TOWARDS UNDERSTANDING TEXT HALLUCINATION OF DIFFUSION MODELS VIA LOCAL GENERATION BIAS

Anonymous authors
Paper under double-blind review

ABSTRACT

Score-based diffusion models have achieved incredible performance in generating realistic images, audio, and video data. While these models produce high-quality samples with impressive details, they often introduce unrealistic artifacts, such as distorted fingers or hallucinated texts with no meaning. This paper focuses on textual hallucinations, where diffusion models correctly generate individual symbols but assemble them in a nonsensical manner. Through experimental probing, we consistently observe that such phenomenon is attributed it to the network's local generation bias. Denoising networks tend to produce outputs that rely heavily on highly correlated local regions, particularly when different dimensions of the data distribution are nearly pairwise independent. This behavior leads to a generation process that decomposes the global distribution into separate, independent distributions for each symbol, ultimately failing to capture the global structure, including underlying grammar. Intriguingly, this bias persists across various denoising network architectures including MLP and transformers which have the structure to model global dependency. These findings also provide insights into understanding other types of hallucinations, extending beyond text, as a result of implicit biases in the denoising models. Additionally, we theoretically analyze the training dynamics for a specific case involving a two-layer MLP learning parity points on a hypercube, offering an explanation of its underlying mechanism.

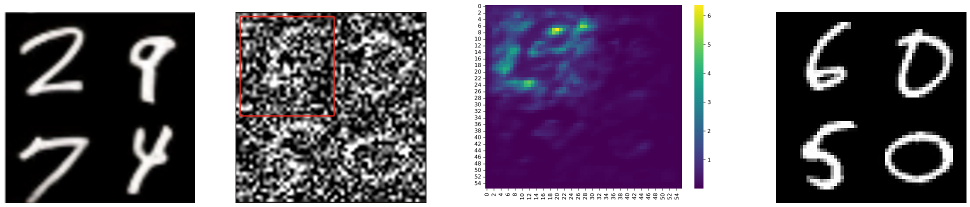
1 INTRODUCTION

Inspired by the diffusion process in physics (Sohl-Dickstein et al., 2015), diffusion models learn to generate samples from a specific data distribution by fitting its score function, gradually transforming pure Gaussian noise into desired samples. These models (Song et al., 2020a; Song & Ermon, 2019; Song et al., 2021; Ho et al., 2020) demonstrate remarkable capability in generating high-quality samples with significant diversity, establishing them as the *de facto* standard generative models for various tasks, including image generation, video generation (Brooks et al., 2024), inpainting (Lugmayr et al., 2022), super-resolution (Gao et al., 2023), and more. However, despite the impressively realistic details produced, diffusion models consistently exhibit artifacts in their outputs. One common issue is the generation of plausible low-level features or local details while failing to accurately model complex 3D objects or the underlying semantics (Borji, 2023; Liu et al., 2023). This phenomenon, known as hallucination, occurs when the generated samples either do not exist in real-world distributions or contain content that lacks semantic meaning. In practice, even large generative models like StableDiffusion (Rombach et al., 2022), trained on enormous datasets, still suffer from these issues—often generating hands with extra, missing, or distorted fingers.

In this work, we primarily focus on a special type of artifacts called text hallucinations, where generative model can correctly generate individual symbol in syllabus but assemble them in nonsensical manner. This naturally raises the following question.

Why do diffusion models typically struggle with generating images that include text content? How do they learn these distributions and end up generating hallucinated samples?

In this work, we take initial steps towards understanding these problems. We find text-form hallucination is closely related to an implicit bias of score networks when trained with score matching objective which we term as ***Local Generation Bias***. This means that the denoising network tends to produce outputs based primarily on local regions of the input. Consequently, the denoising and generation



(a) (b) (c) (d)

Figure 1: An illustration for local generation bias. We construct a synthetic dataset (a) that all images satisfy the rule that sum of first row equals second row, i.e. 2+9=7+4. Diffusion model starts from noise x_t (b) and using denoising network to generate digit images in four quarters. We found that the top-left region's denoising primarily depends on its own data, depicted by saliency map (c). This means the diffusion model independently generates each digit without caring any other digits, ends up with x_0 (d) failing to capture the relation between four digits.

process for each local part operates independently on its own data, end up with a list of symbols uncorrelated. Using synthetic data, we investigate the mechanisms behind this bias, which appears across various denoising architectures and distributions. We observe this phenomenon using saliency map, where gradients are computed with respect to a certain symbol's pixel region to examine its dependence on the input. To measure the degree of local operation, we propose a probe called the local Dependency Ratio (LDR), which quantifies the gradient magnitude within the same local region compared with the entire input. A higher LDR indicates a stronger local generation bias. Interestingly, we discover that a high LDR emerges early in training and persists throughout extensive training steps. This implies that the diffusion model generates text samples in a simplified manner, factorizing the entire distribution into a product of marginal distributions for individual tokens and sampling each token independently, neglecting the subtle connections between tokens and the underlying rules. LDR thus becomes a good indicator for the strength of local generation bias.

One might suspect that this bias originates from the model's architectural design—for instance, basic operations like convolution introduce such inductive biases and cause hallucinations. However, our further experiments and theoretical analysis refute this hypothesis, demonstrating its intricacy. Empirically, even when the network architecture is designed to have a global receptive field to model long-range dependencies, such as transformer (Peebles & Xie, 2023; Vaswani, 2017) and MLP, the same phenomenon persists. The model continues to build its output by relying solely on local information. This suggests that the local generation bias arises from the fundamental training dynamics of score matching for certain distributions, instead of its architecture.

To gain a deeper understanding of this phenomenon, we probe into a simple case, providing insights into its underlying mechanism. Specifically, we analyze a two-layer ReLU network learning a distribution supported on the vertices of a hypercube $\{\pm 1\}^d$. This distribution can be among the vertices that satisfy a parity constraint, where the product of all $\boldsymbol{x}$ entries is 1. When fitting the target denoising function for this distribution, we find that the network has certain training bias, inducing it to separately learn d univariate target function for marginal distributions on each dimension, sampling independently over $\{\pm 1\}$ for each entry. Eventually, the generation process samples uniformly over the entire hypercube rather than parity subset, where hallucination happens. This introduces an instance for how training bias may result in hallucinatory generations, and offers insights into hallucinations across other domains and modalities.

In summary, this paper contributes in three key folds

- **Identification of Local Generation Bias**: We define and analyze the phenomenon of local generation bias in diffusion models, which leads to artifacts like text hallucinations.

- **Mechanistic Explanation:** We provide a detailed theoretical and empirical investigation into the causes of hallucinations, revealing that they stem from fundamental training dynamics rather than architectural limitations.

- **New Analytical Tools:** We introduce the Local Dependency Ratio (LDR) as a measure of local bias and apply it to explore the diffusion model's behavior across training stages.

Figure 2: Some examples of deformed hands artifacts and text hallucination in images generated by StableDiffusion Rombach et al. (2022) and Midjourney. Images from prompting “woman showing her hands”, “a road sign in a grassland” and “a Chinese traditional calligraphy art”.

2 Related Work

Diffusion Model. Diffusion models, initially introduced by Sohl-Dickstein et al. (2015), are probabilistic generative models that iteratively add and remove noise from data. Early work Ho et al. (2020) laid the foundation and proposed Denoising Diffusion Probabilistic Models (DDPM) Ho et al. (2020), which significantly improved sample quality and stability. Song et al. (2020b) also proposed Score-Based Generative Models (SGMs), unifying diffusion models with other generative frameworks. To address efficiency, Song et al. (2021) introduced Denoising Diffusion Implicit Models (DDIM), reducing sampling steps without quality loss. Diffusion models have since been applied beyond image generation, including video generation Brooks et al. (2024), text-to-image models Rombach et al. (2022), and audio synthesis Kong et al. (2020). Despite advancements, challenges remain, particularly in improving sampling speed and generalization to unseen data, as highlighted by recent experiments in video generation and physics-informed modeling.

Hallucination in Language Generative Models. Hallucinations in large language models (LLMs) are a significant challenge, particularly in safety-critical applications, where factually incorrect or logically inconsistent outputs can have severe consequences Ye et al. (2023); Zhang et al. (2023). LLMs may generate erroneous facts, misinterpret instructions, or introduce entirely new information not present in the input, a phenomenon known as input-conflicting hallucination Zhang et al. (2023). Mitigating these hallucinations has become a focus of research, with strategies such as enhancing models with factual data Gunasekar et al. (2023) and integrating retrieval-based mechanisms to ground responses in external knowledge Ram et al. (2023).

Hallucination in Diffusion Models. One common artifact of diffusion models is the generation of distorted or deformed body parts, such as hands and legs, which is frequently observed in models like Stable Diffusion Rombach et al. (2022) and Sora Brooks et al. (2024). Additionally, diffusion models struggle with learning rare concepts, particularly those with fewer than 10,000 samples in the training set Samuel et al. (2024). Other common failure modes include models neglecting spatial relationships or confusing attributes, as discussed in prior research Borji (2023); Liu et al. (2023). These issues highlight the limitations of diffusion models when tasked with generating realistic, complex scenes, especially when dealing with rare data or intricate spatial compositions. Recent work Aithal et al. (2024) explains the hallucination of diffusion model via the perspective of mode interpolation, arguing that the improper interpolation between modes yields non-zero density between them, which is the main cause for hallucination.

3 Preliminary

3.1 Basic Notations

Denote set $\{0, 1, 2, \ldots, n-1\}$ as $[n]$. To compute the cardinality of a set S we write $|S|$. For a vector $\boldsymbol{x}$, we use $\boldsymbol{x}^{(i)} = \boldsymbol{x}^\top \boldsymbol{e}_i$ to denote its i_{th} dimension, and we use $\boldsymbol{e}_i$ to denote the unit vector along the i-th dimension. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ means a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes its density at position $\boldsymbol{x}$. Sampling x from distribution $\mathcal{D}$ is denoted as $x \sim \mathcal{D}$. Asymptotic notation follows the common practice where $f = O(g)$ means there exists a constant $C > 0$ and x_0 such that $f(x) < C \cdot g(x)$ for any $x > x_0$. Similarly we write $f = \Omega(g)$ when $f(x) > C \cdot g(x)$ for any $x > x_0$ and $f(x) = \Theta(g(x))$ if $f = \Omega(g)$ and $f = O(g)$. And $*$ stands for convolution operation between two distributions, $f * g(t) = \int_{\Omega} f(\tau)g(t-\tau)\mathrm{d}\tau$. We use $\Delta(S)$ to denote the set of valid probability distributions over a compact set S. We use $\mathrm{sgn}(x) = 1[x > 0] - 1[x < 0]$ to denote the sign function.

4 Experiment Study

In this section, we introduce the experimental setup and results of our study on text hallucination in diffusion models. We first reproduce text hallucination phenomenon across different modalities and text rules in our simple synthetic setting. To understand how it originates, a key probe called ***Local Dependency Ratio (LDR)*** is introduced to quantitatively measure the denoising function's input dependency on local regions. With LDR as a probing tool, we discover the following important observations that reveal the mechanism of hallucination.

- **High LDR value is always observed when hallucination happens.** This indicates that the denoising model predicts noise by each symbol's region itself, therefore conducting denoising and generation iteration respectively with almost zero entanglement between different symbols. Since the starting Gaussian distribution is also isotropic, the entire generation process for different symbols becomes independent, resulting in incorrect assembly and hallucination.
- **Such phenomenon is ubiquitous across different distributions and architectures**, even for those models with global receptive field such as MLP and DiT(Peebles & Xie, 2023). This indicates such bias is related to ubiquitous implicit bias in training dynamics rather than architectural limitation.
- **As training progresses, LDR decreases and the denoising model starts to overfit**. After extensive training, denoising network overfit to training dataset. This requires it to coordinate different symbols to exact replicate training data, resulting in a drop in LDR.

4.1 Formulation of Text Distribution

The paradigm for constructing a synthetic text-like distribution is as follows. We first define a set of discrete symbols $\mathcal{S} = \{s_1, s_2, \dots, s_K\}$ as syllabus. Define symbol index list $\mathcal{I} = (i_1, i_2, \dots, i_L) \subseteq [K]^L$ which represents a list of token symbol $(s_{i_1}, s_{i_2}, \dots, s_{i_L})$. A spelling/grammar rule is a probability distribution P_G deciding the validity of a symbol sequence by its density $P_G(\mathcal{I})$. Such list of symbol tokens are further rendered into ambient space by a function $h : \mathcal{S} \mapsto \mathbb{R}^d$ which maps each symbol to a vector in ambient space like image pixels or a single scalar. The full signal is obtained by concatenating these vectors. Some examples of the signal are images with texts (where the pixels for different letters do not overlap), time series, text sequences, etc. We wish to learn the distribution of the signals for generation purposes. With a little abuse of notation, we use $h(\mathcal{I}) : \mathcal{S}^L \mapsto \mathbb{R}^{d\times L}$ to denote the rendering process for a list of tokens transformed into input space, which means we can first sample a list of token $\mathcal{I} \sim P_G$ then apply the rendering function h. For simplicity, we fix L throughout all the experiment for the same symbolic system.

In this paper, we mainly test two synthetic symbol assembling rules, including ***(i) Parity Parenthesis***, each sample image contains L parenthesis where left symbol "(" and right ")" both have even numbers; ***(ii) Quarter MNIST***, each sample image consists of four MNIST digits in the corners and the sum of first row equals the second. More details are in appendix.

4.2 Text Hallucination Results

After constructing synthetic text distributions $h(P_G)$. The denoising model is trained to fit the score function of these distributions in the ambient space. For embedding vector ambient space, we employ MLP to learn the score function. For image sample, we use modern denoising network including UNet Ronneberger et al. (2015) and DiT (Peebles & Xie, 2023). Note that both MLP and DiT have global receptive field in its function, enabling them to model long range correlations. UNet also embraces attention module in its pipeline.

Parity Parenthesis. Our initial attempt starts with parity rule with parenthesis symbol. We fix $L = 8, 16$ and use image of left and right parenthesis to represent symbol 1 and -1. A UNet model (with attention) is trained on this image distribution and learn to generate samples. The details of model architecture is in the appendix. We are interested in whether diffusion model can find clues

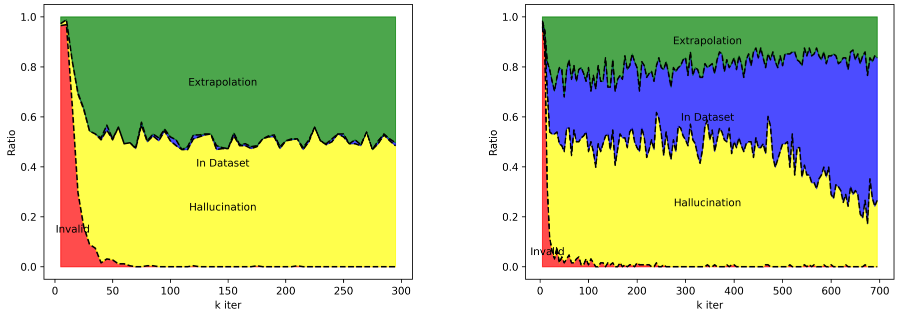


Figure 3: Experimental Result for UNet learning parity parenthesis $L = 16$ (left) and $L = 8$ (right).

about parity rule and faithfully reproduce it. An OCR function is utilized to transform the generated image into binary sequences and test whether it satisfies the parity rule. For $L = 8$ we use half fraction of the valid parity images and 5% for $L = 16$. The generated images are categorized into four types, including ***(i) Invalid**, the low level detail for each symbol is ambiguous and fuzzy hence OCR fails;* ***(ii) Hallucination**, each symbol is clear but the overall combination does not fit in rules;* ***(iii) In Dataset**, the model exactly reproduce dataset images;* ***(iv) Extrapolation**, the model generates data sample that satisfy the rule while not presented in the dataset.*

The diagram for different categories' proportion is in figure 3. Note that random guess has 50% chance of satisfying parity requirement. We can see the model quickly learn to generate individual symbol's appearance, and the proportion of *invalid* drops immediately. However, the diffusion model fails to capture the parity rule, half of whose generated images are hallucination. The situation diverges according to the sequence length. In $L = 8$ case the model eventually successfully overfits to the training dataset, but still generates 25% hallucinated samples. For $L = 16$, The model continues to generate correct samples only by chance till the end of training. This simple experiment demonstrates the difficulty for pure-vision based model to learn underlying rule unconditionally. Detailed generated samples is left in appendix.

Quarter-MNIST: We also test another symbol system, where four MNIST digit images are assigned in four quarters of an image and satisfy simple arithmetic relations. To achieve low divergence between generation distribution and real distribution, the diffusion model not only needs to generate reasonable digits, but also understands the global relations between these digits.

Simple combinatorics tells there are total 670 combination of symbols $(s_1, s_2, s_3, s_4) \in \mathcal{S}^4$ satisfying $s_1 + s_2 = s_3 + s_4$. We randomly leave out 200 combinations as test set and render the images of the rest. Both UNet and DiT undergo a phase that most of its generated samples do not satisfy the addition requirement, which means hallucination. As the training progresses, both models gradually learn to reproduce sample within dataset. DiT performs better accuracy ($\sim 90\%$) in generating samples satisfying addition relations compared with UNet (20.6%). However, *none of them* is able to generate valid symbol tuple beyond the training dataset, with a fraction only less than 0.5%. Therefore there is no extrapolation region in figure 4. In other words, for such text distribution, diffusion model can only struggle between hallucination and overfitting, if no prior knowledge is provided.

4.3 Local Dependency Ratio Analysis

To investigate the mechanism behind text hallucination. We propose a novel probe called **Local Dependency Ratio**, or LDR in abbreviation. LDR quantitatively measures the degree of the diffusion network that performs denoising and generation locally. Given a trained network $\boldsymbol{s_\theta}(\cdot)$ and certain fixed timestep t with corresponding parameter $\bar{\alpha}_t$. In the total input space $\mathbb{R}^{d\times L}$, denote the region of interest as $\mathcal{R} \subseteq [d \times L]$ referring to the set of entries corresponding to one (or few) symbol's area. Define indicator matrix $\boldsymbol{P}_\mathcal{R} = [\boldsymbol{e}_i]_{i\in\mathcal{R}}$ that filters out entries of $\mathcal{R}$. We compute gradient of input $\boldsymbol{x}$ with respect to function $f_{\mathcal{R},\boldsymbol{\theta}}(\boldsymbol{x}) := \boldsymbol{P}_\mathcal{R}^\top \boldsymbol{s_\theta}(\boldsymbol{x})$ and get Jacobian matrix

$$\boldsymbol{J}_{\mathcal{R},\boldsymbol{\theta}}(\boldsymbol{x}) := \frac{\partial f_{\mathcal{R},\boldsymbol{\theta}}}{\partial \boldsymbol{x}} \in \mathbb{R}^{|\mathcal{R}|\times d}. \tag{1}$$

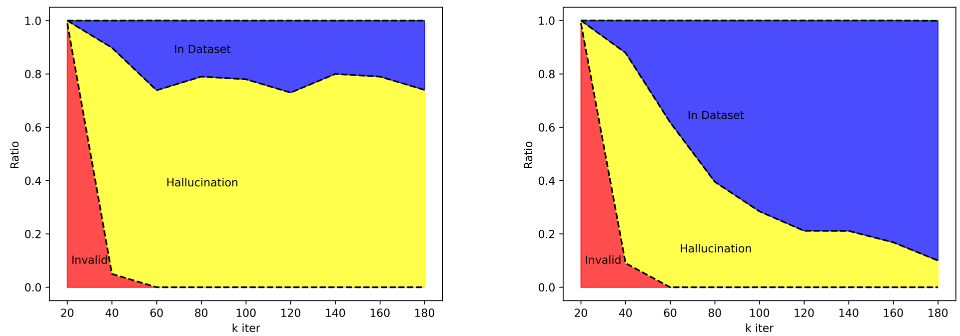


Figure 4: Experimental Result for learning Quarter-MNIST using UNet (left) and DiT (right).

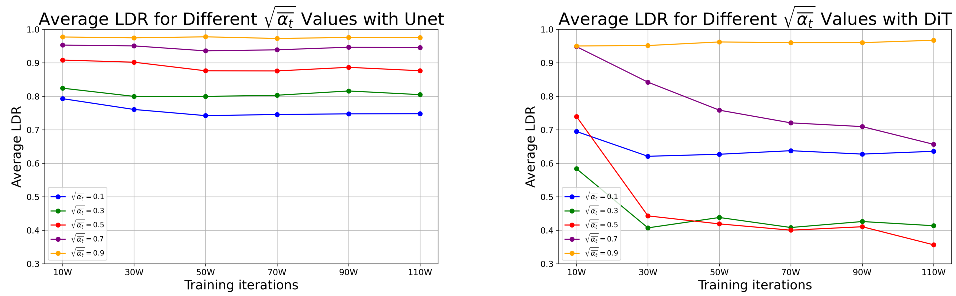


Figure 5: LDR trend for UNet (left) and DiT (right) at different denoising timestep and training iterations. The LDR for UNet remains high throughout the training, therefore it stucks with hallucination. While DiT successfully progress to reduce the LDR, meaning it starts to overfit and memorize the dataset. We select timestep t corresponding to $\sqrt{\bar{\alpha}_t} \approx 0.1, 0.3, 0.5, 0.7, 0.9$.

Define local Dependency Ratio (LDR) function for model $\boldsymbol{s_\theta}$ and region $\mathcal{R}$ as

$$LDR(\boldsymbol{\theta}, \mathcal{R}) = \mathbb{E}_{\boldsymbol{x}\sim p_t}\left[\frac{\mathrm{Tr}(\boldsymbol{P}_{\mathcal{R}}^\top \boldsymbol{J}_{\mathcal{R},\boldsymbol{\theta}}(\boldsymbol{x})^\top \boldsymbol{J}_{\mathcal{R},\boldsymbol{\theta}}(\boldsymbol{x})\boldsymbol{P}_{\mathcal{R}})}{\mathrm{Tr}(\boldsymbol{J}_{\mathcal{R},\boldsymbol{\theta}}(\boldsymbol{x})^\top \boldsymbol{J}_{\mathcal{R},\boldsymbol{\theta}}(\boldsymbol{x}))}\right]. \tag{2}$$

Intuitively, matrix $\boldsymbol{J}$ measures dependency of each output's entry in $\mathcal{R}$ with respect to the input, which is commonly known as *saliency map* (Simonyan, 2013). The difference to conventional saliency map is that each input dimension $\boldsymbol{x}^{(i)}$ receives a gradient vector $\boldsymbol{g}_i \in \mathbb{R}^{|\mathcal{R}|}$ rather a single scalar. It records the sensitivity of output region $\mathcal{R}$ with respect to a certain input entry $\boldsymbol{x}^{(i)}$.

Therefore, $\mathrm{Tr}(\boldsymbol{J}^\top \boldsymbol{J})$ computes the Frobenius norm of $\boldsymbol{J}$, which is the total sum of all gradient vectors' squared norm. Meanwhile, $\boldsymbol{J}\boldsymbol{P}_{\mathcal{R}}$ filters dependency gradient within $\mathcal{R}$ itself, thus $\mathrm{Tr}(\boldsymbol{P}_{\mathcal{R}}^\top \boldsymbol{J}^\top \boldsymbol{J}\boldsymbol{P}_{\mathcal{R}})$ measures the total summation of squared gradient norms within the same local region $\mathcal{R}$. The LDR is thus within range $[0, 1]$, where a higher value indicates a more local denoising and generation manner.

With LDR, we can probe the model trained on different datasets at various checkpoints. Here we mainly present our probing result for Quarter-MNIST dataset. More visualization and other experimental detail is left to appendix. We select $\mathcal{R}$ to be the top left region, namely the first digit's position, and compute LDR for this region at different denoising steps and training iterations. As shown in figure 5, UNet's LDR remains more than 0.75 throughout the entire training process, which means it highly focuses on region $\mathcal{R}$ itself to conduct denoising and generation. This could explain why UNet ends up with a much lower accuracy. DiT also presents similar trend, showing a high LDR value at initial stage of training, therefore generating hallucinated samples. However, due to strong approximation power of transformer architecture, its LDR decreases at 30k to 50k iteration, and this synchronizes with the rapid increase of the generated sample's accuracy (see figure 4).

This result provides evidence for the local generation bias. Despite the capacity to modeling global long-range relations, both attention version UNet and DiT appears to rely on information confined within local regions. This is reasonable because in these symbolic systems P_G, any two symbol's distribution is independent, namely $P_G(s_i = a, s_j = b) = P_G(s_i = a)P_G(s_j = b)$. Such independence leads denoising network to treat symbols as uncorrelated. As a consequence of such local generation preference, the denoising network separately learns and samples from each symbol token's marginal distribution at early training stage, resulting in text hallucination.

This finding is consistent and universal across different denoising architectures and grammar rules. More experiment details for different distributions are left in appendix. We also visualize $\boldsymbol{J}$ as heatmap of each pixel's gradient magnitude and verify its concentration near the selected region $\mathcal{R}$. For real-world distributions that do not satisfy independent condition, please refer to discussion section.

5 Distributions On A HyperCube: A Theoretical Case Study

In this section, we consider a special case of $|\mathcal{S}| = 2$, e.g. the generation of probability distributions on hypercube $\{\pm 1\}^d$. We treat each dimension as a separate token in the distribution, and find that neural networks provably prefers to learn the tokens in their marginal distributions and fails to learn correlations that mark the semantic rules. The training dynamics leads to a sparse feature extraction that let the model primarily builds each dimension of score only depending on same input dimension. Note that our results can be generalized to any sequences that enjoy independent representation space for different tokens, which may not be binary.

5.1 Proof Sketch and Intuition Explanation

The theoretical analysis aims to give a comprehensive explanation under a simple yet non-trivial setting. We admit that it may not directly apply to real-world complicated models, i.e. UNet, DiT, but the point is to provide insight for why training dynamics prefers a local-generation manner. While it is hard to prove things for general models due to technical difficulties, the two-layer network model is widely used in previous theoretical literature for understanding learning phenomenon (Damian et al., 2021; Barak et al.; Lyu et al., 2021).

The theoretical analysis explains why neural networks trained on hypercube data, $\{\pm 1\}^d$, with weakly correlated marginals, tend to focus on local features, resulting in sparse representations and hallucination-prone outputs. When marginals are nearly independent, the gradients primarily reflect single-dimension features, biasing the network towards functions that depend only on individual coordinates $x^{(i)}$. The core proof leverages gradient flow on a two-layer network, $s_\theta(x, t)$, initialized with small weights. It shows that early training amplifies weights $w_{i,j,t}$ aligned with specific dimensions exponentially faster than others, as demonstrated by the growth rate $K_{i,t}$ in Theorem 5.3. Fourier expansion of the target distribution $p_0(x)$ reveals that under proper assumptions 5.1, symmetries in p_0 enforce saddle points in the loss landscape. Theorem 5.2 identifies an invariant set, M, where weights satisfy $a_{i,j,t}(I - e_i e_i^\top) w_{i,j,t} = 0$, forcing $s_\theta^{(i)}$ to depend only on $x^{(i)}$. Training dynamics inherently bias the model towards this set, where it approximates p_0's marginals but fails to capture correlations across dimensions. This result explains why neural networks first prioritize marginal distributions in denoising tasks, requiring prolonged training to escape saddles and learn global dependencies, often leading to hallucination-prone outputs in the early stages.

5.2 Preliminaries

Given a spelling/grammar rule P_G over binary sequences, The marginal distribution p_t of $\boldsymbol{x}_t$ is the convolution of original distribution using a Gaussian noise kernel as $p_t = \sqrt{\bar{\alpha}_t} p_0 * \mathcal{N}(0, (1 - \bar{\alpha}_t)\boldsymbol{I})$. To generate sample from distribution $p_0(x)$, diffusion model learns a denoising function $\boldsymbol{s_\theta}(\boldsymbol{x}_t, t)$ that predicts the mean of the posterior distribution $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$. Given $\boldsymbol{x}_0$, simple Bayesian rule yields its conditional density as

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}\boldsymbol{I}), \tag{3}$$

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\boldsymbol{x}_t, \quad \tilde{\beta}_t := \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t. \tag{4}$$

The unconditional posterior density is the expectation over different starting points $\boldsymbol{x}_0$.

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{x}_0 \sim p_0(x)}[q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)]. \tag{5}$$

Setting. Suppose that we wish to learn a distribution on the hypercube $p_0 \in \Delta(\{\pm 1\}^d)$, or equivalently a binary sequence of length d. For any $t > 0$, the training data is drawn from distribution $x_t \sim p_t$ (**??**) as $x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \xi$, where $x_0 \sim p_0$ and $\xi \sim \mathcal{N}(0, I_d)$. The loss for time t can be decomposed as

$$\begin{aligned}
\mathcal{L}_t(\theta) &= \mathbb{E}_{x_0,\xi} \left\|\xi - s_\theta(x_t,t)\right\|^2 = \mathbb{E}_{x_0,\xi} \left\| \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1-\bar{\alpha}_t}} - s_\theta(x_t,t) \right\|^2 \\
&= \mathbb{E}_{x_0,\xi} \left\| \frac{x_t - \sqrt{\bar{\alpha}_t}\mathbb{E}(x_0|x_t)}{\sqrt{1-\bar{\alpha}_t}} - s_\theta(x_t,t) \right\|^2 + \mathbb{E}_{x_0,\xi} \left\| \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1-\bar{\alpha}_t}} (x_0 - \mathbb{E}(x_0|x_t)) \right\|^2 \\
&= \mathbb{E}_{x_0,\xi} \|y_t(x_t) - s_\theta(x_t,t)\|^2 + C_t
\end{aligned}$$

Therefore the loss can be viewed as the square loss on the target vector $y_t(x) = \frac{x_t - \sqrt{\bar{\alpha}_t}\mathbb{E}(x_0|x_t)}{\sqrt{1-\bar{\alpha}_t}}$. We learn the target via running gradient flow (GF) on a two-layer neural network of hidden dimension m. The i-th entry of the network output is $s_\theta^{(i)}(x,t) = \sum_{j\in[m]} a_{i,j,t}\sigma(w_{i,j,t}^\top x + b_{i,j,t})$ where $\sigma(x) = \max(x, 0)$ is the ReLU function, and the parameters $\theta = (a_{i,j,t}, w_{i,j,t}, b_{i,j,t})$ are updated via GF as $\frac{d}{ds}\theta_s = -\nabla_\theta \mathcal{L}_t(\theta_s)$.

Notice that $\mathcal{L}_t$ is always differentiable and smooth as the data x has a smooth probability density function over the space. For a starting point θ_0, we use $\Phi(\theta_0, s)$ to denote the endpoint θ_s of gradient flow at time s. Given the smoothness of the loss function, there exists a unique solution to the gradient flow given the starting point. We will show that there are invariant sets that gradient flow cannot escape, namely the network may stuck in such invariant set in a way similar to saddle points. While there is no model in the set that recover the true target well, there are models in the set that creates both reasonable and hallucinatory generations. Furthermore, we introduce an instance of training paradigm where the network model is biased towards the invariant set in the early training phase.

5.3 Sparse Weight Is a Saddle Point For Optimization

By Fourier transformation on the hypercube, we can expand the target probability $p_0(x) = \sum_{I\subset[d]} \bar{p}_0(I)x_I$ where $x_I = \prod_{i\in I} x_i$ are the Fourier basis. In the experiments, the observations are that some symmetries of the data may blind the neural network from hidden patterns of the target probability p_0. We formalize some sufficient conditions of data symmetries as follows.

Assumption 5.1. For any $i, j \in [d]$, $\bar{p}_0(i) = 0$ and $\bar{p}_0(i,j) = 0$.

This means that the marginal distribution for any digit in the sequence is uniform, and for any pair of digits is independent. The examples of such distributions are all valid sequence vectors uniformly drawn from a parity rule (e.g. satisfies $\prod_{i\in I} x_i = 1$ for any I that $|I| > 2$).

We observe that such p_0 will induce a diffused distribution p_t and learning target $y_t(x)$ for any $t > 0$. Thought the target may depend on all the coordinates of its input x, we observe in the experiments that the neural network often learns a function that has much lower variance for the off-diagonal entries compared with the true target, and it often takes a much longer training process for the network to recover from such low variance. This hints a potential saddle point that the network may stuck into during training. Please refer to appendix for more observed evidence if interested.

Theorem 5.2. *Under Assumption 5.1, let $M = \{\theta : a_{i,j,t}(I - e_i e_i^\top)w_{i,j,t} = 0\}$, then M is an invariant set under gradient flow. Namely, from any $\theta \in M$, gradient flow $\Phi(\theta, t) \in M$, $\forall t > 0$.*

We observe that for any $\theta \in M$, $s_\theta^{(i)}(x,t) = \sum_{j\in[m]} a_{i,j,t}\sigma(w_{i,j,t}^\top x + b_{i,j,t})$ is irrelevant to the dimensions of x other than $x^{(i)}$, therefore it cannot represent the true target for any distribution that involves correlations over multiple dimensions. Actually since the denoising process starts from a Gaussian distribution that enjoys independent entries for different dimensions, the denoised sample will also have independent entries. For instance, for the generation of the parity rule, the best model in M can only generate the uniform distribution over all of the hypercube, giving a half chance of hallucination. Therefore it is favorable for optimization to avoid the model stucking in the saddle set of M. However, we will show that having $w_{i,j,t}$ sparse towards e_i is implicitly induced by running gradient flow with small initialization.

5.4 Training Is Biased Towards Sparse weight In Early Phase

A series of previous works (Woodworth et al., 2020; Jin et al., 2023) adopts small initialization to assist representation learning for MLP networks, as opposed to the kernel regime where the network representations barely changes. We adopt the idea to check how network's parameters change in the early phase of training, and finding provably biased towards sparse feature extraction. which means the second layer only receives information from the same corresponding dimension in input.

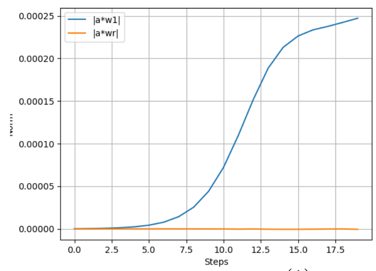


Figure 6: When learning $s_\theta^{(1)}(x_t,t)$, the average norm of neurons' 1_{st} dimension weight $|aw^{(1)}|$ increase much faster than $\|a(w-w^{(1)})\|$. It means $s_\theta^{(1)}(x_t,t)$ becomes a univariate function for $x_t^{(1)}$.

For a small constant σ_{init}, we use the initialization scheme for $\theta = (a_{i,j,t}, w_{i,j,t}, b_{i,j,t})$ as

$$w_{i,j,t}(0) \sim \mathcal{N}(0,\sigma_{init}^2), \quad b_{i,j,t}(0) \sim \mathcal{N}(0,\sigma_{init}^2 r^2), \quad a_{i,j,t}(0) \sim \text{Unif}(\{\pm 1\})\sqrt{\|w_{i,j,t}(0)\|^2 + b_{i,j,t}(0)^2}.$$

Inspired by the G-function (Maennel et al., 2018; Lyu et al., 2021), when the initialization is very small, the neural network output $s_\theta(x,t) \approx 0$, so we can expand the loss as

$$\mathcal{L}_t(\theta) = \mathbb{E}_{x_t}\|s_\theta(x_t,t) - y_t(x_t)\|^2 = \mathbb{E}_{x_t}[\|y_t(x_t)\|^2 - 2s_\theta(x_t,t)^\top y_t(x_t)] + O(\mathbb{E}_{x_t}(s_\theta^2(x_t)))$$

So the initial trajectory of the neural network aims to optimize a surrogate loss $\tilde{\mathcal{L}}(\theta) = \mathbb{E}_{x_t}[-2s_\theta(x_t,t)^\top y_t(x_t)]$. Hereinafter we consider a trajectory $(\tilde{a}_{i,j,t}, \tilde{w}_{i,j,t}, \tilde{b}_{i,j,t})$ on such a surrogate loss.

Theorem 5.3. *Under Assumption 5.1, the weight of each neuron $|a_{i,j,t}|$ grows exponentially in time: for every i,t, there exists a function $K_{i,t}: S^d \to \mathbb{R}$ such that*

$$|\tilde{a}_{i,j,t}(s)| = |\tilde{a}_{i,j,t}(0)| \exp\left(2\text{sgn}(\tilde{a}_{i,j,t}(0)) \int_0^s K_{i,t}\left(\frac{\tilde{w}_{i,j,t}(\tau)}{|\tilde{a}_{i,j,t}|(\tau)}, \frac{\tilde{b}_{i,j,t}(\tau)}{|\tilde{a}_{i,j,t}|(\tau)}\right) d\tau\right)$$

The function $K_{i,t}$ marks the growth rate. The rate satisfy

- $0 < K_{i,t}(w,b) < \sqrt{1-\bar{\alpha}_t}$ *when* $w^{(i)} > 0$*;* $0 > K_{i,t}(w,b) > -\sqrt{1-\bar{\alpha}_t}$*, when* $w^{(i)} < 0$.
- *When* $w^{(i)} > 0$*, the maximal value of* $K_{i,t}(w,b)$ *is uniquely achieved at* $(w,b) = \frac{1}{\sqrt{1+(D^*)^2}}(e_i, D^*)$ *for some* $D^* \in \mathbb{R}$*; When* $w^{(i)} < 0$*, the minimal value of* $K_{i,t}(w,b)$ *is uniquely achieved at* $(w,b) = \frac{1}{\sqrt{1+(D^*)^2}}(-e_i, -D^*)$.
- *the maximally-growing neuron directions* $\left(\frac{\tilde{w}_{i,j,t}(\tau)}{|\tilde{a}_{i,j,t}|(\tau)}, \frac{\tilde{b}_{i,j,t}(\tau)}{|\tilde{a}_{i,j,t}|(\tau)}\right) = \frac{1}{\sqrt{1+(D^*)^2}}(e_i, D^*)$ *for* $a_{i,j,t} > 0$ *and* $\left(\frac{\tilde{w}_{i,j,t}(\tau)}{|\tilde{a}_{i,j,t}|(\tau)}, \frac{\tilde{b}_{i,j,t}(\tau)}{|\tilde{a}_{i,j,t}|(\tau)}\right) = \frac{1}{\sqrt{1+(D^*)^2}}(-e_i, -D^*)$ *for* $a_{i,j,t} < 0$ *are invariant under gradient flow.*

The theorem depicts the representation learning process in the early phase of training. For a neuron along the direction $\tilde{w}_{i,j,t}(0) = \|\tilde{w}_{i,j,t}(0)\|e_i$, with proper bias, the growth rate of its norm is maximal and its direction does not alter, and therefore after a fixed period it will have an exponentially larger impact than a neuron of suboptimal direction. Therefore after a few epochs, a neuron $(a_{i,j,t}, w_{i,j,t}, b_{i,j,t})$ either still has a small magnitude ($|a_{i,j,t}|$ and $\|w_{i,j,t}\|$), or its weight will be close to the optimal direction $\tilde{w}_{i,j,t}(s) \simeq \|\tilde{w}_{i,j,t}(s)\|e_i$, making the whole network close to the saddle set M introduced in Theorem 5.2. While it may take a long time for the network to escape the neighborhood of M, the network can operate inside M to learn denoising functions that recovers each marginal distribution of p_0. In this way the model independently conducts denoising on each individual dimension, performing local generation that introduces hallucination.

6 DISCUSSION

Does local generation bias still hold for real world distribution? In our analysis and synthetic text distribution, the condition that different token symbols are independent plays a critical role. Is our discovered bias and mechanism still robust when the distribution does not strictly satisfy this requirement? We conduct experiments on real world text distributions to answer these concerns. We construct two datasets, one rendering 1,000 common English words and the other contains 1,000 common Chinese characters. When using diffusion model to learn score matching and generate samples, we observe similar phenomenon persists to happen. Both models go through the "fuzzy-hallucination-overfitting" three phases, randomly assembling radicals and letters in nonsensical way at intial stage, and overfit to duplicate training data after long period of training. We also test LDR in these scenarios, finding the same decreasing pattern. Note that we selcet $\sqrt{\bar{\alpha}_t} = 0.2$ because signal-noise-ratio increase significant at this stage and it is critical for determining the final content. Result shows the same local generation bias still exists at early stage of training and leads to hallucination for real text distribution. It also confirms that the decrease of LDR implies overfitting.

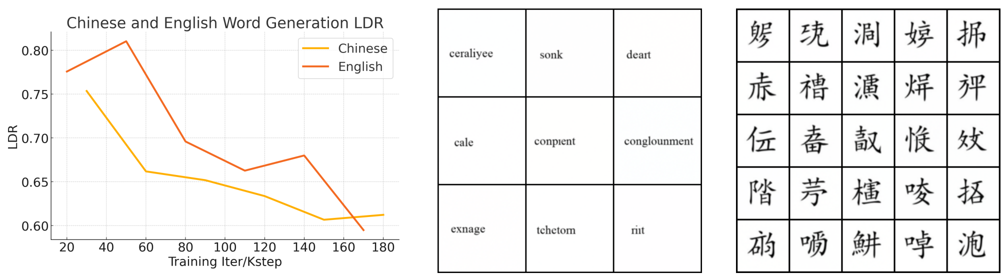


Figure 7: LDR trend for UNet learning Chinese and English texts (left). Hallucination example for misspelling words (middle) and Chinese characters (right).

No-Free Lunch Issue. It is not surprising that a vision generative model with zero prior knowledge fails to learn human-contrived rule for spelling. Because without any assumption on the possible grammar/spelling rules, it is intrinsically ambiguous to determine what is the real law. Therefore, the main point of this work is to investigate the bias of modern deep learning models, i.e. MLP, UNet, DiT, to understand how they tend to generate samples and show its discrepancies to real distributions. Our work thus highlights the importance of introducing explicit modeling of text knowledge, without which diffusion model is inevitable to hallucinate.

7 CONCLUSION

In this paper, we have presented a detailed investigation into the phenomenon of hallucinations in generating text-related contents. By combining empirical observations with theoretical analysis, we have demonstrated it to be closely related to the implicit bias of denoising network called local-generation bias. We find that such bias is not a consequence of the model's architecture but rather an inherent property of the training dynamics driven by score matching. It is shown that diffusion models, despite their global receptive field capabilities, tend to rely on local information during the denoising process, generating symbols independently without capturing the global structure. The key to form this bias is the nearly pairwise independence between marginal distributions for each token symbol. We further introduced the Local Dependency Ratio (LDR) as a novel metric to quantify the extent of this local generation bias and applied it to various diffusion models, showing that this bias emerges early in training and persists through extensive training phases, even in real-world distribution where independent condition does not strictly hold. This study may shed light on the mechanisms behind hallucinations in diffusion models, highlighting the importance of addressing local generation bias for more accurate and coherent generation in tasks requiring complex global structure understanding.

REFERENCES

Sumukh K Aithal, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. *arXiv preprint arXiv:2406.09358*, 2024.

Boaz Barak, Benjamin L Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit, january 2023. *URL http://arxiv. org/abs/2207.08799*.

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

Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise SGD provably prefers flat global minimizers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 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*, pp. 10021–10030, 2023.

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.

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

Jikai Jin, Zhiyuan Li, Kaifeng Lyu, Simon Shaolei Du, and Jason D Lee. Understanding incremental learning of gradient descent: A fine-grained analysis of matrix sensing. In *International Conference on Machine Learning*, pp. 15200–15238. PMLR, 2023.

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

Qihao Liu, Adam Kortylewski, Yutong Bai, Song Bai, and Alan Yuille. Intriguing properties of text-guided diffusion models. *arXiv preprint arXiv:2306.00974*, 2, 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*, pp. 11461–11471, 2022.

Kaifeng Lyu, Zhiyuan Li, Runzhe Wang, and Sanjeev Arora. Gradient descent on two-layer nets: Margin maximization and simplicity bias. *Advances in Neural Information Processing Systems*, 34:12978–12991, 2021.

Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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*, pp. 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, pp. 4695–4703, 2024.

Karen Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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

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

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*, 2020b.

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, 2021.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

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.

A Appendix

A.1 Diffusion Model

We adopt the conventions for diffusion models from Ho et al. (2020). Let $p_0(x)$ be the real data distribution. We define a forward process $\boldsymbol{x}_t$ where the signal gradually shrinks and Gaussian noise is added at each timestep for total T steps.

$$\boldsymbol{x}_0 \sim p_0(x), \quad p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}),\ t = 1, 2, \ldots, T. \tag{6}$$

Here $0 < \beta_t < 1$ is a scale schedule of adding noise. Let $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_T = \prod_{t=1}^{T} \alpha_t$, and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. To sample from the distribution p_0, we train a neural network $\boldsymbol{s}_{\boldsymbol{\theta}}$ as the score model and iteratively remove noise from a random gaussian variable as follows.

$$\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I}), \quad \boldsymbol{x}_{t-1} \sim \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{s}_\theta\left(\boldsymbol{x}_t, t\right)\right), \tilde{\beta}_t \boldsymbol{I}\right) \tag{7}$$

The score model is trained as the following process. For a batch of training data $\{(\boldsymbol{x}_0)_i\}_{i\in[N]}$, we add independent Gaussian noise $\{\boldsymbol{\xi}_i \sim \mathcal{N}(0, \boldsymbol{I})\}_{i\in[N]}$, construct $(\boldsymbol{x}_t)_i = \sqrt{\bar{\alpha}_t}(\boldsymbol{x}_0)_i + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\xi}_i$, and then train a network $\boldsymbol{s_\theta}$ to predict $\boldsymbol{\xi}$ given $\boldsymbol{x}_t$ and t. Namely, we train the score model to minimize the loss function

$$\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{TN}\sum_{t\in[T]}\sum_{i\in[N]}\lambda_t \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\boldsymbol{I})}\left[\left\|\boldsymbol{\xi} - \boldsymbol{s_\theta}(\sqrt{\bar{\alpha}_t}(\boldsymbol{x}_0)_i + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\xi}, t)\right\|^2\right]. \tag{8}$$

The λ_t characterizes loss weight for different timesteps, and by default we set $\lambda_t = 1$.

A.2 Fourier Transformation

For real-valued functions supported on the hypercube $\{\pm 1\}^d$, the indicator functions $x_I = \prod_{i\in I} x_i$ forms an orthonormal basis over the uniform distribution $\mathcal{D}$. That is, $\mathbb{E}_{x\sim\mathcal{D}} x_I x_J = 1[I = J]$. For any function $f : \{\pm 1\}^d \to \mathbb{R}$, we can expand the function in the Fourier basis as

$$f(x) = \sum \bar{f}_I x_I(x), \qquad \bar{f}_I = \mathbb{E}_{x\sim\mathcal{D}} f(x) x_I(x).$$

When f is a probability function, the set $F_S = \{\bar{f}_I : I \subset S\}$ gives the marginal distribution of $(x_i : i \in S)$. For instance, when $\bar{f}_{(i,j)} = \bar{f}_{(i)}\bar{f}_{(j)}$, x_i and x_j are independent in their marginal distribution.

A.3 Proof for the saddle point Theorem 5.2

For any $\theta \in M$, we can set $w_{i,j,t} = c_{i,j,t} e_i$ for any neuron that $a_{i,j,t} \neq 0$, where $c_{i,j,t} \in \mathbb{R}$ are scalars. Then the network outputs $s_\theta^{(i)}(x) = \sum a_{i,j,t}\sigma(c_{i,j,t}x^{(i)} + b_{i,j,t})$ depend only on the i-th coordinate of the input x.

Now we compute the gradient for $w_{i,j,t}$ as

$$\begin{aligned}
\frac{d}{ds} w_{i,j,t} &= -2\mathbb{E}_{x_t}(s_\theta^{(i)}(x_t) - y_i(x_t)) a_{i,j,t}\sigma'(w_{i,j,t}^\top x_t + b_{i,j,t}) x_t \\
&= -2\mathbb{E}_{x_0,\xi} s_\theta^{(i)}(x_t) a_{i,j,t}\sigma'(c_{i,j,t}x_t^{(i)} + b_{i,j,t}) x_t \\
&+ 2\mathbb{E}_{x_0,\xi} a_{i,j,t}\xi^{(i)}\sigma'(\sqrt{\bar{\alpha}_t}c_{i,j,t}x_0^{(i)} + \sqrt{1-\bar{\alpha}_t}c_{i,j,t}\xi^{(i)} + b_{i,j,t}) x_t
\end{aligned}$$

Now we wish to show that $\frac{d}{ds} w_{i,j,t}$ is still along the direction of e_i, thereby gradient flow does not escape the region of M. This is done by the following two lemmas.

Lemma A.1. *For any $k \neq i$,*

$$\mathbb{E}_{x_0,\xi} s_\theta^{(i)}(x_t) a_{i,j,t}\sigma'(c_{i,j,t}x_t^{(i)} + b_{i,j,t}) e_k^\top x_t = 0.$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{x_0,\xi} s_\theta^{(i)}(x_t)\sigma'(c_{i,j,t}x_t^{(i)} + b_{i,j,t}) e_k^\top x_t \\
&= \mathbb{E}_{x_0,\xi}\sum_{j'} a_{i,j',t}\sigma(c_{i,j',t}x_t^{(i)} + b_{i,j',t})\sigma'(c_{i,j,t}x_t^{(i)} + b_{i,j,t}) x_t^{(k)}
\end{aligned}$$

Notice that by Assumption 5.1, $x_0^{(i)}$ and $x_0^{(k)}$ are independent, and for standard Gaussian $\xi^{(i)}$ and $\xi^{(k)}$ are independent. Furthermore $\mathbb{E}x_t^{(k)} = \sqrt{\bar{\alpha}_t}\mathbb{E}x_0^{(k)} + \sqrt{1-\bar{\alpha}_t}\mathbb{E}\xi^{(k)} = 0$. Therefore,

$$\begin{aligned}
&\mathbb{E}_{x_0,\xi} s_\theta^{(i)}(x_t)\sigma'(c_{i,j,t}x_t^{(i)} + b_{i,j,t}) e_k^\top x_t \\
&= \sum_{j'} a_{i,j',t}[\mathbb{E}_{x_t^{(i)}}\sigma(c_{i,j',t}x_t^{(i)} + b_{i,j',t})\sigma'(c_{i,j,t}x_t^{(i)} + b_{i,j,t})][\mathbb{E}_{x_t^{(k)}} x_t^{(k)}] \\
&= 0.
\end{aligned}$$

$\square$

Lemma A.2. *For any $k \neq i$,*

$$\mathbb{E}_{x_0,\xi}\xi^{(i)}\sigma'(\sqrt{\bar{\alpha}_t}c_{i,j,t}x_0^{(i)} + \sqrt{1-\bar{\alpha}_t}c_{i,j,t}\xi^{(i)} + b_{i,j,t})e_k^\top x_t = 0.$$

Proof. Similarly, $x_t^{(k)} = \sqrt{\bar{\alpha}_t}x_0^{(k)} + \sqrt{1-\bar{\alpha}_t}\xi^{(k)}$ has mean 0. Since $(x_0^{(k)}, \xi^{(k)})$ and $(x_0^{(i)}, \xi^{(i)})$ are independent, we know

$$\begin{aligned}&\mathbb{E}_{x_0,\xi}\xi^{(i)}\sigma'(\sqrt{\bar{\alpha}_t}c_{i,j,t}x_0^{(i)} + \sqrt{1-\bar{\alpha}_t}c_{i,j,t}\xi^{(i)} + b_{i,j,t})e_k^\top x_t\\ &= [\mathbb{E}_{x_0^{(i)},\xi^{(i)}}\xi^{(i)}\sigma'(\sqrt{\bar{\alpha}_t}c_{i,j,t}x_0^{(i)} + \sqrt{1-\bar{\alpha}_t}c_{i,j,t}\xi^{(i)} + b_{i,j,t})][\mathbb{E}_{x_t^{(k)}}x_t^{(k)}]\\ &= 0.\end{aligned}$$

□

Besides, for a neuron that $a_{i,j,t} = 0$, from Lemma A.3 we know $w_{i,j,t} = 0$ and $b_{i,j,t} = 0$, so $\frac{d}{ds}a_{i,j,t} = 0$. So $a_{i,j,t}$ will keep zero along the trajectory. THus M is indeed an invariant set under gradient flow.

A.4 Proof for the implicit training bias Theorem 5.3

Here we consider a fixed t and target dimension i, and we omit the subscripts of t and i for the simplicity of notations. Thus for the network $s_\theta(x) = \sum_{j\in[m]} a_j\sigma(w_j^\top x + b_j)$, we optimize it via GF on the square loss $\mathcal{L}(\theta) = \mathbb{E}_{x_t=x}(s_\theta(x) - y_i(x))^2$ as

$$\begin{aligned}\frac{d}{ds}a_j &= -2\mathbb{E}_x(s_\theta(x) - y_i(x))\sigma(w_j^\top x + b_j)\\ \frac{d}{ds}w_j &= -2\mathbb{E}_x(s_\theta(x) - y_i(x))a_j\sigma'(w_j^\top x + b_j)x\\ \frac{d}{ds}b_j &= -2\mathbb{E}_x(s_\theta(x) - y_i(x))a_j\sigma'(w_j^\top x + b_j)\end{aligned}$$

We write $\theta(s)$ to denote the value of the parameters at time s. First we observe that the two layers of the network stay balanced throughout the course of the training process.

Lemma A.3. $\frac{d}{ds}(a_j^2 - \|w_j\|^2 - b_j^2) = 0.$

Proof. This is obtained directly as

$$\begin{aligned}\frac{d}{ds}(a_j^2 - \|w_j\|^2 - b_j^2) &= 2a_j\frac{d}{ds}a_j - 2w_j^\top\frac{d}{ds}w_j - 2b_j\frac{d}{ds}b_j\\ &= -4\mathbb{E}_x(s_\theta(x) - y(x))a_j\left[\sigma(w_j^\top x + b_j)\right.\\ &\left.-\sigma'(w_j^\top x + b_j)(w_j^\top x + b_j)\right]\\ &= 0.\end{aligned}$$

□

Therefore $a_j = \mathrm{sgn}(a_j)\sqrt{\|w_j\|^2 + b_j^2}$ through out the process.

A.4.1 Growth Rate For the First Layer Weight

Inspired by the G-function (Maennel et al., 2018; Lyu et al., 2021), when the initialization is very small, the neural network output $s_\theta(x) \approx 0$, so we can expand the loss as

$$\mathcal{L}(\theta) = \mathbb{E}_x(s_\theta(x) - y_i(x))^2 = \mathbb{E}_x y_i(x)^2 - 2s_\theta(x)y_i(x) + O(s_\theta^2(x))$$

So the initial trajectory of the neural network aims to optimize a surrogate loss $\tilde{\mathcal{L}}(\theta) = \mathbb{E}_x - 2s_\theta(x)y_i(x)$. We define the parameters $\tilde{\theta} = (\tilde{a}_j, \tilde{w}_j, \tilde{b}_j)$ to be the parameters run specifically for the surrogate loss, namely, let $\tilde{\theta}(0) = \theta(0)$ and

$$\begin{aligned}\frac{d}{ds}\tilde{a}_j &= 2\mathbb{E}_x y_i(x)\sigma(\tilde{w}_j^\top x + \tilde{b}_j)\\ \frac{d}{ds}\tilde{w}_j &= 2\mathbb{E}_x y_i(x)\tilde{a}_j\sigma'(\tilde{w}_j^\top x + \tilde{b}_j)x\\ \frac{d}{ds}\tilde{b}_j &= 2\mathbb{E}_x y_i(x)\tilde{a}_j\sigma'(\tilde{w}_j^\top x + \tilde{b}_j).\end{aligned}$$

We will have similarly, $\frac{d}{ds}(\tilde{a}_j^2 - \|\tilde{w}_j\|^2 - \tilde{b}_j^2) = 0$, so $\tilde{a}_j = \mathrm{sgn}(\tilde{a}_j)\sqrt{\|\tilde{w}_j\|^2 + \tilde{b}_j^2}$ through out the process. Then we can actually show that the scale of $\tilde{a}_j$ grows exponentially as a function of the direction of $(\frac{\tilde{w}_j}{|\tilde{a}_j|}, \frac{\tilde{b}_j}{|\tilde{a}_j|})$ as

Lemma A.4. *There exists a function $K : S^d \to \mathbb{R}$ such that*

$$|\tilde{a}_j(s)| = |\tilde{a}_j(0)| \exp\left(2\int_0^s K\left(\frac{\tilde{w}_j(\tau)}{|\tilde{a}_j|(\tau)}, \frac{\tilde{b}_j(\tau)}{|\tilde{a}_j|(\tau)}\right) d\tau\right)$$

when $\tilde{a}_j(0) > 0$, and

$$|\tilde{a}_j(s)| = |\tilde{a}_j(0)| \exp\left(-2\int_0^s K\left(\frac{\tilde{w}_j(\tau)}{|\tilde{a}_j|(\tau)}, \frac{\tilde{b}_j(\tau)}{|\tilde{a}_j|(\tau)}\right) d\tau\right)$$

when $\tilde{a}_j(0) < 0$.

Proof. We know

$$\begin{aligned}\frac{d}{ds}\tilde{a}_j &= 2\mathbb{E}_x y_i(x)\sigma(\tilde{w}_j^\top x + \tilde{b}_j)\\ &= 2|\tilde{a}_j|\mathbb{E}_x y_i(x)\sigma(\frac{\tilde{w}_j}{|\tilde{a}_j|}^\top x + \frac{\tilde{b}_j}{|\tilde{a}_j|})\end{aligned}$$

The proof is then done by taking $K(w, b) = \mathbb{E}_x y_i(x)\sigma(w^\top x + b)$. Notice that we only query K when $\|w\|^2 + b^2 = 1$. ∎

Now we take a closer examination of the function K. Let e_i be the unit vector along the i-th dimension and $P_i = I - e_i e_i^\top$ be the projection matrix removing the i-th dimension. Since the data x is sampled through the process $x = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\xi$ for $x_0 \in \{\pm 1\}^d$ and $\xi \sim \mathcal{N}(0, I)$, we know

$$\begin{aligned}K(w, b) &= \mathbb{E}_x y_i(x)\sigma(w^\top x + b)\\ &= \mathbb{E}_{x_0,\xi}(\xi^{(i)}\sigma(\sqrt{\bar{\alpha}_t}w^\top x_0 + \sqrt{1-\bar{\alpha}_t}w^\top P_i\xi + \sqrt{1-\bar{\alpha}_t}w^{(i)}\xi^{(i)} + b))\end{aligned}$$

Define $A = \sqrt{\bar{\alpha}_t}w^\top x_0 + \sqrt{1-\bar{\alpha}_t}w^\top P_i\xi + b$, $B = \sqrt{1-\bar{\alpha}_t}w^{(i)}$. since both A, B are independent to $\xi^{(i)}$,

$$\begin{aligned}K(w, b) &= \mathbb{E}_{x_0,\xi}\xi^{(i)}\sigma(A + B\xi^{(i)})\\ &= \frac{1}{2}\mathbb{E}_A(B + |B|\mathrm{erf}(\frac{A}{\sqrt{2}B}))\end{aligned}$$

where we use the standard error function as $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-s^2}ds \in [-1, 1]$. Furthermore, define $C = (\sqrt{\bar{\alpha}_t}w^\top P_i x_0 + \sqrt{1-\bar{\alpha}_t}w^\top P_i\xi + b)^2 + (w^{(i)})^2$, $D = \frac{\sqrt{\bar{\alpha}_t}w^\top P_i x_0 + \sqrt{1-\bar{\alpha}_t}w^\top P_i\xi + b}{w^{(i)}}$, we know

$B = \text{sgn}(B)\sqrt{(1-\bar{\alpha}_t)\frac{C}{1+D^2}}$, and

$$K(w,b) = \frac{\sqrt{1-\bar{\alpha}_t}}{2}\left(\mathbb{E}\text{sgn}(B)\sqrt{\frac{C}{1+D^2}} + \mathbb{E}_{x_0^{(i)}=1}\frac{1}{2}\sqrt{\frac{C}{1+D^2}}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right.$$
$$\left. + \mathbb{E}_{x_0^{(i)}=-1}\frac{1}{2}\sqrt{\frac{C}{1+D^2}}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right)$$

Observe that as x_0 and ξ are independent with $\mathbb{E}x_0 = \mathbb{E}\xi = 0$, and x_0 have pairwise independent entries,

$$\mathbb{E}C|x_0^{(i)} = \mathbb{E}\bar{\alpha}_t(w^\top P_i x_0)^2 + (1-\bar{\alpha}_t)(w^\top P_i \xi)^2 + (w^{(i)})^2 + b^2 = \|w\|^2 + b^2 = 1.$$

When $w^{(i)} > 0$, since $\text{erf}(x) \in [-1,1]$, we always have $K(w,b) > 0$. In this case, by Jensen's inequality,

$$K(w,b) \leq \frac{\sqrt{1-\bar{\alpha}_t}}{2}\sup_{D\in\mathbb{R}}\frac{1}{\sqrt{1+D^2}}\left(1 + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right.$$
$$\left. + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right)$$

Symmetrically as erf is an odd function, when $w^{(i)} < 0$, there is $K(w,b) < 0$, and

$$K(w,b) \geq -\frac{\sqrt{1-\bar{\alpha}_t}}{2}\sup_{D\in\mathbb{R}}\frac{1}{\sqrt{1+D^2}}\left(1 + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right.$$
$$\left. + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right).$$

The maximum of $|K|$ is achieved when $\mathbb{E}C = (\mathbb{E}\sqrt{C})^2$ and $D = D^*$ that maximizes the above functions, namely when $w^\top P_i = 0$ and $\frac{b}{w^{(i)}} = D^*$. By the first-order condition of optimality, let

$$f(D) = \frac{1}{\sqrt{1+D^2}}\left(1 + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right) + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right)$$

then we know $\frac{d}{dD}f(D^*) = 0$, namely

$$\frac{1+(D^*)^2}{D^*\sqrt{2\pi(1-\bar{\alpha}_t)}}\left(\exp\left[-\left(\frac{D^*}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)^2\right] + \exp\left[-\left(\frac{D^*}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)^2\right]\right)$$
$$= \left(1 + \frac{1}{2}\text{erf}\left(\frac{D^*}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right) + \frac{1}{2}\text{erf}\left(\frac{D^*}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right)$$

Lemma A.5. $|K(w,b)| \leq \sqrt{1-\bar{\alpha}_t}$.

Proof. By symmetry, WLOG we consider the case where $w^{(i)} > 0$. Then

$$\begin{aligned}K(w,b) &\leq \frac{\sqrt{1-\bar{\alpha}_t}}{2}\sup_{D\in\mathbb{R}}\frac{1}{\sqrt{1+D^2}}\left(1 + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} + \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right.\\ &\quad\left. + \frac{1}{2}\text{erf}\left(\frac{D}{\sqrt{2(1-\bar{\alpha}_t)}} - \sqrt{\frac{\bar{\alpha}_t}{2(1-\bar{\alpha}_t)}}\right)\right)\\ &\leq \frac{\sqrt{1-\bar{\alpha}_t}}{2}(1 + \frac{1}{2} + \frac{1}{2})\\ &= \sqrt{1-\bar{\alpha}_t}.\end{aligned}$$

□

Next we examine the dynamics of a neuron along this optimal direction $\left(\frac{\tilde{w}_j(\tau)}{|\tilde{a}_j|(\tau)}, \frac{\tilde{b}_j(\tau)}{|\tilde{a}_j|(\tau)}\right) = \frac{1}{\sqrt{1+(D^*)^2}}(e_i, D^*)$ with $w^\top P_i = 0$ and $\frac{b}{w^{(i)}} = D^*$. By Theorem 5.2 we know that the gradient of $\tilde{w}_j$ is also along the direction of e_i; furthermore, direct calculation plugging the above first-order condition, we arrive at

$$\frac{\frac{d}{ds}b}{\frac{d}{ds}w^{(i)}} = \frac{\mathbb{E}_{x_0,\xi}\xi\sigma'((\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\xi + D^*)w^{(i)})}{\mathbb{E}_{x_0,\xi}\xi(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\xi)\sigma'((\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\xi + D^*)w^{(i)})} = D^*.$$

This shows that a neuron along the direction $\left(\frac{\tilde{w}_j(\tau)}{|\tilde{a}_j|(\tau)}, \frac{\tilde{b}_j(\tau)}{|\tilde{a}_j|(\tau)}\right) = (w, b) = \frac{1}{\sqrt{1+(D^*)^2}}(e_i, D^*)$ keeps the same direction during the course of the dynamics, thus the weight $\tilde{a}_j$ can maintain the maximum growing rate.

A.5 Experimental Details

The details of experiment formulation is as below. Recall that a text distribution includes a set of discrete symbols $\mathcal{S} = \{s_1, s_2, \ldots, s_K\}$ and a spelling/grammar rule P_G. A list of symbol tokens are further rendered into ambient space by a function $h : \mathcal{S} \mapsto \mathbb{R}^d$ which maps each symbol to a vector in ambient space like image pixels or a single scalar. The full signal is obtained by concatenating these vectors. We describe $\mathcal{S}$ and P_G we used in experiments.

Parity: There are only two symbols $\mathcal{S} = \{1, -1\}$. The rule is that there needs to be even number of symbol s_1. Namely $P_G(\mathcal{I}) = \frac{1}{2^{L-1}} \cdot \mathbb{I}\left[\prod_{j=1}^{L} s_{i_j} = 1\right]$. The ambient space rendering function can either by a single scalar $h(s_i) = s_i$. It can also be two pixel image or embedding vector templates in ambient observation space $\{\boldsymbol{o}_{-1}, \boldsymbol{o}_1\}$ and $h(s_i) = \boldsymbol{o}_{s_i}$.

Quarter-MNIST: We combine four MNIST digits' image to become a whole figure. the symbol system is all digits $\mathcal{S} = [9]$. We fix the length $L = 4$ and requires that $s_1 + s_2 = s_3 + s_4$, and we have $P_G(\mathcal{I}) = \frac{1}{Z} \cdot \mathbb{I}[s_1 + s_2 = s_3 + s_4]$, $Z = 670$ is some normalization constant. The ambient space rendering function is a probabilistic image drawing function $h : \{0, \ldots, 9\} \mapsto \mathbb{R}^d$ which maps each digit to its hand-writing image.

Dyck: We also test dyck grammar, where $\mathcal{S} = \{+1, -1\}$. A dyck sequence must have even number of tokens $s_1, \ldots, s_{2k}$, and satisfy $\sum_{j=1}^{i} s_i \geq 0$ for $i \in [2k]$. Also it requires $\sum_{j=1}^{2k} s_i = 0$. This can be regarded as a valid operation sequence for a stack where $+1$ means push and -1 means pop. The requirement essentially means the stack cannot pop if it is empty, also it needs to be empty at start and end. The rendering function is similar as parity. In our experiment we use left and right parenthesis to represent $+1$ and -1, respectively.

As for denoising networks, we use attention-augmented UNet, where each block is equipped with linear attention and middle bottleneck equipped with full attention. The image size is 64 and the initial hidden width is 64, which means the bottleneck dimension is 512. We also adopted standard DiT-B model with hidden size 384 and patch size 8. We train with Adam optimizer with $lr = 8 \times 10^{-5}$, batch size $bs = 16$, total schedule ranging from 160k to 700k iterations.

Training Schedule and Model Details We use mainly two types of model for training in our experiments, namely DiT and UNet augmented with attention. The DiT is standard DiT-S model, with $33M$ parameter. The UNet initial channel is 64, and total parameter is $\sim 35.7M$. We training the score matching objective with equal λ_t. The training batch size is 16, 180k iteration for Quater-MNIST and $1.1M$ iteration for parity parathensis images.

A.6 Recent Model's Results

We also conduct experiment on most recent models such as StableDiffusion 3.5-medium (Esser et al., 2024) and FLUX1-dev. We use prompt that requires the model to generate rich text content without specifying concrete words. For instance, "A piece of calligraphy art", "A newspaper reporting news", "A blackboard with formulas". And here are the test results, we can see that text hallucination is still ubiquitous. The seeds of six images of each model under same prompt are from 0 to 2, so all people can reproduce these results.

Figure 8: Visualizations of StableDiffusion 3.5's results on prompt "A blackboard with formulas"

Figure 9: Visualizations of StableDiffusion 3.5's results on prompt "A piece of calligraphy art."

Figure 10: Visualizations of StableDiffusion 3.5's results on prompt "A newspaper reporting news."

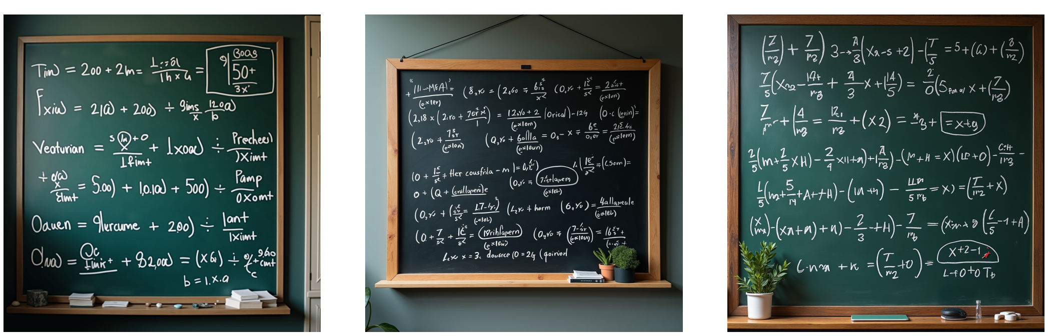


Figure 11: Visualizations of FLUX1's results on prompt "A blackboard with formulas"

Figure 12: Visualizations of FLUX1's results on prompt "A piece of calligraphy art."

Figure 13: Visualizations of FLUX1's results on prompt "A newspaper reporting news."

A.7 Visualizations

In this part, we will show detailed visualizations of our experiments. For each experiment, we visualize following

- Generated hallucination samples.
- The trend of LDR along training process.
- The heatmap of LDR at some critical denoising timestep.

A.7.1 Parity Parenthesis

Please refer to figure 7 for details of generated hallucination sample and LDR analysis. The LDR heatmap is in figure 8.

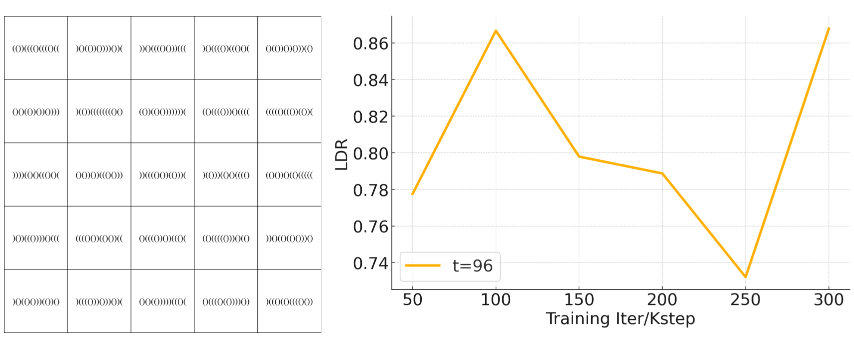


Figure 14: Some examples of generated hallucination samples at 300k steps. Note that only half of them satisfy parity constraint (even number of both parenthesis). The LDR at $\sqrt{\bar{\alpha}_t} = 0.1$ is high through all training procedure.

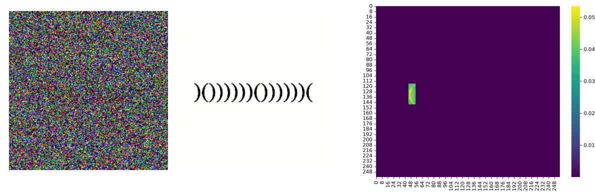


Figure 15: An example of x_t, x_0 and LDR heatmap. The LDR in this image is 0.9736 and the reference region is the second parenthesis. We can see the denoising model primarily only focues on this parenthesis' region to generate it. Therefore all the symbols are generated independent and fail to satisfy parity constraint.

A.7.2 Dyck Parenthesis.

Perhaps surprisingly, we found that UNet model is capable of generating valid dyck sequences. After 60k iterations, the UNet model drops down and the accuracy for generated image increases. This can also be validated from probing a parenthesis' region to see which part of input noise the model is looking at. We found that model will only focus on local noise at hallucination phase, resulting in a high LDR. And when it overfits the data, the saliency map spreads globally and LDR decreases. See figure 10 for more details.

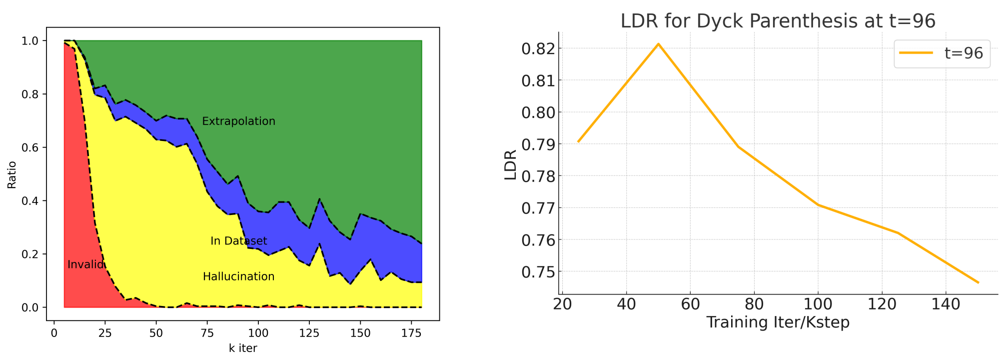


Figure 16: Generation proportion graph and LDR for $t = 96$. The reference region is the position at second parenthesis. Although seemed difficult, Dyck grammar is actually much easier to learn and extrapolate, since there are strong correlations between parenthesis. Interesting, there is still a hallucination phase, and hallucinations fades as LDR decreases.

A.7.3 Quarter MNIST.

The LDR analysis is shown in main content. Here we show some hallucinated generation and heatmap of LDR. As shown in figure 11, we can see that both UNet and DiT generate the top-left digit solely by local region's noise. As a consequence, these four digits are generated independently, therefore can not capture the innate relationship and rules.

A.7.4 English word and Chinese characters.

Does hallucination in real-world text distribution also stem from local generation bias? We run experiments to verify this mechanism with image distribution contains common English words and Chinese characters. These English words are

[a, abandon, ability, able, about, above, accept, according, account, across, act, action,activity, actually, add, address, administration, admit, adult, affect, after, again,against, age, agency, agent, ago, agree, agreement, ahead, air, all, allow, almost, alone,along, already, also, although, always, American, among, amount, analysis, and, animal, another, answer, any, anyone, anything, appear, apply, approach, area, argue, arm, around,arrive, art, article, artist, as, ask, assume, at, attack, attention, attorney, audience,author, authority, available, avoid, away, baby, back, bad, bag, ball, bank, bar, base,

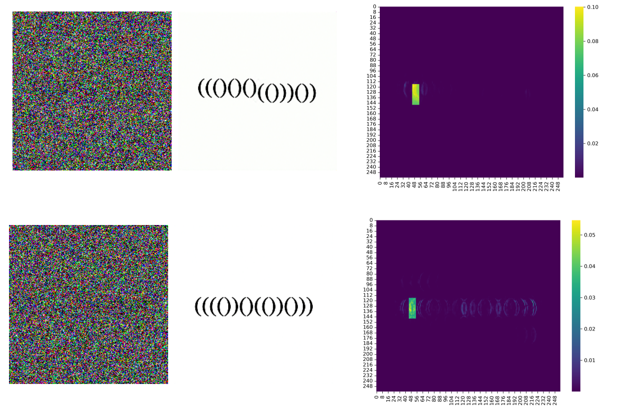


Figure 17: The LDR analysis for 20k training steps (first row, in hallucination) and 170k training steps (second row, correctly extrapolate). We can see a discrepancy for model's behaviors in terms of local v.s. global dependency. When model learns to correctly generate symbols, it will attend to overall region for coordinating different symbols, which means LDR is low. From left to right columns are x_t, x_0 and LDR heatmap.

Figure 18: DiT generated Hallucinated samples for Quarter-MNIST dataset. Each four digit form a sample

be,beat,beautiful, because, become, bed, before, begin, behavior, behind, believe, benefit, best,better, between, beyond, big, bill, billion, bit, black, blood, blue, board, body,book,born, both, box, boy, break, bring, brother, budget, build, building, business, but,buy,by, call, camera, campaign, can, cancer, candidate, capital, car, card, care, career,carry,case, catch, cause, cell, center, central, century, certain, certainly, chair, challenge,chance, change, character, charge, check, child, choice, choose, church, citizen, city,civil,claim, class, clear, clearly, close, coach, cold, collection, college, color, come,commercial,common, community, company, compare, computer, concern, condition, conference, Congress,consider,consumer, contain, continue, control, cost, could, country, couple, course, court, cover,create, crime, cultural, culture, cup, current, customer, cut, dark, data, daughter,day,dead, deal, death, debate, decade, decide, decision, deep, defense, degree, Democrat,democratic,describe, design, despite, detail, determine, develop, development, die, difference,different,difficult, dinner, direction, director, discover, discuss, discussion, disease, do, doctor,dog, door, down, draw, dream, drive, drop, drug, during, each, early, east, easy,eat,economic, economy, edge, education, effect, effort, eight, either, election, else,employee,end, energy, enjoy, enough, enter, entire, environment, environmental, especially,establish,even, evening, event, ever, every, everybody, everyone, everything, evidence, exactly,example,executive, exist, expect, experience, expert, explain, eye, face, fact, factor, fail,fall,family, far, fast, father, fear, federal, feel, feeling, few, field, fight, figure,fill,film, final, finally, financial, find, fine, finger, finish, fire, firm, first, fish,five,floor, fly, focus, follow, food, foot, for, force, foreign, forget, form, former,forward,four, free, friend, from, front, full, fund, future, game, garden, gas, general,generation,get, girl, give, glass, go, goal, good, government, great, green, ground,

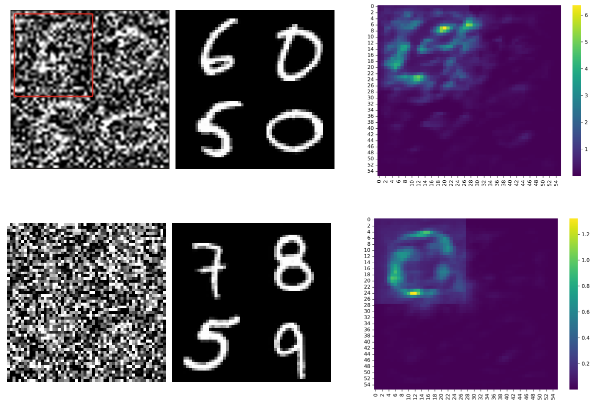


Figure 19: The LDR analysis for UNet (top row) and DiT (second) learning Quarter-MNIST dataset. $\sqrt{\bar{\alpha}_t} = 0.1$ and the reference region is top-left quarter. From left to right columns are x_t, x_0 and LDR heatmap.

group, grow,growth, guess, gun, guy, hair, half, hand, hang, happen, happy, hard, have, he,head,health, hear, heart, heat, heavy, help, her, here, herself, high, him, himself, his,history,hit, hold, home, hope, hospital, hot, hotel, hour, house, how, however, huge, human,hundred,husband, I, idea, identify, if, image, imagine, impact, important, improve, in, include,including, increase, indeed, indicate, individual, industry, information, inside, instead,institution,interest, interesting, international, interview, into, investment, involve, issue, it, item,its,itself, job, join, just, keep, key, kid, kill, kind, kitchen, know, knowledge, land,language,large, last, late, later, laugh, law, lawyer, lay, lead, leader, learn, least, leave,left,leg, legal, less, let, letter, level, lie, life, light, like, likely, line, list,listen,little, live, local, long, look, lose, loss, lot, love, low, machine, magazine, main,maintain,major, majority, make, man, manage, management, manager, many, market, marriage, material,matter,may, maybe, me, mean, measure, media, medical, meet, meeting, member, memory, mention,message,method, middle, might, military, million, mind, minute, miss, mission, model, modern,moment,money, month, more, morning, most, mother, mouth, move, movement, movie, Mr, Mrs,much, music,must, my, myself, name, nation, national, natural, nature, near, nearly, necessary,need, network,never, new, news, newspaper, next, nice, night, no, none, nor, north, not, note,nothing, notice,now, n't, number, occur, of, off, offer, office, officer, official, often, oh, oil,ok, old,on, once, one, only, onto, open, operation, opportunity, option, or, order,organization, other,others, our, out, outside, over, own, owner, page, pain, painting, paper, parent,part, participant,particular, particularly, partner, party, pass, past, patient, pattern, pay, peace,people, per,perform, performance, perhaps, period, person, personal, phone, physical, pick, picture,piece, place,plan, plant, play, player, PM, point, police, policy, political, politics, poor,popular, population,position, positive, possible, power, practice, prepare, present, president, pressure,pretty, prevent,price, private, probably, problem, process, produce, product, production, professional,professor, program,project, property, protect, prove, provide, public, pull, purpose, push, put, quality,question, quickly,quite, race, radio, raise, range, rate, rather, reach, read, ready, real, reality,realize, really,reason, receive, recent, recently, recognize, record, red, reduce, reflect, region,relate, relationship,religious, remain, remember, remove, report, represent, Republican, require, research,resource, respond,response, responsibility, rest, result, return, reveal, rich, right, rise, risk, road,rock, role,room, rule, run, safe, same, save, say, scene, school, science, scientist, score, sea,season, seat,second, section, security, see, seek, seem, sell, send, senior, sense, series, serious,serve, service,set, seven, several, sex, sexual, shake, share, she, shoot, short, shot, should,shoulder, show, side,sign, significant, similar, simple, simply, since, sing, single, sister, sit, site,situation, six, size,skill, skin, small, smile, so, social, society, soldier, some, somebody, someone,something, sometimes,son, song, soon, sort, sound, source, south, southern, space, speak, special, specific,speech, spend,sport, spring, staff, stage, stand, standard, star, start, state, statement, station,stay, step, still,stock, stop, store, story, strategy, street, strong, structure, student, study, stuff,style, subject, success,successful, such, suddenly, suffer, suggest, summer, support, sure, surface, system,table, take, talk, task,tax, teach, teacher, team, technology, television, tell, ten, tend, term, test, than,thank, that, the, their,them, themselves, then, theory, there, these, they, thing, think, third, this, those,though, thought, thousand,threat, three, through, throughout, throw, thus, time, to, today, together, tonight,too, top, total, tough, toward,town, trade, traditional, training, travel, treat, treatment, tree, trial, trip, trouble,true, truth, try, turn,TV, two, type, under, understand, unit, until, upon, use, usually, value, various,very, victim, view, violence,visit, voice, vote, wait, walk, wall, want, war, watch, water, way, we, weapon,wear, week, weight, well, west,western, what, whatever, when, where, whether, which, while, white, who, whole, whom,whose, why, wide, wife, will,win, wind, window, wish, with, within, without, woman, wonder, word, work, worker,world, worry, would, write,writer, wrong, yard, yeah, year, yes, yet, you, young, your, yourself].

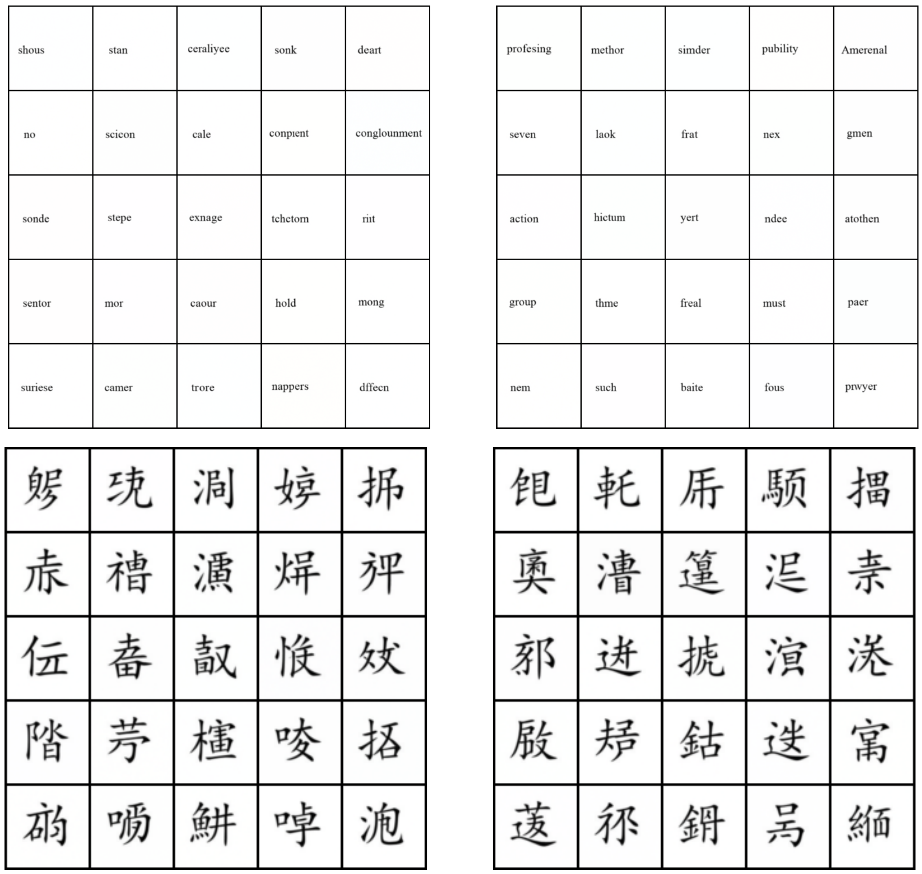


Figure 20: Diffusion generated results when trained on English common words' image (first row) and Chinese characters (second row). We find similar misspelling phenomenon for English generation and glyph by randomly assembling radicals in Chinese characters.

Also we construct a dataset using 3,000 common Chinese characters and render them in Kai font images. We use UNet to learn to generate images of these texts. The early stage generation results are shown in figure 14. Interestingly, we observe very similar pattern as in modern large scale diffusion models like StableDiffusion and Midjourney in our synthetic experiment. We probe denoising model at stage when it has hallucination, and finds that they all have very high LDR, indicating they generate letter or radicals independently and combine them.

A.8 Validation of Theoretical Findings

In this section, we corroborate our theoretical findings by experiments. We set $d = 8, 16$ and learn just the first dimension of score function for parity points using a two-layer ReLU-activated MLP. The model has 2000 hidden neurons and we set initialization scheme $\sigma_{init} = 1e-3, b_i(0) = 0, a_i = \|w_i\|^2$. We train with small learning rate $\eta = 1e-6$ and discover following interesting phenomena.

- The loss curves exhibit a stair-like shape, meaning it has three phases.

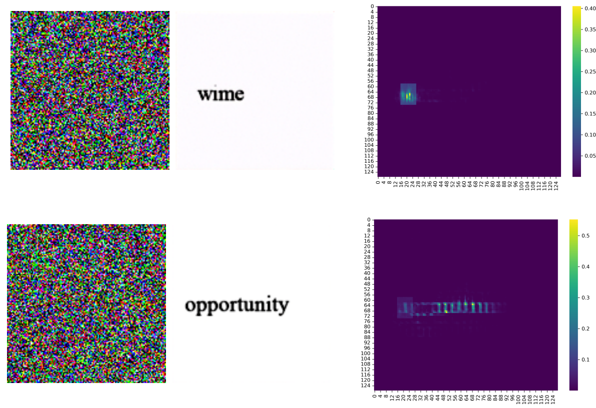
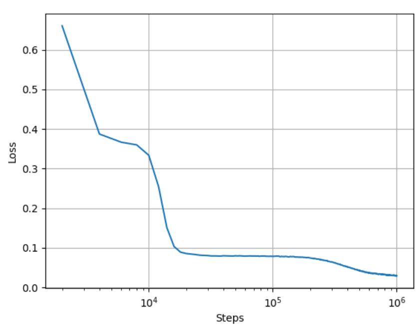
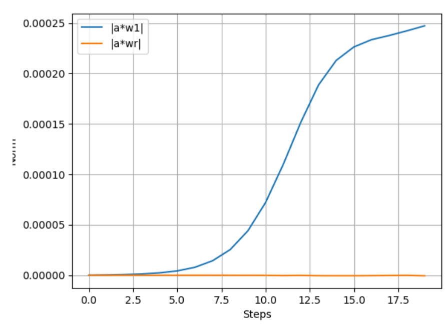


Figure 21: The LDR analysis for model at 20k steps (first row) and 200k steps (second row) learning on common English words dataset. $\sqrt{\bar{\alpha}_t} = 0.1$ and the reference region is the first and second letter. We can see that when model hallucinates, it only attends to local region, therefore randomly spelling the letters. It will account globally when overfitting to reproduce words within the training dataset. From left to right columns are x_t, x_0 and LDR heatmap.

- These three phases correspond to best linear interpolation, best univariate interpolation, and optimal approximator.
- At initial stage, the network's weight w_i aligns well with e_1, and stick with this state through the first and second stage.

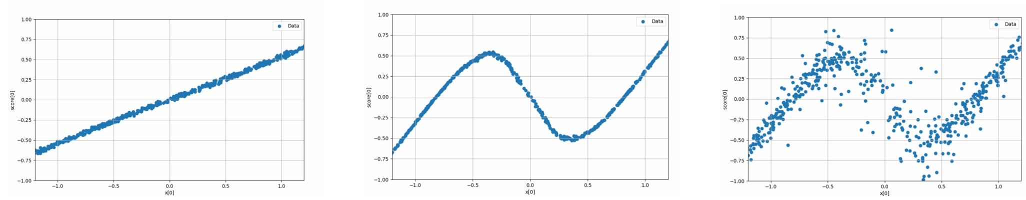


(a) Three stairs shape loss. Each stage represents a saddle point.



(b) The average norm for the first dimension and the rest of weight parameter among hidden neurons. In the first stage, the model only extracts the input's first dimension's information, resulting in a local and sparse input dependency.

As shown in figure 17. While the ground truth score function is not a univariate function of $x^{(1)}$ as in left. The shaded area near the origin means the score function has also dependency on other input dimensions $x^{(j)}, j > 1$. However,The MLP is biased towards learning a univariate function. Even though MLP has access to value from all input dimensions. This results in a local generation bias and let the model independently sample each dimension. Therefore this model essentially samples on all vertices on hypercube rather than parity points.



(c) Phase 1: optimal linear interpolation. (d) Phase 2: optimal univariate interpolation. (e) Phase 3: global optimal multivariate approximation.

Figure 22: Three-phase functionality of learned MLP score network. The x-axis is $x^{(1)}$. The MLP performs local generation, if its output against $x^{(i)}$ is nearly a function curve with no ambiguity in mapping.

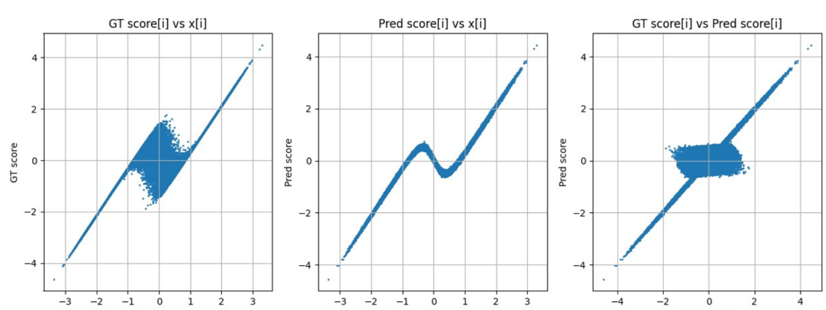


Figure 23: The local generation bias in MLP.