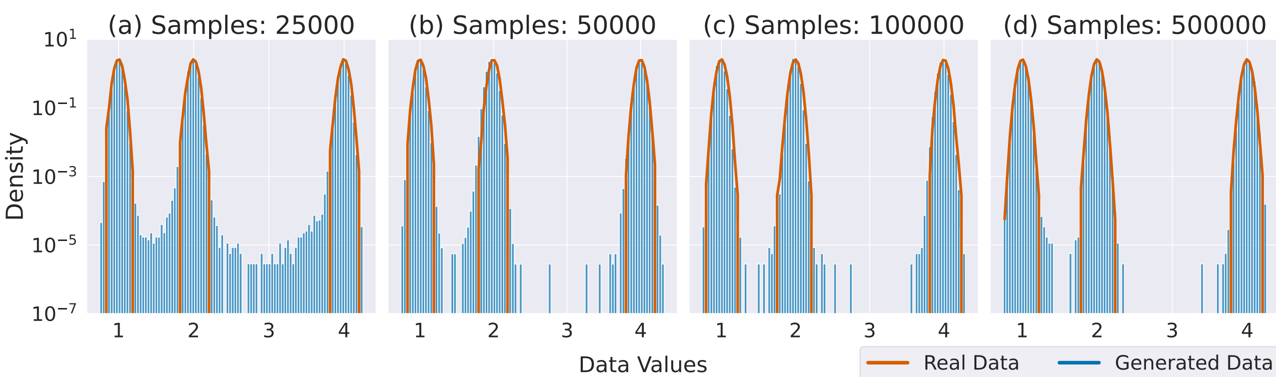

Figure 11: Mixture of 3 Gaussians with $\mu = [1, 2, 4]$. We vary the number of training samples and observe that mode interpolation decreases with increase in the size of training data.

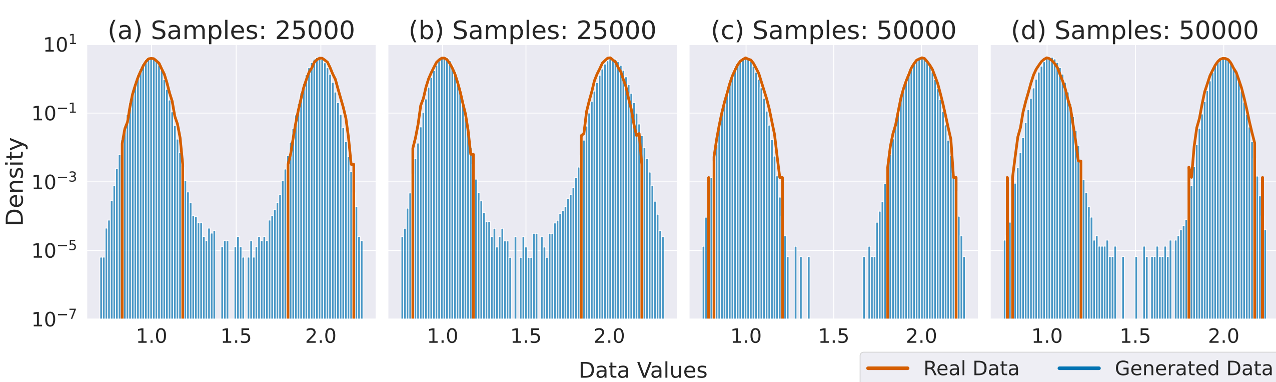

Figure 12: Mixture of 2 1D Gaussians with varying number of training samples. (a) and (b) have the same number of training samples but with two different seeds. Similarly for (c) and (d).

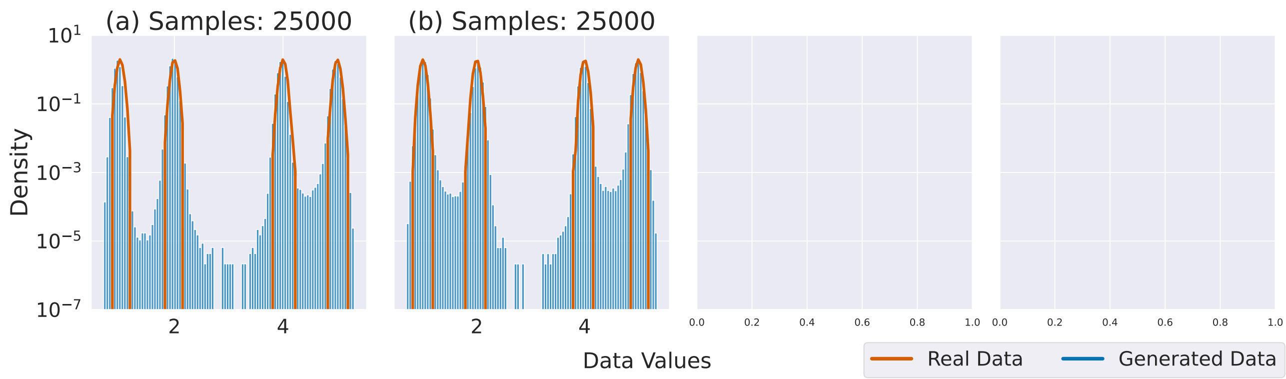

Figure 13: Mixture of 4 1D Gaussians ($\mu = [1, 2, 4, 5]$) with varying number of training samples. (a) and (b) have the same number of training samples but with two different seeds. We clearly see more samples in the region between modes $\mu_1 = 1$ and $\mu_2 = 2$ compared to $\mu_2 = 2$ and $\mu_3 = 4$.

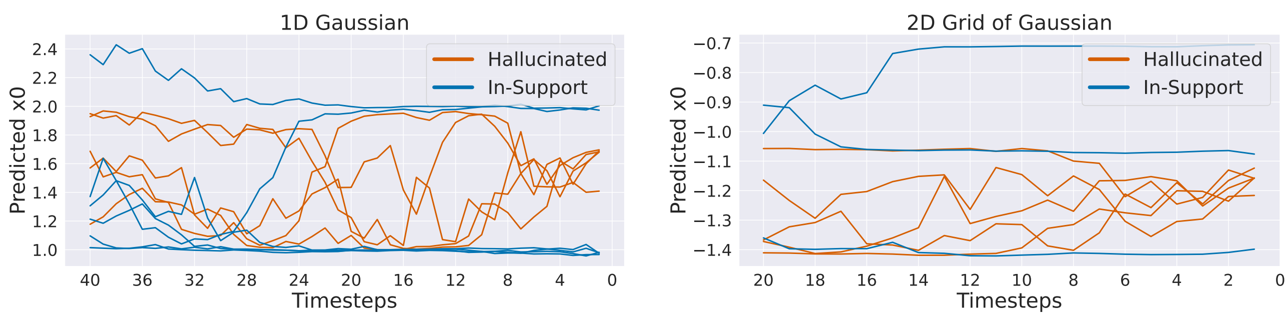

Figure 14: **Variance of $\hat{x_0}$ Trajectories**. The trajectory of the predicted $\hat{x_0}$ for hallucinated (shades of red), and non-hallucinated samples (shades of blue). We see that non-hallucinated samples stabilize in their prediction in the last 20 time steps for both 1D GAUSSIAN and 2D GAUSSIAN setups, whereas the hallucinated samples have high variance in the predicted $\hat{x_0}$ across time steps.

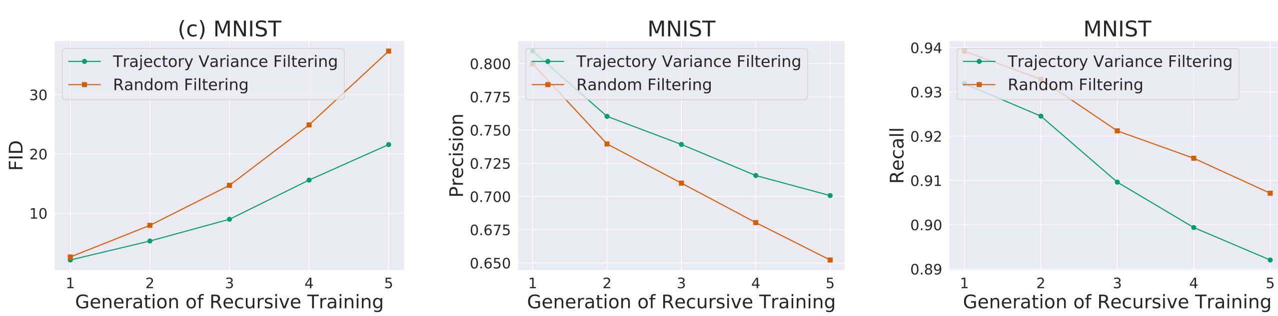

Figure 15: Recursive Generative Training on MNIST with Variance and Random Filtering. We observe that the proposed filtering mechanism can discard low quality samples while maintaining sufficient diversity.

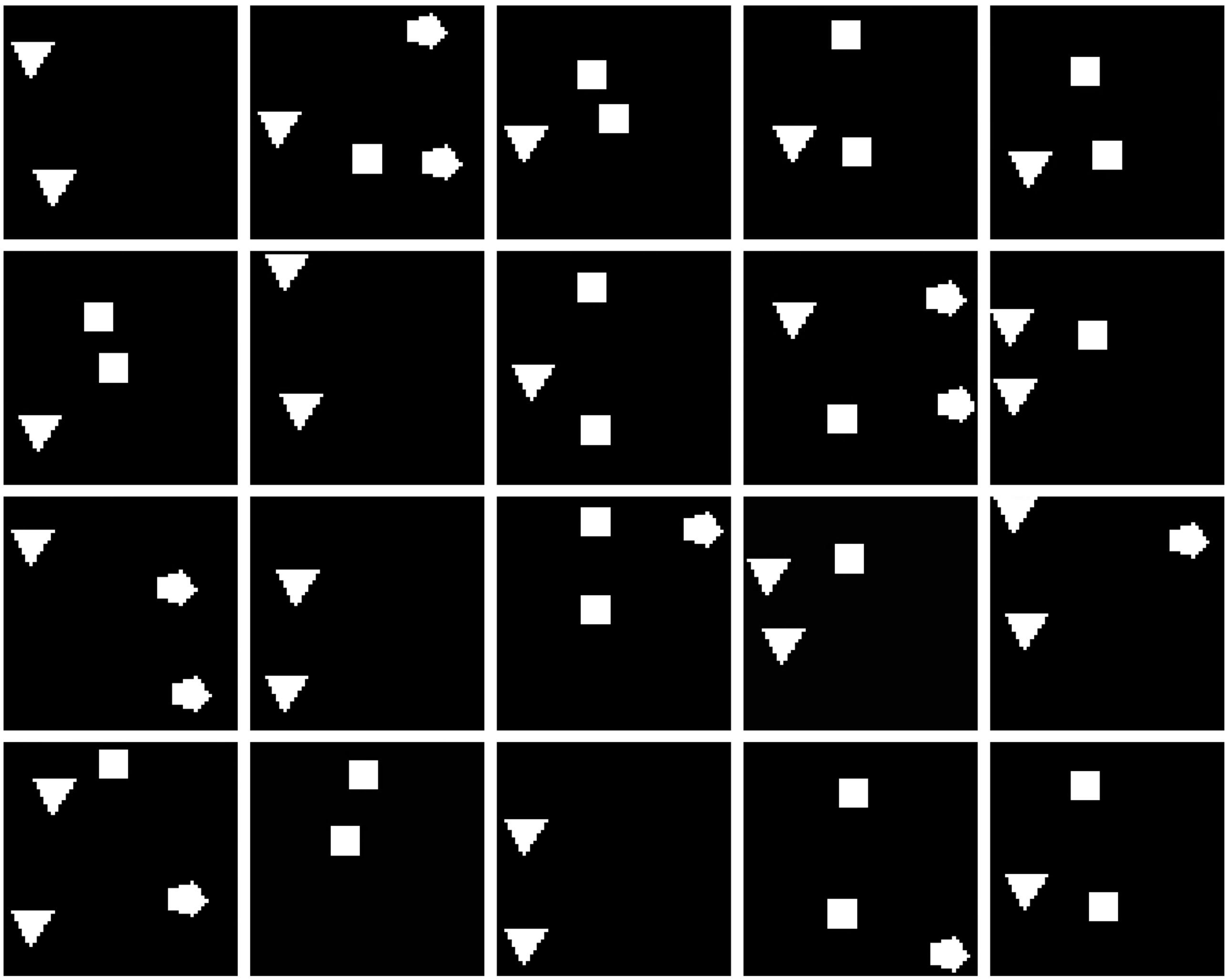

Figure 16: Example of Generated Hallucinated Images

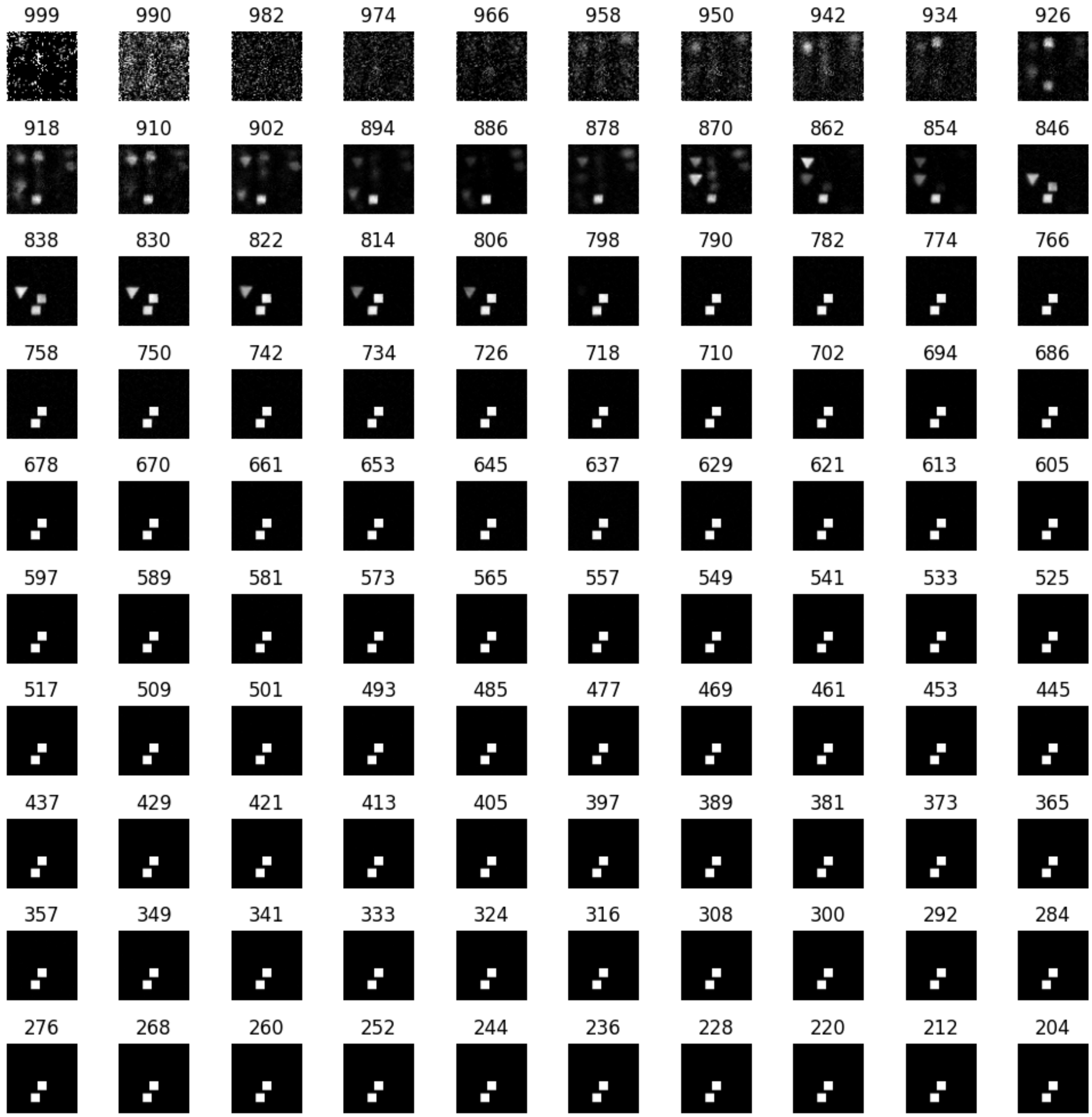

Figure 17: $\hat{x_0}$ for Hallucinated Sample. Here, we observe that the predicted x_0 has a lot of variance around $t = 700$ to $t = 850$. This clearly motivates our proposed metric.

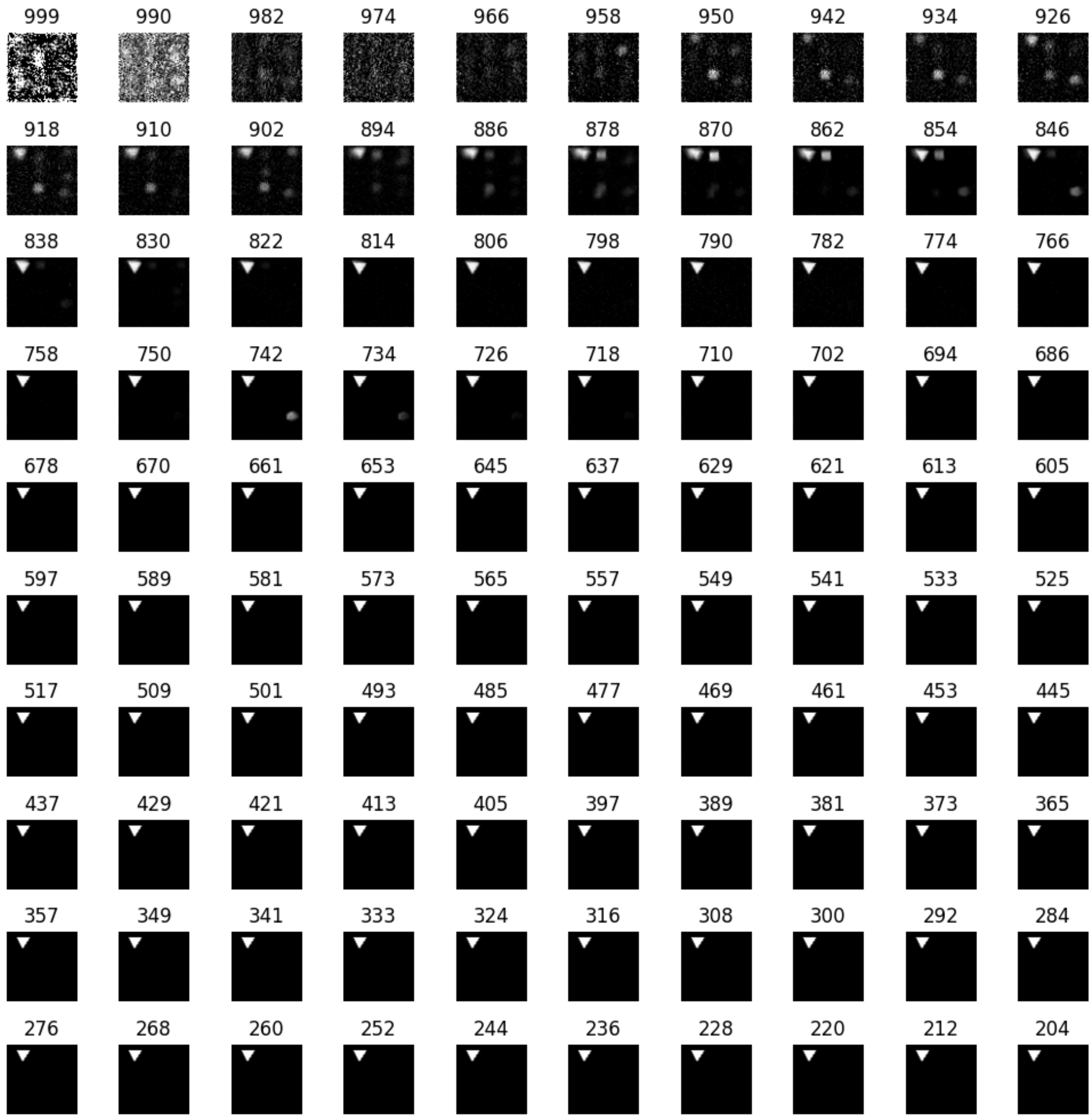

Figure 18: $\hat{x_0}$ for In-Support Sample. Here, we observe that the predicted x_0 is more consistent around $t = 700$ to $t = 850$.