CAN DIFFUSION MODELS LEARN HIDDEN INTER-FEATURE RULES BEHIND IMAGES?

Yujin Han[*] The University of Hong Kong
Andi Han[*] RIKEN AIP
Wei Huang RIKEN AIP
Chaochao Lu Shanghai AI Laboratory
Difan Zou[†] The University of Hong Kong

ABSTRACT

Despite the remarkable success of diffusion models (DMs) in generation, they exhibit specific failure cases with unsatisfactory outputs. We focus on one such limitation: the ability of DMs to learn hidden rules between image features. Specifically, for image data with dependent features ($\mathbf{x}$) and ($\mathbf{y}$) (e.g., the height of the sun ($\mathbf{x}$) and the length of the shadow ($\mathbf{y}$)), we investigate whether DMs can accurately capture the inter-feature rule ($p(\mathbf{y}|\mathbf{x})$). Empirical evaluations on mainstream DMs (e.g., Stable Diffusion 3.5) reveal consistent failures, such as inconsistent lighting-shadow relationships and mismatched object-mirror reflections. Inspired by these findings, we design four synthetic tasks with strongly correlated features to assess DMs' rule-learning abilities. Extensive experiments show that while DMs can identify coarse-grained rules, they struggle with fine-grained ones. Our theoretical analysis demonstrates that DMs trained via denoising score matching (DSM) exhibit constant errors in learning hidden rules, as the DSM objective is not compatible with rule conformity. To mitigate this, we introduce a common technique - incorporating additional classifier guidance during sampling, which achieves (limited) improvements. Our analysis reveals subtle signals of fine-grained rules are challenging for the classifier to capture, providing insights for future exploration.

1 INTRODUCTION

Despite the remarkable capabilities demonstrated by diffusion models (DMs) in generating realistic images (Ho et al., 2020; Song et al., 2020; Vahdat et al., 2021; Dhariwal & Nichol, 2021; Karras et al., 2022; Tian et al., 2024b), videos (Ho et al., 2022; Yu et al., 2024; Yuan et al., 2024), and audio (Liu et al., 2023a; Yang et al., 2024; Lemercier et al., 2024), they still encounter specific failures in synthesis quality, such as anatomically incorrect human poses (Borji, 2023; Zhang et al., 2024; Huang et al., 2024) and misalignment between generated content and prompts (Feng et al., 2022; Borji, 2023; Chefer et al., 2023; Liu et al., 2023b; Lim & Shim, 2024), which could harm the reliability and applicability of DMs in real-world scenarios. We focus on a specific type of failure with limited attention: the failure of DMs in learning hidden inter-feature rules behind images. Specifically, consider image data containing dependent feature pairs $(\mathbf{x}, \mathbf{y})$, such as the height of the sun ($\mathbf{x}$) affecting the length of a pole's shadow ($\mathbf{y}$). Our investigation centers on whether DMs targeting the joint distribution $p(\mathbf{x}, \mathbf{y})$ can accurately capture the underlying relationships between $\mathbf{x}$ and $\mathbf{y}$, effectively recovering the conditional distribution $p(\mathbf{y}|\mathbf{x})$. Theoretically, a diffusion model that perfectly estimates the joint distribution should naturally capture the conditional distribution, thereby learning the latent rules between features. However, in practice, numerous factors, such as non-negligible score function estimation errors, can cause the sampled joint distribution to deviate significantly from the true distribution (Chen et al., 2022; 2023; Benton et al., 2024). How do these deviations propagate to inter-feature rule learning? This gap between theory and practice remains largely unexplored.

[*]Equal contribution: yujinhan@connect.hku.hk, andi.han@riken.jp
[†]Corresponding author: dzou@cs.hku.hk

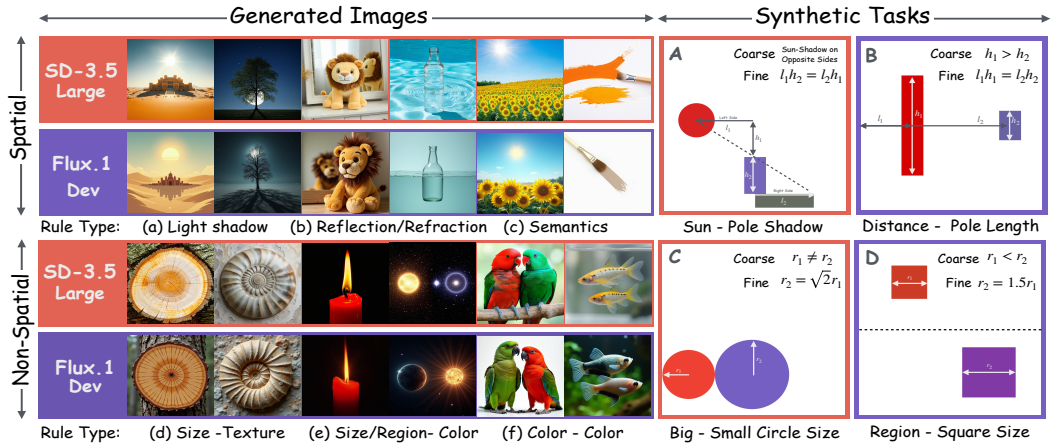


Figure 1: **Synthetic Tasks Inspired by Real-World Insights.** Based on whether inter-feature rules involve spatial dependencies, we categorize the failure cases into spatial and non-spatial rules. **Spatial rules** include: (a) Light-shadow, where evaluated DMs generate unreasonable multiple shadows or incorrect shadow flips; (b) Reflection/Refraction, showing incorrect mirror rules or missing refraction effects below water surface; (c) Semantics, such as inconsistencies between sunflower orientation and sun position, or brush and canvas colors. **Non-spatial rules** involve: (d) Size-Texture, like mismatches between tree diameter and growth rings; (e) Size/Region-Color, where evaluated models fail to capture burning candle's color variations and star size-color relationships (e.g., red giants and white dwarf); (f) Color-Color, as in Eclectus parrots' body-beak color correlations that DMs fail to maintain. Appendix C provides detailed explanations for each case. These failures of mainstream DMs in handling real-world inter-feature rules inspire our design of four synthetic tasks.

Although existing studies have explored whether DMs can learn specific rules, they primarily focus on independent features, such as DMs' compositional capabilities (Okawa et al., 2024; Deschenaux et al., 2024; Wiedemer et al., 2024). Some works have investigated inter-feature dependencies in DMs, but the varying complexity of rules has led to contradictory findings. For example, DDPM has been reported to fail in generating images satisfying numerical equality constraints (Anonymous, 2025), while succeeding in reasoning about shape patterns in RAVEN task (Wang et al., 2024a). These inconsistencies highlight the need for a unified experimental setting that allows for adjustable rule difficulty, enabling an accurate evaluation of DMs' rule-learning capabilities. Moreover, existing studies rely heavily on empirical observations, lacking theoretical analysis to elucidate the limitations of DMs in rule conformity. Due to space limitations, Appendix B provides a more detailed discussion of existing work and its differences from this study.

Our investigation into inter-feature rules begins with observing the limited ability of mainstream DMs (e.g., `SD-3.5 Large`, `Flux.1 Dev`) to capture real-world inter-feature rules, as illustrated in Fig.1, although these models perform well on metrics like FID [1]. Their errors in inter-feature rules are evident in various scenarios, such as inconsistent relationships between sun positions and building shadows, mismatched reflections of toys in mirrors, and sunflowers failing to face the sun. Then, we carefully design four synthetic tasks to reflect real-world rule failures, ensuring the practical relevance of our findings. The rule of each task features two difficulty levels: coarse-grained rules (e.g., the sun and a pole's shadow should be on opposite sides) and fine-grained rules (e.g., the shadow's length as a precise function of the sun's height). This hierarchical, controllable framework enables a comprehensive evaluation of DMs' abilities. Next, through extensive experiments considering various factors including model architectures, training data size, and image resolution, we reach a consistent conclusion: *DMs effectively learn coarse-grained rules but struggle with fine-grained ones*.

Furthermore, we develop a rigorous theoretical analysis using a multi-patch data model with an inter-feature rule specified in terms of norm. We prove a constant error lower bound on learning the hidden rule via optimizing the DSM objective (Ho et al., 2020) with a two-layer network. This demonstrates the incompatibility between learning joint distributions and identifying specific inter-feature rules.

Recognizing DMs' difficulty in learning inter-feature rules, we mitigate this issue by constructing contrastive pairs that satisfy either fine-grained or coarse-grained rules and then using them to train a classifier as additional guidance. While this strategy enhances rule-compliant sample generation, further improvements are still achievable. The in-depth analysis identifies that fine-grained rules

[1] Appendix A lists Mixture Gaussian as an example to demonstrate that low FID and incorrect inter-feature relationships in DMs' generations are not contradictory.

exhibit weak signals, making accurate classifier training particularly challenging. We summarize our **key contributions** as follows:

Empirically, inspired by mainstream DMs' struggles with real-world inter-feature rules, we innovatively create synthetic tasks with coarse/fine-grained rules to systematically assess DMs' rule learning ability in Section 2.1.1. Extensive experiments in Section 2.3 show that while DMs can learn coarse rules, their ability to grasp precise rules is limited.

Theoretically, we rigorously analyze DMs on a synthetic multi-patch data distribution with a hidden norm dependency in Section 3. We prove that the unconditional DDPM cannot learn the precise rule of norm constraint, which exhibits at least a constant error in approximating the desired score function. This identifies the limitation of the current DMs training paradigm and necessitates further improvements for learning hidden rules behind images.

Methodologically, we mitigate DMs' inability to learn fine-grained rules by introducing guided diffusion with a contrastive-trained classifier in Section 4. However, the challenges of accurately classifying fine-grained rules identify room for improvement in our strategy. This problem, distinct from traditional classification tasks, involves detecting subtle distinctions between fine-grained and coarse-grained rules, highlighting valuable insights for future exploration.

2 Exploring Inter-feature Rule Learning via Synthetic Tasks

In real-world image generation tasks, rules between features are often complex and difficult to define or quantify precisely. To systematically investigate DMs' ability in rule learning, we design simplified and controllable synthetic tasks in Fig.1. These synthetic tasks not only provide explicitly defined inter-feature rules but also abstract essential feature rules present in real-world data, thereby making our conclusions practically relevant. For example, Synthetic Task A in Fig.1 simulates the *Light-Shadow* relationship, while Task B simplifies the physical rules of *Reflection/Refraction*.

2.1 Synthetic Tasks Inspired by Real-World Insights

2.1.1 Real-World Hidden Inter-Feature Rules

Following Borji (2023), we explore common inter-feature rules (fig. 1) and classify them into *spatial* and *non-spatial* rules based on whether they stem from spatial arrangements or feature attributes.

Spatial Rules are defined as constraints on the relative positions and layouts between features, such as the correlation between the sun's height and the shadow's length. In Fig.1, scenario *Light-shadow* demonstrates how the position of a light source should precisely determine the placement of building shadows. However, both 8-billion Multimodal `SD-3.5 Large`[2](Rombach et al., 2022) and 12-billion model `Flux.1 Dev`[3](Labs, 2023), fail to generate proper shadows, either producing incorrect directions or merely creating symmetrical duplicates of the actual buildings. Similarly, in scenario *Reflection/Refraction*, while objects in front of mirrors should dictate the layout of their reflections, we observe completely unreasonable generations from both models.

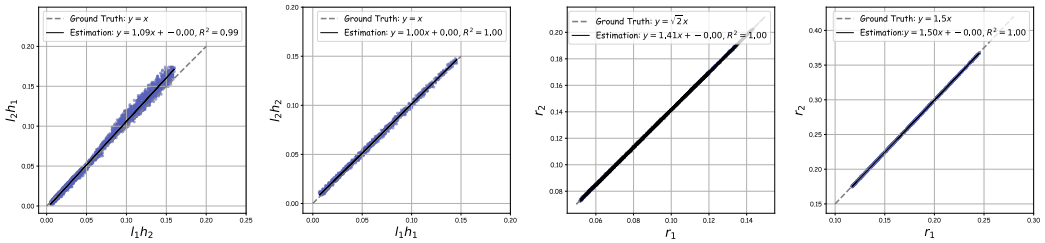
Non-Spatial Rules are defined as correlations between intrinsic feature attributes, such as the relationship between an object's size and its color. For instance, in type *Size -Texture*, tree trunk features should exhibit precise correlations between the diameter and annual ring count, and candle flames in type *Size/Region- Color* should show constrained relationships between different flame zones and their colors. However, these fine-grained inter-feature constraints are ignored by both `SD-3.5 Large` and `Flux.1 Dev`. More detailed discussion and additional experiments for more advanced DMs are deferred to Appendix C.

2.1.2 Synthetic Tasks

Inspired by real-world rules in Section 2.1.1, we design four synthetic tasks (A-D), each with two levels of rule granularity (coarse and fine), as shown in Fig.1. We provide a brief overview of synthetic tasks here, with more details presented in Appendix D. Specially, **Task A** is inspired by the spatial

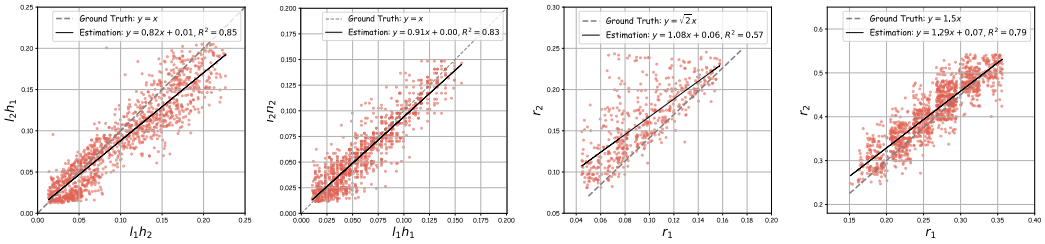
[2]https://huggingface.co/spaces/stabilityai/stable-diffusion-3.5-large

[3]https://fal.ai/models/fal-ai/flux/dev



(a) Task A (b) Task B (c) Task C (d) Task D

Figure 2: **Synthetic training data satisfies fine-grained rules.** To validate the evaluation method, we extract relevant features from the synthetic training data and check if they meet expectations, focusing on generations within $[2.5\%, 97.5\%]$ for stability. The closely matching Estimation and Ground Truth lines, along with an R^2 value near 1, demonstrate effectiveness of the evaluation method.



(a) Task A (b) Task B (c) Task C (d) Task D

Figure 3: **Generated data does not satisfy fine-grained rules.** Considering generated samples within $[2.5\%, 97.5\%]$, we extract focused features and check if they meet fine-grained rules. The Estimation line, far from the Ground Truth line, and an R^2 value less than 1, reveal DMs' failure in learning fine-grained rules. Appendix E.2 shows generated images that violate the fine-grained rules.

rules behind the *Light-shadow* case, simulating the physical law between the sun and pole shadows. In Task A of Fig.1, the *coarse-grained rule* requires the sun and shadow to be on opposite sides of the pole, while the *fine-grained rule* requires sun's center, pole top, and shadow endpoint align linearly, i.e., satisfying $l_1h_2 = l_2h_1$ (see notations in Task A, Fig.1).

Task B abstracts the spatial rule from *Reflection/Refraction* case, where an object's reflection size depends on its size and distance from the mirror. Task B uses two rectangles with lengths h_1 and h_2 (notations shown in Task B, Fig.1) to simulate this perspective rule, where size diminishes with distance. Assuming the viewpoint is at the leftmost edge, the *coarse-grained rule* requires the left rectangle (closer to the viewpoint) to be longer than the right one (farther from the viewpoint), i.e., $h_1 > h_2$, while *fine-grained rule* dictates rectangle lengths be proportional to their distances from the viewpoint, i.e., $l_1h_2 = l_2h_1$.

Task C consists of two tangent circles of different radii, aiming to capture the relationship between shape/outlook and size as illustrated in non-spatial rule. The *coarse-grained rule* simply requires distinct radii for the two circles, i.e., $r_1 \neq r_2$, while the *fine-grained rule* specifies a precise ratio between the radii, requiring $r_2 = \sqrt{2}r_1$.

Task D simplifies the non-spatial rule from *Size/Region- Color* in Fig.1, where, in candle flame generations, colors transition from blue near the wick to yellow at the outer regions. We construct two squares, with smaller squares positioned in the upper half and larger ones in the lower half of the image. The *coarse-grained rule* requires that the upper square's side length l_1 be smaller than the lower square's side length l_2, i.e., $l_1 < l_2$, while the *fine-grained rule* specifically requires $l_2 = 1.5l_1$.

2.2 Experimental and Evaluation Setup

Experimental Setup. In subsequent experiments, we train DDPM (Ho et al., 2020) on four synthetic tasks. Unlike latent-space DMs (e.g., `SD-3.5 Large`), pixel-space DDPM makes the conformity of inter-feature relationships potentially simpler, as no additional compression-induced information loss occurs (Rombach et al., 2022; Yao & Wang, 2025). Following the training setting (Aithal et al., 2024), we fix the total timesteps at $T = 1000$ and employ the widely-

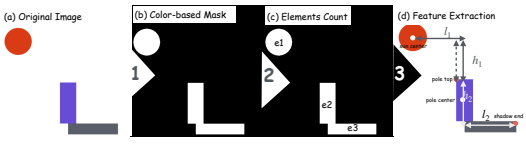


Figure 4: **Pipeline for extracting features.**

Table 1: **DMs satisfy coarse rules.** The invalid ratio is around 20%–40%. And DMs can learn coarse rules with one exception in Task A, which is visualized in appendix E.1.

Task	Invalid (%)	Coarse-Grained Violations
A	30.15	1
B	40.45	0
C	41.75	0
D	24.90	0

Table 2: **Comparison** between **DDPM**, **Guided DDPM** (Guidance), and **Filtered DDPM** (Filtering): Additional guidance and filtering improve generation with lower Error and higher R^2.

Task	Error ↓			R^2↑		
	DDPM	Guidance	Filtering	DDPM	Guidance	Filtering
A	0.25	0.21	0.17	0.85	0.90	0.90
B	0.11	0.10	0.05	0.83	0.85	0.86
C	0.41	0.26	0.25	0.57	0.67	0.64
D	0.46	0.43	0.39	0.79	0.84	0.85

used U-Net architecture (Ronneberger et al., 2015) as the denoiser. Appendix E.1 provides more details, including training data size and advanced architectures such as DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024).

Evaluation. To evaluate whether generated images follow inter-feature rules, Fig.4 designs a three-step feature extraction pipeline: (1) Color-based Mask: Segment element masks (e.g., sun, pole, shadow in Task A) based on predefined color (HSV) ranges when synthesizing training data; (2) Elements Count: Apply contour detection based on masks to verify the presence of essential elements, marking images as `Invalid` if any are missing; (3) Feature Extraction: Extract key feature points (e.g., sun center, pole top/center and shadow endpoint in Fig.4) and compute features of interest, such as horizontal sun-to-pole distance l_1, vertical sun-to-pole-top distance h_1, pole height h_2, shadow length l_2. All features are scaled to $[0, 1]$ by normalizing with image size to remove scale effects.

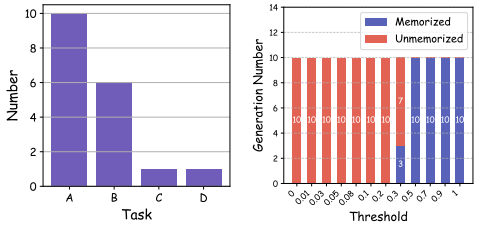
We verify whether generated images satisfy predefined rules using these extracted features. For example, in Task A, we check: (1) Coarse-grained rule: sun and shadow are on opposite sides of the pole; (2) Fine-grained rule: validate $l_1 h_2 = l_2 h_1$. We extend the feature extraction approach to validate inter-feature rules in Tasks B, C, and D. Applying this evaluation to synthetic training data to validate our approach, we show close alignment between estimation and ground truth in Fig.2.

2.3 Experimental Results

For each synthetic task, we generate 2000 samples and report the evaluated results as follows:

DMs' Success on Coarse-Grained Rules. Table 1 demonstrates DMs rarely generate samples that violate coarse-grained rules across all tasks. This observation aligns with expectations: generating samples violate coarse-grained rules requires DMs to generate out of the (training) distribution (OOD) - an extrapolation challenge for DMs observed in prior work (Okawa et al., 2024; Kang et al., 2024). In Task A, for example, all training samples place the sun and shadow on opposite sides of the pole; violating this rule would require generating a never-seen mode with both elements on the same side.

DMs' Failure on Fine-Grained Rules. While following coarse-grained rules only requires DMs to avoid unreasonable OOD generations, fine-grained rules are much harder, demanding accurate learning of the in-distribution training data. Fig.3 demonstrates models' performance across four synthetic tasks, where deviations from the ground truth in linear fitting and the coefficient of determination R^2 below 1 indicate DMs fail to capture the predefined fine-grained rules. Additionally, we observe that DMs struggle more with learning non-spatial rules, such as Task C, compared to spatial rules, such as Task A, as evidenced by worse linear fitting and smaller R^2. This discrepancy likely arises from the fact that non-spatial rules are more implicit and lack explicit cues, such as object positions and lengths, which are readily available in spatial rules. More experiments for various settings (e.g., other backbone models) are deferred to appendix E.3, which shows consistent empirical observations that DMs can capture coarse-grained rules but struggle to master fine-grained ones.



(a) Ideal generations. (b) Memorization .

Figure 5: **DMs generate rule-conforming samples.** Define Rule-conforming generations have ratios (e.g., $\frac{l_2 h_1}{l_1 h_2}$ in Task A) within ± 0.01 of true ratio (1 in Task A). Fig.5(a) shows DDPM's ability to generate rule-conforming samples across tasks. Fig.5(b) indicates that nearest neighbor distances between 10 idel samples in Task A and training data are large (> 0.3), suggesting novel generation rather than memorization.

Despite Instabilities, DMs Can Generate Fine-Grained Samples. While fine-grained rule experiments show DMs generally struggle to exactly satisfy underlying rules, we observe that they can occasionally generate rule-conforming samples in Fig.5(a), albeit with instability. For example, in Task A, there are 10 ideal generated samples that (almost) satisfy the fine-grained rule, i.e., $\frac{l_2 h_1}{l_1 h_2} \in [0.99, 1.01]$. To determine whether these 10 ideal samples originate from DDPM's generation or are merely training data replicas (Somepalli et al., 2023a;b; Wang et al., 2024b), we analyze memorization behaviors. For Task A, we represent each sample with a 13D vector capturing key features (l_1, l_2, h_1, h_2) and encoding RGB colors of sun, pole, and shadow. We then compute Euclidean distances to their nearest neighbors, considering samples as replicas if the distance is below a given threshold. fig. 5(b) shows rule-conforming generations are not mere duplicates, achieving 100% memorization at a large threshold (0.3). Appendix E.2 shows 10 ideal samples and their nearest neighbors, highlighting differences. This suggests DMs can generate rule-conforming samples. Inspired by this, Section 4 presents a mitigation strategy with guidance to improve generation consistency.

3 DMs' Failure from a Theoretical Perspective

This section theoretically explains why DMs struggle with precise rule learning, showing that without prior knowledge, DDPM-trained DMs exhibit constant rule-conformity errors. We consider the following multi-patch data setup, which has been widely employed for theoretical analysis of classification (Allen-Zhu & Li, 2020; Cao et al., 2022; Zou et al., 2023; Lu et al., 2024), and recently for diffusion models (Han et al., 2024a).

Definition 3.1 (Data distribution with Inter-Feature Rules)**.** Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ be two orthogonal feature vectors with unit norm, i.e., $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$ and $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. Let ζ be a random variable with its distribution $\mathcal{D}_\zeta$ supporting on a bounded domain $[\underline{c}_\zeta, \bar{c}_\zeta]$ for some constants $0 < \underline{c}_\zeta < \bar{c}_\zeta < \infty$. Each image data consists of multiple patches

$$\mathbf{x} = [\mathbf{x}^{(1)\top}, \mathbf{x}^{(2)\top}, \cdots, \mathbf{x}^{(P)\top}]^\top, \text{ where } \quad \mathbf{x}^{(1)} = \zeta \mathbf{u},\ \mathbf{x}^{(2)} = (1-\zeta)\mathbf{v},$$

and $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ are *independent* with the remaining patches.

Definition 3.1 specifies a *inter-feature rule* on the first two patches of the data, requiring that the norm of the first two feature patches sum up to one, i.e., $\|\mathbf{x}^{(1)}\| + \|\mathbf{x}^{(2)}\| = 1$. Then, we show such a rule will further lead to a structural constraint on the score function. Specifically, let $\mathbf{x}_0 = [\zeta \mathbf{u}^\top, (1-\zeta)\mathbf{v}^\top, \mathbf{x}^{(3)\top}, \cdots, \mathbf{x}^{(P)\top}]$ represent an input image. For arbitrary noise scedules $\{\alpha_t, \beta_t\}$, $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \beta_t \boldsymbol{\epsilon}_t$ represents the noised image at timestep t. We derive the score function along the diffusion path as follows.

Theorem 3.2. *The score function is* $\nabla \log p_t(\mathbf{x}_t) = [\nabla \log p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})^\top, \nabla \log p_t(\mathbf{x}_t^{(3)}, ..., \mathbf{x}_t^{(P)})^\top]^\top$, *where*

$$\nabla \log p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}) = -\frac{1}{\beta_t^2}\mathbf{x}_t + \frac{\alpha_t}{\beta_t^2}\begin{bmatrix} \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)\zeta]\mathbf{u} \\ \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)(1-\zeta)]\mathbf{v} \end{bmatrix}$$

where $\pi_t(\zeta, \mathbf{x}_t) = \frac{\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d})}{\mathbb{E}_{\mathcal{D}_\zeta}[\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d})]}$, $\boldsymbol{\mu}_t(\zeta) = [\alpha_t \zeta \mathbf{u}^\top, \alpha_t(1-\zeta)\mathbf{v}^\top]^\top$.

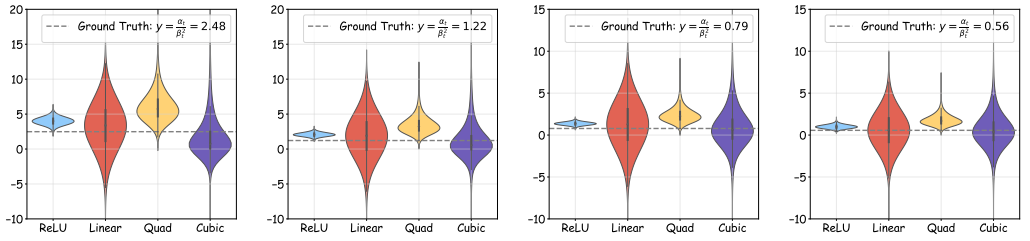
It is clearly noted ground truth score (restricted to first two patches) exhibits the following identity:

$$\mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)\zeta] + \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)(1-\zeta)] = \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)] = 1. \tag{$*$}$$

Then, we aim to investigate whether a score network, trained via DSM objective, can accurately conform to such a hidden rule eq.$*$. Specifically, we follow (Han et al., 2024a) and consider the following two-layer neural network model: $s_w(\mathbf{x}_t) = [s_w^{(1)}(\mathbf{x}_t)^\top, ..., s_w^{(P)}(\mathbf{x}_t)^\top]^\top$, with

$$s_w^{(p)}(\mathbf{x}_t) = -\frac{1}{\beta_t^2}\mathbf{x}_t^{(p)} + \sum_{r=1}^{m} \sigma(\langle \mathbf{w}_{r,t}^{(p)}, \mathbf{x}_t^{(p)} \rangle)\mathbf{w}_{r,t}^{(p)}, \tag{1}$$

where each patch is processed with a separate set of m neurons, and $\sigma(\cdot)$ is an (non-constant) polynomial activation function. Such a network mimics the structure of U-Net (Ronneberger et al., 2015) with shared encoder and decoder weights. The network also contains a residual connection that aligns with the score function (Theorem 3.2). Similar network design has been considered in



(a) $t = 0.2$ (b) $t = 0.4$ (c) $t = 0.6$ (d) $t = 0.8$

Figure 6: **Diffusion model exhibits non-vanishing error on synthetic multi-patch data with norm constraint.** We observe for a variety of timestep t and activation functions, a (two-layer) DM cannot learn precisely the hidden norm constraint as in Definition 3.1, with both bias and variance error.

(Shah et al., 2023; Han et al., 2024a). We train the score network by minimizing the DSM loss (Ho et al., 2020) with expectation on the diffusion noise and the input:

$$L(\mathbf{W}_t) = \mathbb{E}_{\boldsymbol{\epsilon}_t, \mathbf{x}_0} \sum_{p=1}^{P} \left\| s_w^{(p)}(\mathbf{x}_t^{(p)}) - \boldsymbol{\epsilon}_t^{(p)} \right\|^2 \tag{2}$$

where $\mathbf{x}_t^{(p)} = \alpha_t \mathbf{x}_0^{(p)} + \beta_t \boldsymbol{\epsilon}_t^{(p)}$. We next define the *rule-conforming error* to measure the learning outcome of the hidden rule eq.$*$.

Definition 3.3 (Rule-conforming error)**.** For score network s_w of DMs with weights $\mathbf{w}_{r,t}^{(p)*}$, let

$$\psi_t(\mathbf{x}_t) \coloneqq \left\langle s_w^{(1)}(\mathbf{x}_t) + \frac{1}{\beta_t^2}\mathbf{x}_t^{(1)}, \mathbf{u} \right\rangle + \left\langle s_w^{(2)}(\mathbf{x}_t) + \frac{1}{\beta_t^2}\mathbf{x}_t^{(2)}, \mathbf{v} \right\rangle$$

be the coefficient along directions $\mathbf{u}, \mathbf{v}$ at time t for $\mathbf{x}_t$. We say the diffusion model conforms to rule eq.$*$ if $\psi_t(\mathbf{x}_t) = \frac{\alpha_t}{\beta_t^2}$ holds for *any* $\mathbf{x}_t$. We define the *rule-conforming error* as:

$$\mathcal{E} = \mathbb{E}_{\mathbf{x}_t}\left[\left(\psi_t(\mathbf{x}_t) - \frac{\alpha_t}{\beta_t^2}\right)^2\right].$$

Then, we consider training s_w by gradient descent over eq.2 starting from initialization $\{\mathbf{w}_{r,t}^{(p),0}\}_{r\in[m],p\in[P]}$. The following theorem derives a lower bound on the rule-conforming error for the trained score network model.

Theorem 3.4. *Let $\mathbf{w}_{r,t}^{(p)*}$, $r \in [m]$ be a stationary point of the DDPM loss eq.2. Then we can lower bound*

$$\mathcal{E} \geq \mathbb{E}_{\zeta, \boldsymbol{\epsilon}_{t,-}^{(1)}}\left[\mathrm{Var}_{|\zeta, \boldsymbol{\epsilon}_{t,-}^{(1)}}\left(\widetilde{\sigma}^{(1)}(\langle \mathbf{u}, \boldsymbol{\epsilon}_{t,\perp}^{(1)}\rangle)\right)\right] + \mathbb{E}_{\zeta, \boldsymbol{\epsilon}_{t,-}^{(2)}}\left[\mathrm{Var}_{|\zeta, \boldsymbol{\epsilon}_{t,-}^{(2)}}\left(\widetilde{\sigma}^{(2)}(\langle \mathbf{v}, \boldsymbol{\epsilon}_{t,\perp}^{(2)}\rangle)\right)\right]$$

where we decompose $\boldsymbol{\epsilon}_t^{(p)} = \boldsymbol{\epsilon}_{t,-}^{(p)} + \boldsymbol{\epsilon}_{t,\perp}^{(p)}$ with $\boldsymbol{\epsilon}_{t,-}^{(p)}$ being the projection of $\boldsymbol{\epsilon}_t^{(p)}$ onto $\mathrm{span}(\mathbf{w}_{1,t}^{(p),0}, ..., \mathbf{w}_{m,t}^{(p),0})$. $\mathrm{Var}_{(|A)}(\cdot) \coloneqq \mathrm{Var}(\cdot|A)$ *is the conditional variance and $\widetilde{\sigma}^{(p)}(\cdot)$ is a polynomial with coefficients depending on $\langle \mathbf{w}_{r,t}^{(1)*}, \mathbf{u}\rangle, \langle \mathbf{w}_{r,t}^{(2)*}, \mathbf{v}\rangle$.*

Theorem 3.4 immediately suggests a non-vanishing rule-conforming error, as long as the polynomial $\widetilde{\sigma}$ is non-constant and dimension d is sufficiently larger than network width m to ensure variability in the random noise $\boldsymbol{\epsilon}_{t,\perp}$, which is independent of $\mathbf{u}$ and $\mathbf{v}$.

We now show that when simplifying the model to linear activation $\sigma(x) = x$ and single neuron ($\mathbf{w}_t^{(p)}$), the rule-conforming error can be computed as the sum of bias and variance errors, both of them are lower bounded by some constants. Specifically, we decompose

$$\mathcal{E} = \underbrace{\left|\mathbb{E}_{\mathbf{x}_t}\left[\psi_t(\mathbf{x}_t)\right] - \frac{\alpha_t}{\beta_t^2}\right|^2}_{\mathcal{E}_{\text{bias}}^2} + \underbrace{\mathrm{Var}\left[\psi_t(\mathbf{x}_t)\right]}_{\mathcal{E}_{\text{variance}}}.$$

Following theorem suggests there exist a constant bias and variance error for any stationary point $\mathbf{w}_t^*$.

Theorem 3.5. *Suppose $\sigma(x) = x$, $m = 1$ and consider t such that $\alpha_t, \beta_t = \Theta(1)$. We train the network with the gradient descent on DDPM loss eq.2 from small Gaussian initialization, i.e., $\mathbf{w}_t^{(p),0} \sim \mathcal{N}(0, \sigma_0^2 \mathbf{I}_d)$, $\sigma_0 = O(d^{-1/2})$ and $d = \widetilde{\Omega}(1)$. Let $\mathbf{w}_t^{(p)*}$ be any stationary point. Then*

- $\langle \mathbf{w}_t^{(1)*}, \mathbf{u} \rangle, \langle \mathbf{w}_t^{(2)*}, \mathbf{v} \rangle = \Theta(1)$.
- *There exists constants* $C_0, C_1 > 0$ *(depending on* $\mathbb{E}[\zeta], \mathbb{E}[\zeta^2], \alpha_t, \beta_t$*) such that* $\mathcal{E}_{\text{bias}} = C_0, \mathcal{E}_{\text{variance}} = C_1$.

Theorem 3.5 shows that (1) all data features $\mathbf{u}$ and $\mathbf{v}$ can be discovered, which is consistent with the results in Han et al. (2024a) and verifies the ability of DMs to conform to coarse rules in the data, i.e., the existence of the key features. (2) It also verifies that DMs fail to learn the fine-grained hidden rule when no constraint or guidance is imposed over the training of DMs. Both of these two results are consistent with our empirical findings in Section 2.

Empirical verification. We train score networks under the theoretical setup and evaluate rule-conforming error in fig. 6 using four activation functions (see Appendix G for details). The error in learning eq.$*$ and the distribution of $\psi_t(\mathbf{x}_t)$ over 5000 samples show significant rule-conforming errors, confirming our theory and DMs' limitation in learning hidden rules.

4 Mitigation Strategy with Guided Diffusion

Motivated by finding that DMs can produce rule-conforming samples but instability, we mitigate this by a common technique, **Guided DDPM**, which introduces additional classifier guidance (Dhariwal & Nichol, 2021) during sampling. Specifically, we train classifier $f_\theta(\mathbf{x}, t)$ with constructed contrasting data pairs, where positive samples follow fine-grained rules while negative samples violate fine rules while maintaining coarse-grained compliance. The training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda \cdot \mathcal{L}_{\text{contrastive}}, \tag{3}$$

where λ is weight parameter, $\mathcal{L}_{\text{classification}}$ is Cross-Entropy loss and $\mathcal{L}_{\text{contrastive}}$ is NT-Xent loss (Sohn, 2016). More details on NT-Xent loss are in Appendix H.1. Then, following Dhariwal & Nichol (2021), gradients from $f_\theta(\mathbf{x}, t)$ are used to guide sampling toward fine-grained rule compliance.

Additionally, based on constructed contrastive data, we directly train a classifier in raw images to determine whether a generation satisfies fine-grained rules. We filter samples predicted as non-rule-conforming to ensure generation quality. This approach, called **Filtered DDPM**, provides guidance based on the noise-free pixel space, can be seen as upper bound for guided diffusion strategies.

4.1 Experiment Results

Setup. Details of data construction and training process are provided in Appendix H.1.

Results. In addition to R^2, inspired by the theorical analysis in section 3, we introduce Error, a metric capturing how well DMs learn hidden rules from variance and bias. Given the Ground Truth line $y = \beta_1 x$ and the Estimation line $\hat{y} = \hat{\beta}_1 x + \hat{\beta}_0$ in Fig.2 and 3, Error is defined as:

$$\text{Error} := \underbrace{|\hat{\beta}_1 - \beta_1| + |\hat{\beta}_0|}_{\text{Bias Error}} + \underbrace{\sqrt{\text{Var}(\hat{y} - y)}}_{\text{Variance Error}} \tag{4}$$

We measure the bias error $|\mathbb{E}[y - \hat{y}]|$ with the deviation in the estimated coefficients $\hat{\beta}_1, \hat{\beta}_0$. The variance error in eq.4 corresponds to the square root of $\mathcal{E}_{\text{variance}}$ in Section 3. Table 2 presents results, Error and R^2, before (DDPM) and after applying classifier guidance (Guided DDPM), along with DDPM filtered by pixel-space classifier (Filtered DDPM). Both Guided DDPM and Filtered DDPM outperform the baseline DDPM across all tasks, showing reduced Error and improved R^2, with Filtered DDPM achieving the best performance on most tasks.

4.2 Discussions on the Limitation of Guided Diffusion

While guided and filtered diffusion helps with rule learning, the improvement is limited. Unlike conventional classification tasks, fine-grained rules in contrastive samples have subtle signals, making classifier training difficult. Appendix H.1 provides evidence, showing test accuracy remains between 60% and 80% even on simple tasks. Additionally, the effectiveness of this strategy relies on prior knowledge of fine-grained rules. In real-world scenarios, fine-grained rules are often difficult to accurately define and detect, making the construction of contrastive data impossible. We leave the solution to DMs' inability to learn fine-grained rules in real-world scenarios for future work.

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.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Anonymous. Towards understanding text hallucination of diffusion models via local generation bias. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SKW10XJlAI.

Joe Benton, VD Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. 2024.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

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

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. *Advances in Neural Information Processing Systems*, 35:25237–25250, 2022.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pp. 4735–4763. PMLR, 2023.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

Yinbo Chen, Oliver Wang, Richard Zhang, Eli Shechtman, Xiaolong Wang, and Michael Gharbi. Image neural field diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8007–8017, 2024.

Justin Deschenaux, Igor Krawczuk, Grigorios Chrysos, and Volkan Cevher. Going beyond compositions, ddpms can produce zero-shot interpolations. In *Forty-first International Conference on Machine Learning*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022.

Huan Fu and Guoqing Cheng. Enhancing semantic mapping in text-to-image diffusion via gather-and-bind. *Computers & Graphics*, 125:104118, 2024.

Tejas Gokhale, Hamid Palangi, Besmira Nushi, Vibhav Vineet, Eric Horvitz, Ece Kamar, Chitta Baral, and Yezhou Yang. Benchmarking spatial relationships in text-to-image generation. *arXiv preprint arXiv:2212.10015*, 2022.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Andi Han, Wei Huang, Yuan Cao, and Difan Zou. On the feature learning in diffusion models. *arXiv preprint arXiv:2412.01021*, 2024a.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis, 2024b. URL https://arxiv.org/abs/2412.04431.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Xin Huang, Ruizhi Shao, Qi Zhang, Hongwen Zhang, Ying Feng, Yebin Liu, and Qing Wang. Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4568–4577, 2024.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2023.

Jean-Marie Lemercier, Julius Richter, Simon Welker, Eloi Moliner, Vesa Välimäki, and Timo Gerkmann. Diffusion models for audio restoration. *arXiv preprint arXiv:2402.09821*, 2024.

Hao Li, Yang Zou, Ying Wang, Orchid Majumder, Yusheng Xie, R Manmatha, Ashwin Swaminathan, Zhuowen Tu, Stefano Ermon, and Stefano Soatto. On the scalability of diffusion-based text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9400–9409, 2024a.

Xinghui Li, Jingyi Lu, Kai Han, and Victor Adrian Prisacariu. Sd4match: Learning to prompt stable diffusion model for semantic matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27558–27568, 2024b.

Youngsun Lim and Hyunjung Shim. Addressing image hallucination in text-to-image generation through factual image retrieval. *arXiv preprint arXiv:2407.10683*, 2024.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023a.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Qihao Liu, Adam Kortylewski, Yutong Bai, Song Bai, and Alan Yuille. Discovering failure modes of text-guided diffusion models via adversarial search. *arXiv preprint arXiv:2306.00974*, 2023b.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Miao Lu, Beining Wu, Xiaodong Yang, and Difan Zou. Benign oscillation of stochastic gradient descent with large learning rate. In *The Twelfth International Conference on Learning Representations*, 2024.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.

Shweta Mahajan, Tanzila Rahman, Kwang Moo Yi, and Leonid Sigal. Prompting hard or hardly prompting: Prompt inversion for text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6808–6817, 2024.

Arash Marioriyad, Parham Rezaei, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. Diffusion beats autoregressive: An evaluation of compositional generation in text-to-image models. *arXiv preprint arXiv:2410.22775*, 2024.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Axi Niu, Trung X Pham, Kang Zhang, Jinqiu Sun, Yu Zhu, Qingsen Yan, In So Kweon, and Yanning Zhang. Acdmsr: Accelerated conditional diffusion models for single image super-resolution. *IEEE Transactions on Broadcasting*, 2024.

Maya Okawa, Ekdeep S Lubana, Robert Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. *Advances in Neural Information Processing Systems*, 36, 2024.

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.

Anton Pelykh, Ozge Mercanoglu Sincan, and Richard Bowden. Giving a hand to diffusion models: a two-stage approach to improving conditional human image generation. *arXiv preprint arXiv:2403.10731*, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

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.

Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the ddpm objective. *Advances in Neural Information Processing Systems*, 36:19636–19649, 2023.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

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*, pp. 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023b.

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.

Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. Can diffusion model achieve better performance in text generation? bridging the gap between training and inference! *arXiv preprint arXiv:2305.04465*, 2023.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024a. URL https://arxiv.org/abs/2404.02905.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024b.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

Binxu Wang, Jiaqi Shang, and Haim Sompolinsky. Do diffusion models generalize on abstract rules for reasoning? 2024a.

Hanyu Wang, Yujin Han, and Difan Zou. On the discrepancy and connection between memorization and generation in diffusion models. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024b.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36, 2024.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

Muqiao Yang, Chunlei Zhang, Yong Xu, Zhongweiyang Xu, Heming Wang, Bhiksha Raj, and Dong Yu. Usee: Unified speech enhancement and editing with conditional diffusion models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7125–7129. IEEE, 2024.

Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

Yufei Ye, Poorvi Hebbar, Abhinav Gupta, and Shubham Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19717–19728, 2023.

Sihyun Yu, Weili Nie, De-An Huang, Boyi Li, Jinwoo Shin, and Anima Anandkumar. Efficient video diffusion models via content-frame motion-latent decomposition. *arXiv preprint arXiv:2403.14148*, 2024.

Hangjie Yuan, Shiwei Zhang, Xiang Wang, Yujie Wei, Tao Feng, Yining Pan, Yingya Zhang, Ziwei Liu, Samuel Albanie, and Dong Ni. Instructvideo: instructing video diffusion models with human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6463–6474, 2024.

Yiming Zhang, Zhe Wang, Xinjie Li, Yunchen Yuan, Chengsong Zhang, Xiao Sun, Zhihang Zhong, and Jian Wang. Diffbody: Human body restoration by imagining with generative diffusion prior. *arXiv preprint arXiv:2404.03642*, 2024.

Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. The benefits of mixup for feature learning. In *International Conference on Machine Learning*, pp. 43423–43479. PMLR, 2023.

A LOW FID AND WORSE INTER-FEATURE LEARNING: A GAUSSIAN MIXTURE CASE

In this section, we provide a toy example based on the Gaussian Mixture Distribution to explain how low FID and incorrect inter-feature relationships can coexist. This supports the point that even though DMs may perform excellently on classical metrics such as FID, this does not necessarily mean they can perfectly learn the hidden inter-feature rules.

Consider a 2-dimensional population, i.e., the true distribution $p(x, y)$, which is a Gaussian Mixture Model (GMM) with two components as:

$$p(x,y) = \mathcal{F}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p) = \frac{1}{2} \cdot \mathcal{N}\left(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) + \frac{1}{2} \cdot \mathcal{N}\left(\begin{bmatrix}-1\\-1\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right). \tag{5}$$

where we can have

$$\boldsymbol{\mu}_p = \frac{1}{2} \cdot \begin{bmatrix}1\\1\end{bmatrix} + \frac{1}{2} \cdot \begin{bmatrix}-1\\-1\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

and the covaraince matirx as

$$\begin{aligned}\boldsymbol{\Sigma}_p &= \sum_{i=1}^{2} w_i \left(\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_q)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_q)^\top\right) \\ &= 0.5 \cdot \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} + \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\right) + 0.5 \cdot \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} + \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\right) = \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}.\end{aligned}$$

We assume the estimated data distribution learned by DMs is a joint Gaussian distribution:

$$q(\hat{x}, \hat{y}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}\right). \tag{6}$$

With means and covariance matrices of true distributon p and estimated distribution q are identical, that is $\boldsymbol{\mu}_p = \boldsymbol{\mu}_q = [0, 0]^\top$ and $\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_q = \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}$, we easily have the FID between $p(x, y)$ and $p(\hat{x}, \hat{y})$ is computed as:

$$\begin{aligned}\text{FID} &= \|\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\|_2^2 + \text{Tr}\left(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q - 2\left(\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_q\right)^{1/2}\right) \\ &= \|0 - 0\|_2^2 + \text{Tr}\left(\begin{bmatrix}2 & 1\\1 & 2\end{bmatrix} + \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix} - 2\begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}\right) \\ &= 0\end{aligned} \tag{7}$$

Although the FID is small (i.e., 0), the inter-feature relationships between x and y in true and estimated distribution are fundamentally different. In the true distribution, x and y are independent within each Gaussian component but exhibit dependence in the overall distribution due to the mixture of components. In the estimated distribution $q(\hat{x}, \hat{y})$, $\hat{x}$ and $\hat{y}$ are dependent with $\text{Cov}(x, y) = 1$. Therefore, low FID does not imply a correct recovery of the inter-feature rules.

B RELATED WORK

We summarize prior studies on the ability of DMs to learn specific rules, and discuss the relations to inter-feature rules.

Factual Knowledge Rules. The violation of factual rules in DMs refers to generated images failing to accurately reflect factual information and common sense, often characterized as hallucinations in existing work (Aithal et al., 2024; Lim & Shim, 2024; Anonymous, 2025). Typical examples include violating common sense, such as extra, missing, or distorted fingers (Aithal et al., 2024; Pelykh et al., 2024; Ye et al., 2023), unreadable text (Gong et al., 2022; Tang et al., 2023; Xu et al., 2024) and snowy deserts (Lim & Shim, 2024). Additionally, inconsistencies between generated images and given textual prompts (Liu et al., 2023b; Fu & Cheng, 2024; Mahajan et al., 2024; Li et al., 2024b) can be regarded as violations of prompt-based knowledge. Unlike inter-feature rules,

Table 3: **Real-World Inter-Feature Rules.** For each scenario containing inter-feature rules, table 3 provides detailed prompts and annotates the existing inter-feature relationships. By comparing the genuine inter-feature relationships with those in generated images, we can evaluate DMs' ability to learn inter-feature relationships.

[Spatial Rule] (a) Light shadow: Prompt 1: A desert scene with a majestic palace under a bright sun. Inter-Feature Rule 1: `Sun position affects palace's shadow direction.` Prompt 2: `The moonlight casts a clear shadow of a tall tree onto the ground.` Inter-Feature Rule 2: Moon position affects tree's shadow direction
[Spatial Rule] (b) Reflection/Refraction Prompt 1: `A plush lion toy in front of the mirror. Its front side is facing the camera. There is its reflection in the mirror.` Inter-Feature Rule 1: The lion toy's orientation relative to the mirror determines its reflection's orientation. Prompt 2: `A transparent glass bottle partially submerged in a calm, clear pool of water. The upper half of the bottle extends above the water's surface and the lower half of the bottle is submerged.` Inter-Feature Rule 2: The water surface's position dictates the bottle's shape distortion.
[Spatial Rule] (c) Semantics Prompt 1: `A field of sunflowers under a clear blue sky with the sun shining brightly above.` Inter-Feature Rule 1: Sun direction dictates sunflower orientation. Prompt 2: `A paintbrush fully loaded with paint, making a stroke on a blank white canvas.` Inter-Feature Rule 2: Brush tip color matches canvas paint.
[Non-Spatial Rule] (d) Size -Texture Prompt 1: `The cross-section of a sturdy tree, covered with annual rings.` Inter-Feature Rule 1: The diameter of a tree is related to its growth rings. Prompt 2: `A nautilus fossil, showing its intricate spiral shell structure with visible growth chambers.` Inter-Feature Rule 2: Nautilus fossil size correlates with spiral patterns.
[Non-Spatial Rule] (e) Size/Region- Color Prompt 1: `An artistic representation showing the expanded star phase and cooling star phase of the same star.` Inter-Feature Rule 1: Celestial body size and color should align, exemplified by red giants and white dwarfs. Prompt 2: `A burning red candle in a dark with the flame, which is vibrant, dynamic, and glowing intensely against the darkness.` Inter-Feature Rule 2: Candle flame color varies with distance from the wick.
[Non-Spatial Rule] (f) Color - Color Prompt 1: `Two Eclectus parrots, showcasing the striking sexual dimorphism of the species.` Inter-Feature Rule 1: Eclectus parrots' body and beak colors match—green and yellow for males, red and black for females. Prompt 2: `A male Poecilia reticulata and a female Poecilia reticulata are swimming gracefully in a clear, freshwater aquarium, showcasing the striking sexual dimorphism of the species` Inter-Feature Rule 2: Guppies' body and tail colors match—males are equally colorful in both.

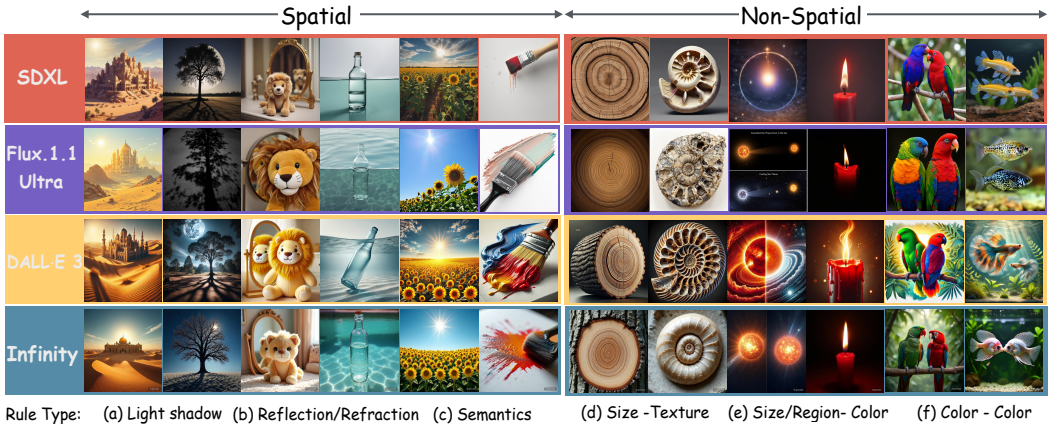


Figure 7: **Evaluating More Mainstream DMs on Real-World Inter-Feature Rules.** We evaluate more mainstream DMs on scenarios with inter-feature rules, with 5 random generations and manual selection of unreasonable samples. Despite their success in metrics like FID, none of these DMs achieve complete correctness in cases involving inter-feature relationships.

factual knowledge rules *do not involve relationships between multiple features* and are typically attributed to imbalanced training data distribution (Samuel et al., 2024) or mode interpolation caused by inappropriate smoothing of training data (Aithal et al., 2024).

Independent Features Rules. Prior work has investigated DMs' ability to combine independent features, i.e., compositionality. Through controlled studies with independent concepts (e.g., color, shape, size), Okawa et al. (2024) observe that DDPM can successfully compose different independent features. Similar findings are reported in (Deschenaux et al., 2024), where interpolation between portraits without and with clear smiles resulted in generations with mild smiles. However, numerous studies highlight DMs' limitations in complex compositional tasks (Liu et al., 2022; Gokhale et al., 2022; Feng et al., 2022; Marioriyad et al., 2024), potentially due to insufficient training data for reconstructing each individual feature (Wiedemer et al., 2024). These studies primarily examine compositional tasks with *independent features*, in contrast to our focus on feature dependencies.

Abstract (Dependent Feature) Rules. This type closely aligns with our work, which studies feature relationships like shape consistency in generations. Prior studies give mixed conclusions on DDPM's rule-learning ability. For example, DDPM struggles with numerical addition rule (Anonymous, 2025) but maintains shape consistency rule in RAVEN task (Wang et al., 2024a). Inconsistent rule complexity leads to ambiguous evaluation conclusions, and the lack of theoretical analysis leaves the underlying factors behind DMs' performance in rule learning poorly understood. Through controlled experiments with adjustable rule complexity, we provide a unified assessment of DMs' rule-learning abilities and offer a theoretical explanation of their fundamental limitations, as a result of their training paradigm.

C Details and More Example on Real-Wold Hidden Inter-Feature Rules

table 3 provides a detailed description of the prompts for each case in fig. 1 and fig. 7, including scenarios with inter-feature rules and the corresponding rules themselves. We also consider more DMs such as `SDXL`[4] (Podell et al., 2023), `Flux.1.1 Ultra`[5] (Labs, 2023), `DALL·E 3`[6] (Betker et al., 2023), and VAR-based (Tian et al., 2024a) text-to-image model `Infinity`[7] (Han et al., 2024b) in the evaluation. By comparing these rules, we observe that most mainstream DMs fail in some or all scenarios. For instance, in the *Reflection/Refraction* scenario, none of the DMs successfully generate

[4]https://fal.ai/models/fal-ai/fast-lightning-sdxl
[5]https://fal.ai/models/fal-ai/flux-pro/v1.1-ultra
[6]https://chatgpt.com/g/g-iLoR8U3iA-dall-e3
[7]https://github.com/FoundationVision/Infinity?tab=readme-ov-file

plausible images: the reflected toy in the mirror faces the camera just like the real one, and the submerged bottle shows no refraction. Our evaluation covers both classic latent diffusion models (e.g., `SD-3.5 Large`) and the latest next-scale prediction-based diffusion models (e.g., `Infinity`). Surprisingly, none of them can perfectly handle these inter-feature relationships, highlighting the widespread limitation of DMs in this regard.

D Details and More Example on Synthetic Tasks

This section presents supplementary details and examples regarding our synthetic datasets.

Task A generates synthetic images featuring a simple outdoor scene composed of a vertical pole, a sun, and their corresponding shadow. The height of the pole is randomly selected within the range of $[6.4, 12.8]$ pixels, which corresponds to $[20\%, 40\%]$ of the total image size (32×32 pixels). The sun's horizontal position is sampled from two predefined distance intervals: far distances ($0-6$ pixels or $26-32$ pixels) and near distances ($10-16$ pixels or $16-22$ pixels), ensuring a varied distribution of sun locations. The shadow length is computed using the formula:

$$\text{shadow\_length} = \frac{\text{pole\_height} \times |\text{sun\_distance}|}{\text{sun\_height} - \text{pole\_height}} \tag{8}$$

where the sun height is determined as twice the pole height, clipped within $[9.6, 25.6]$ pixels (30%-80% of the image size). Colors for the sun, pole, and shadow are randomly selected from predefined HSV (Hue-Saturation-Value) ranges: Sun color (yellowish tones) has a hue in $[0, 30]$, saturation in $[100, 255]$, and value in $[200, 255]$. Pole color (blue-green tones) has a hue in $[90, 150]$, saturation in $[100, 255]$, and value in $[100, 255]$. Shadow color (dark tones like black, brown, gray) has a hue in $[0, 180]$, saturation in $[0, 50]$, and value in $[50, 150]$.

Task B generates synthetic images containing two rectangular objects placed within a 32×32 pixel space. The first rectangle's position and size are determined as follows: its leftmost position l_1 is chosen randomly from the range $[0, 9.6]$ pixels (30% of the image width), and its height l_2 is chosen randomly from $[6.4, 19.2]$ pixels (20% to 60% of the image height). The color of the first rectangle is randomly selected from a yellowish hue range with hue $[0, 30]$, saturation $[100, 255]$, and value $[200, 255]$ in HSV space. The second rectangle's position is determined by h_1, which is chosen randomly within a range dependent on l_1. Specifically, h_1 is sampled from the range $[l_1 + 6.4, 25.6]$ pixels (ensuring $h_1 > l_1$). The height of the second rectangle h_2 is calculated based on the first rectangle's height l_2, ensuring the relation $l_1 h_1 = h_2 l_2$. The color of the second rectangle is chosen randomly from a blue-green hue range with hue $[90, 150]$, saturation $[100, 255]$, and value $[100, 255]$ in HSV space.

Task C generates images containing two circles: one large and one small. The large circle's diameter is randomly chosen between 10% and 30% of the image size, and the small circle's diameter is determined to be $\sqrt{2}$ times the diameter of the large circle. The colors of the circles are randomly selected from predefined color ranges in the HSV color space. Specifically, the large circle is assigned a color from the blue-green hue range, with hue values between 90 and 150, saturation between 100 and 255, and brightness between 100 and 255. The small circle is assigned a color from the yellowish hue range, with hue values between 0 and 30, saturation between 100 and 255, and brightness between 200 and 255. The circles are randomly positioned such that they are adjacent to each other—either on the left, right, top, or bottom of the large circle.

Task D generates images containing two squares: one smaller and one larger. The small square's size is randomly chosen to be between 30% and 70% of the top half of the image's size. The larger square's size is then set to be 1.5 times the size of the small square. The color of the small square is randomly selected from a yellowish hue range, with hue values between 0 and 30, saturation between 100 and 255, and brightness between 200 and 255. The color of the large square is randomly chosen from a blue-green hue range, with hue values between 90 and 150, saturation between 100 and 255, and brightness between 100 and 255. The position of the squares is determined within specific regions of the image. The top half and the bottom half of the image are divided into distinct regions, with the small square being placed in the top half and the large square in the bottom half. The exact position of each square is randomly chosen within its respective region, while ensuring that the squares do not exceed the image's boundaries. Both squares are positioned such that they do not overlap with each other and remain entirely within the image frame.

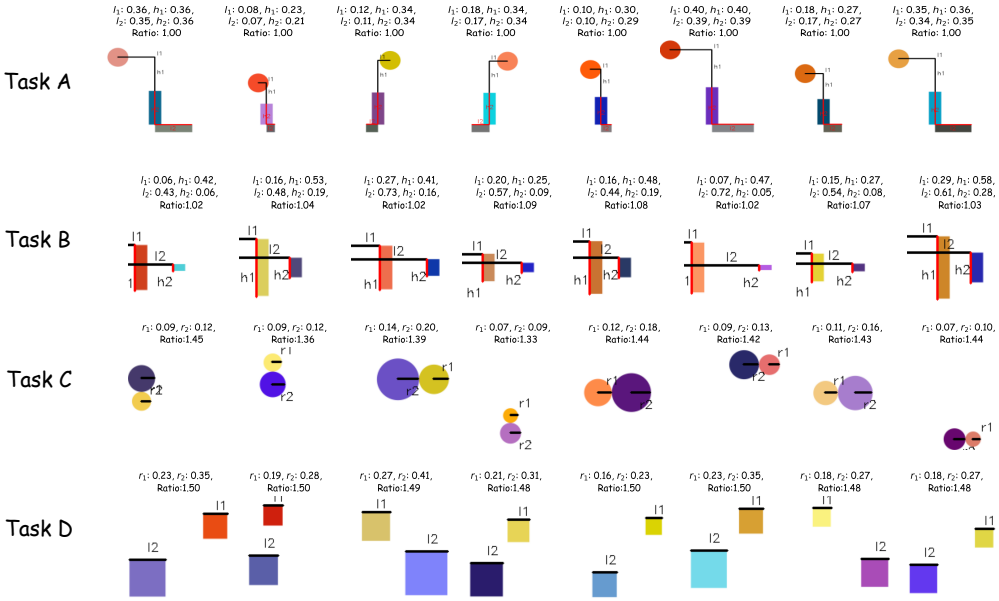


Figure 8: **Synthetic Data Examples.** We present synthetic samples in four synthetic tasks, with annotations of features of interest and Ratio calculations. The target Ratios for Tasks A, B, C, and D are 1, 1, $\sqrt{2}$, and 1.5, respectively.

E More Synthetic Tasks Setup and Results

E.1 More Details of Experimental Setup

4000, 2000, 2000, and 2000 samples are generated for synthetic task A, B, C and D, respectively, with an image size of 32×32. Additionally, in appendix E.3, we explore more advanced architectures such as DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024), alongside larger synthetic datasets of 20000 and 40000 samples and higher image resolutions of 64×64. These factors enhance the training of DMs, thus leading to better alignment between generated and real data distributions (Chen et al., 2022; Benton et al., 2024; Chen et al., 2023) and enabling more effective learning of hidden rules. We use the U-Net architecture as the denoising network, consisting of several down-sampling and up-sampling blocks, each with two convolutional layers followed by ReLU activation. Each down-sampling block incorporates a Self-Attention mechanism and skip connections to preserve fine details. Pooling layers are used to reduce spatial dimensions and capture abstract features. A final 1×1 convolution layer produces the denoised output image. We use AdamW (Loshchilov, 2017) as the optimizer with a learning rate of $3e-4$. The noisy steps are set to $T = 1000$, with a linear noise schedule ranging from $1e-4$ to $2e-2$. For Tasks A, B, C, and D, the sample sizes are 4000, 2000, 2000, and 2000, respectively, and the input data size is $(3, 32, 32)$. The training is performed on a single NVIDIA A800 GPU for 400, 800, 1600, and 1000 epochs, respectively.

E.2 More Results of Synthetic Tasks

This section provides additional details to complement the experimental results in section 2.3. Notably, to ensure more accurate quality assessment of generated images, we upscale the 32×32 images to 128×128 during evaluation. This allows the training data to precisely exhibit the expected rule patterns, thereby enabling more reliable evaluation of the generated samples.

Generations that Violate Coarse-Grained Rules. table 1 illustrates the DDPM's ability to learn coarse-grained rules. We observe that in all four synthetic tasks, the number of samples violating the coarse-grained rules is almost zero, except for Task A, where one generated sample, shown in fig. 10, has the sun and shadow on the same side of the pole.

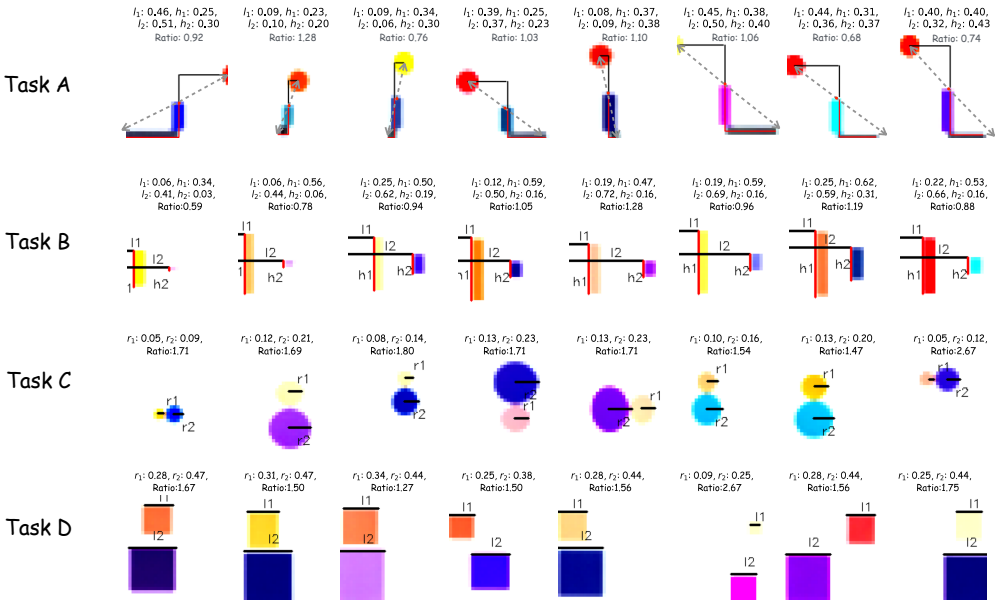


Figure 9: **Examples of Rule-violating Generations.** We present samples generated by DDPM that violate fine-grained rules in four synthetic tasks, with annotations of features of interest and Ratio calculations. The target Ratios for Tasks A, B, C, and D are 1, 1, $\sqrt{2}$, and 1.5, respectively.

Generations that Violate Fine-Grained Rules. We then proceed to show the samples generated by DDPMs that do not satisfy the fine rules in fig. 9, and highlight the features of interest using the evaluation method developed in section 2.2.

Generations that Satisfy Fine-Grained Rules. Here, we use two coordinate systems: a 4D representation capturing key features (l_1, l_2, h_1, h_2) and a 13D representation that additionally encodes the RGB colors of the sun, pole, and shadow. This dual-coordinate analysis allows us to distinguish whether differences between generated and training samples arise from structural variations or merely from different color combinations within similar structures (Okawa et al., 2024). We then compute the Euclidean distances between each generated sample and its nearest neighbor in both 4D and 13D spaces. As a supplement to the DDPM memory experiment in section 2.3, fig. 11 presents the three nearest neighbors in the training data for high-quality generated samples (with ratios in $[0.99, 1.01]$) in both 4-dimensional and 13-dimensional coordinates. We observe that the 4-dimensional coordinates effectively capture the spatial structure of the nearest neighbors in the training data, while the 13-dimensional coordinates provide a more comprehensive understanding of the similarity of the generated samples, accounting for both color and structure.

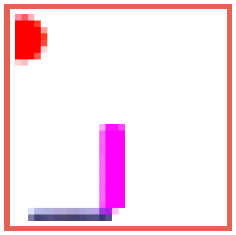


Figure 10: For Task A, while all training samples have the sun and shadow on opposite sides, DDPM generates one sample violating this coarse-grained rule where the sun and shadow appear on the same side.

E.3 More Setting of Synthetic Tasks

In this section, we consider additional factors, such as more powerful model architectures and larger training datasets, to evaluate the diffusion model's ability to learn precise rules in Task A. Furthermore, detailed experimental results not included in section 2.3, such as samples that violate coarse rules, will be presented in this section.

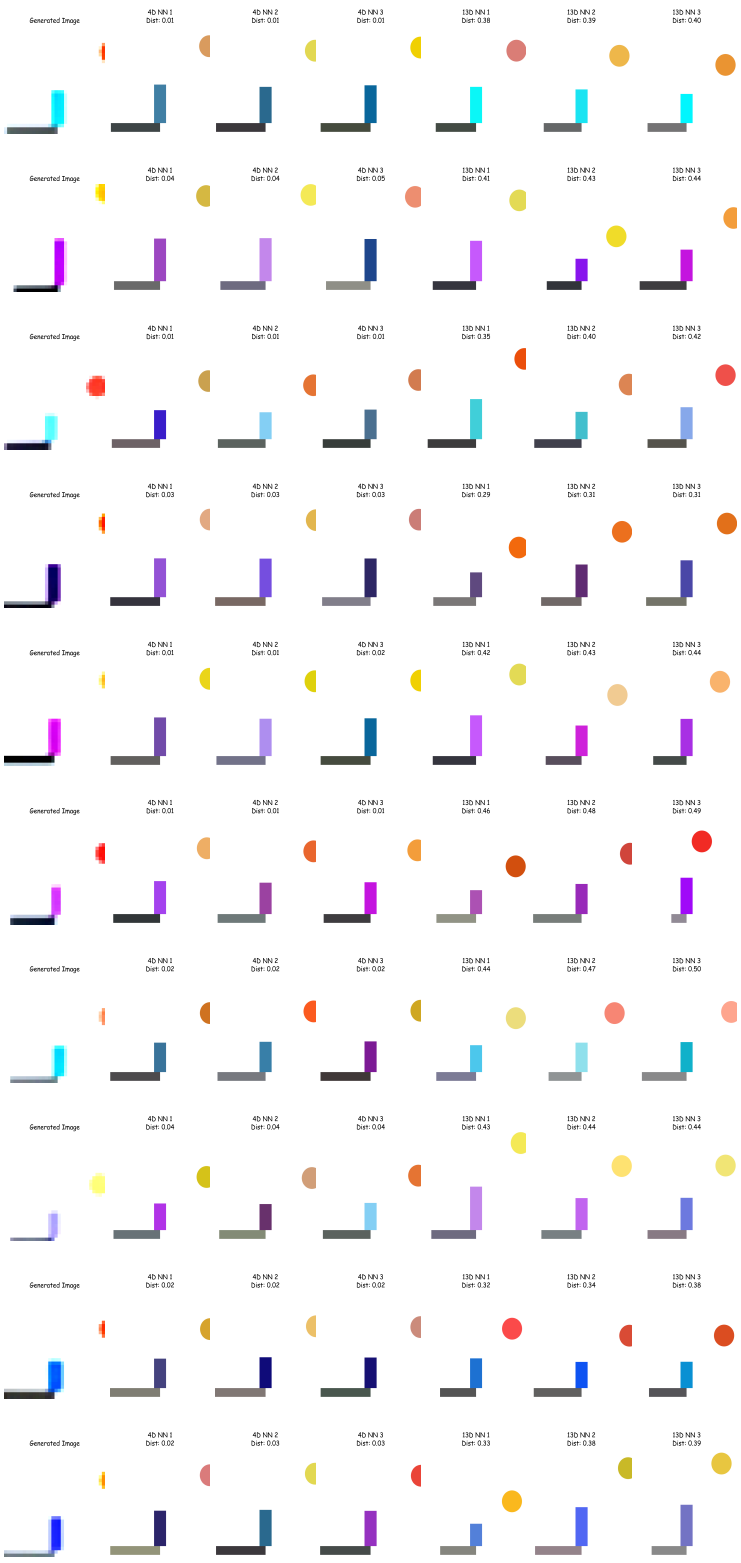


Figure 11: **Generations that Violate Fine-Grained Rules.** Taking Task A as an example, we show 10 high-quality generated samples and their Top-3 nearest neighbors from the training data. The first column visualizes the generated samples, while columns 2-4 display the Top-3 nearest neighbors from training data in 4D coordinates, where similarity mainly reflects spatial structure. Columns 5-7 show the top-3 nearest neighbors in 13D coordinates, where similarity primarily reflects object colors.

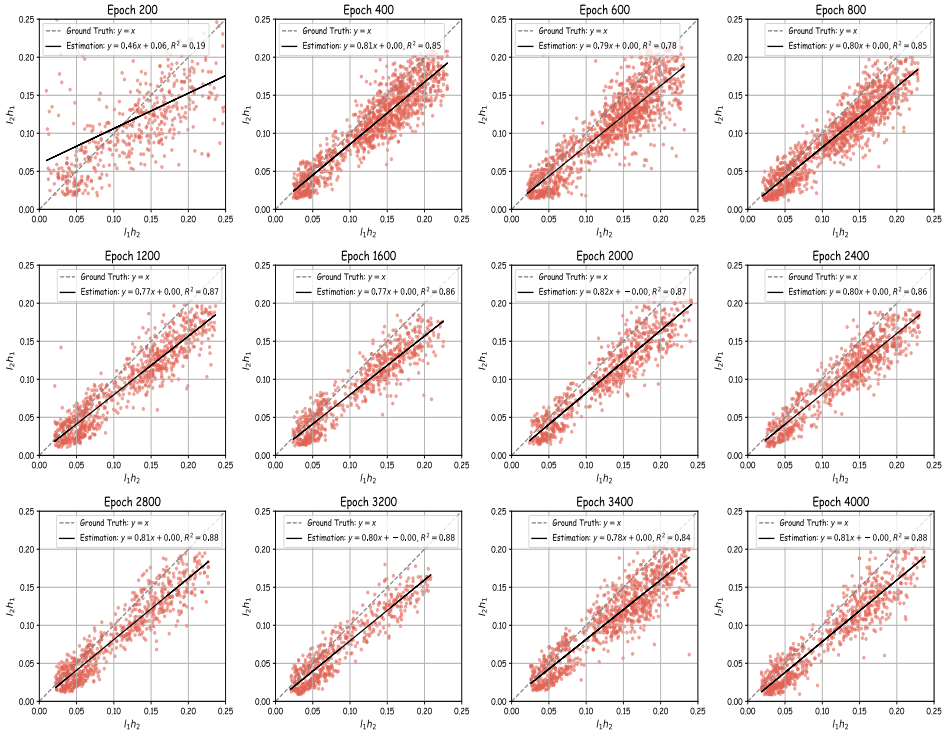
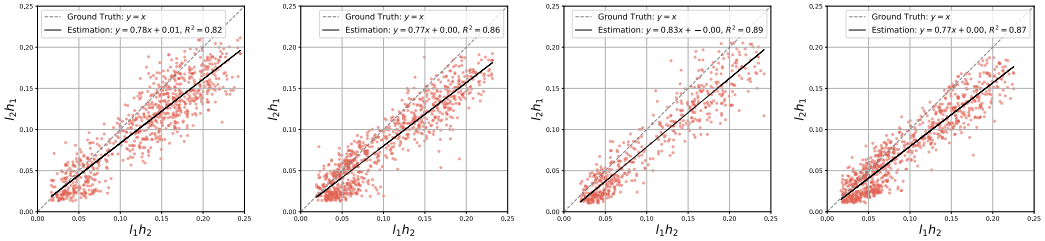


Figure 12: **The learning capability of DDPM for fine-grained rules across training epochs.** We observe that as epochs increase from 200 to 4000, DDPM's ability to learn fine-grained rules shows no significant improvement, as evidenced by the stable Estimation line and R^2. This suggests that increasing training iterations does not alleviate DMs' difficulty in learning fine-grained rules. The visualized generated samples fall within the interval $[2.5\%, 97.5\%]$.

More Training Epochs. Taking Task A as an example, fig. 12 shows the impact of more training epochs on learning fine-grained rules. We observe that as the number of training epochs increases, the DDPM's ability to learn fine-grained rules improves significantly from 200 to 400 epochs, with R^2 increasing from 0.19 to 0.85. This indicates that the relationship between l_1h_2 and l_2h_1 is better described by the linear model. However, even as the training continues up to 4000 epochs, there is no noticeable improvement in the model's ability to learn the fine-grained rules, as reflected by the slight changes in the fitted line coefficients and R^2 remaining around 0.85.

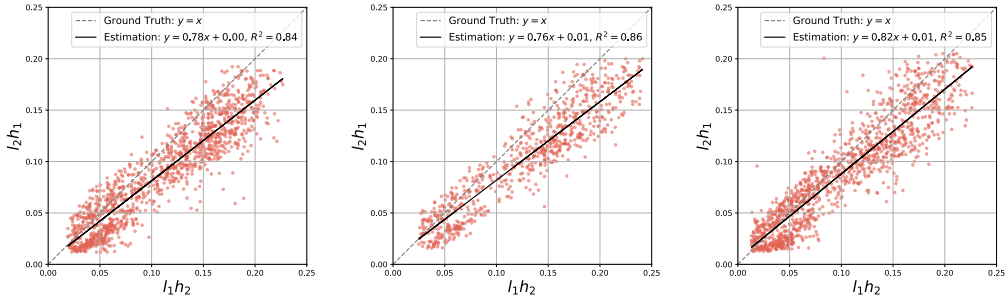
More Model Architectures. Then, we consider the factor modle architectures and use more powerful backbones, DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024), to replace U-Net as the denoising network. Specifically, we consider two sizes of DiT and SiT: DiT Small with 33M parameters and patch size (DiT-S/2), DiT Base with 130M parameters and patch size (DiT-B/2), SiT Small with 33M parameters and patch size (SiT-S/2), and SiT Base with 130M parameters and patch size (SiT-B/2). Keeping the number of training epochs, noise time steps, and other hyperparameters consistent, we find that, compared to the 14M parameter U-Net, the parameter count of SiT and DiT has increased by 2 to 10 times. However, as revealed in fig. 13, although all models follow coarse rules, the deficiency in DDPM's ability to learn fine-grained rules does not significantly improve with the increase in parameter count, and there is even a slight decrease in performance with DiT-S/2.

More Training Data. Next, we consider the impact of training data size. For Task A, we gradually increase the sample size from 4000 to 20000 to 40000 and observe whether increasing the sample size improves the DMs' ability to learn rules. fig. 14(a) and fig. 14(b) show that the increase in sample size does not enable DMs to learn fine-grained rules better, as evidenced by the almost unchanged R^2 and the fitted linear model. Similarly, we do not observe DMs violating coarse rules with large samples.



(a) 33M, DiT-S/2 (b) 130M, DiT-B/2 (c) 33M, SiT-S/2 (d) 130M, DiT-B/2

Figure 13: **DDPM's capability in learning fine-grained rules with more powerful backbones.** Even with larger and more advanced denoising networks, DDPM still cannot avoid generating samples that violate fine-grained rules. This indicates that DDPM's inability to learn fine-grained rules is decoupled from model architecture. The visualized generated samples fall within the interval $[2.5\%, 97.5\%]$.



(a) Training Data Size 20000 (b) Training Data Size 40000 (c) Image Size 64×64

Figure 14: **DDPM's capability in learning fine-grained rules with increased training samples and larger image sizes.** We observe that increasing training samples and image sizes does not significantly improve DDPM's ability to learn fine-grained rules, as evidenced by the stable Estimation line and R^2. This suggests that neither expanding the training dataset nor increasing image resolution alleviates DMs' difficulty in learning fine-grained rules. The visualized generated samples fall within the interval $[2.5\%, 97.5\%]$.

More Image Size Choice. Our final consideration is image size. In the main text, the images are only 32×32. Existing studies suggest that low-resolution images may lead to the loss of details in diffusion models' generation (Chen et al., 2024; Niu et al., 2024; Li et al., 2024a). Therefore, we consider larger input resolutions of $(3, 64, 64)$, as shown in fig. 14(c). We observe almost no improvement in the DMs' ability to learn underlying rules with generated samples that do not violate coarse rules. Due to computational constraints, we were unable to explore even higher resolutions. But it is clear that for a relatively simple task like Task A, which does not contain rich semantics, DMs are unable to recover the underlying feature relationships even at the 64×64 resolution. This itself highlights the difficulty DMs face in learning hidden features.

F PROOFS

Proof of Theorem 3.2. Given the independence, we can write $p_t(\mathbf{x}_t) = p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}) p_t(\mathbf{x}_t^{(3)}, ..., \mathbf{x}_t^{(P)})$. We derive $p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})$ as follows. First, we notice that $\mathbf{x}_t^{(1)}|\zeta \sim \mathcal{N}(\alpha_t \zeta \mathbf{u}, \beta_t^2 \mathbf{I})$ and $\mathbf{x}_t^{(2)}|\zeta \sim \mathcal{N}(\alpha_t(1-\zeta)\mathbf{v}, \beta_t^2 \mathbf{I})$. Then we obtain

$$p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}) = \mathbb{E}_{\mathcal{D}_\zeta}[\mathcal{N}(\boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d})]$$

where we denote $\boldsymbol{\mu}_t(\zeta) = [\alpha_t \zeta \mathbf{u}^\top, \alpha_t(1-\zeta)\mathbf{v}^\top]^\top$.

Thus the score can be computed as $\nabla \log p_t(\mathbf{x}_t) = [\nabla \log p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})^\top, \nabla \log p_t(\mathbf{x}_t^{(3)}, ..., \mathbf{x}_t^{(P)})^\top]^\top$ where $\nabla \log p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})^\top \in \mathbb{R}^{2d}$ and can be derived as

$$\begin{aligned}
\nabla \log p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}) = \frac{\nabla p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})}{p_t(\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})} &= \frac{\mathbb{E}_{\mathcal{D}_\zeta}[\nabla \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d})]}{\mathbb{E}_{\mathcal{D}_\zeta}[\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d})]} \\
&= \frac{\mathbb{E}_{\mathcal{D}_\zeta}\big[-\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d}) \beta_t^{-2} \big(\mathbf{x}_t - \boldsymbol{\mu}_t(\zeta)\big)\big]}{\mathbb{E}_{\mathcal{D}_\zeta}[\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \beta_t^2 \mathbf{I}_{2d})]} \\
&= -\beta_t^{-2} \mathbf{x}_t + \beta_t^{-2} \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t) \boldsymbol{\mu}_t(\zeta)] \\
&= -\beta_t^{-2} \mathbf{x}_t + \alpha_t \beta_t^{-2} \begin{bmatrix} \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)\zeta] \mathbf{u} \\ \mathbb{E}_{\mathcal{D}_\zeta}[\pi_t(\zeta, \mathbf{x}_t)(1-\zeta)] \mathbf{v} \end{bmatrix}
\end{aligned}$$

where with a slight abuse of notation, we let $\mathbf{x}_t = [\mathbf{x}_t^{(1)\top}, \mathbf{x}_t^{(2)\top}]^\top$ and denote

$$\pi_t(\zeta, \mathbf{x}_t) = \frac{\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \boldsymbol{\Sigma}_t)}{\mathbb{E}_{D_\zeta}[\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\zeta), \boldsymbol{\Sigma}_t)]}.$$

□

Proof of Theorem 3.4. According to the decomposition of the rule-respecting error in terms of bias and variance, we have $\mathcal{E}_{\text{mse}} = \mathcal{E}_{\text{bias}}^2 + \mathcal{E}_{\text{variance}}$, where we compute

$$\begin{aligned}
\mathcal{E}_{\text{bias}} &= \left| \sum_{r=1}^m \mathbb{E}\Big[\sigma\big(\langle \mathbf{w}_r^{(1)}, \mathbf{x}_t^{(1)} \rangle\big)\Big] \langle \mathbf{w}_r^{(1)}, \mathbf{u} \rangle + \sum_{r=1}^m \mathbb{E}\Big[\sigma\big(\langle \mathbf{w}_{r,t}^{(2)}, \mathbf{x}_t^{(2)} \rangle\big)\Big] \langle \mathbf{w}_{r,t}^{(2)}, \mathbf{v} \rangle - \frac{\alpha_t}{\beta_t^2} \right| \\
\mathcal{E}_{\text{variance}} &= \text{Var}\Big(\sum_{r=1}^m \sigma(\langle \mathbf{w}_{r,t}^{(1)}, \mathbf{x}_t^{(1)} \rangle) \langle \mathbf{w}_{r,t}^{(1)}, \mathbf{u} \rangle + \sum_{r=1}^m \sigma(\langle \mathbf{w}_{r,t}^{(2)}, \mathbf{x}_t^{(2)} \rangle) \langle \mathbf{w}_{r,t}^{(2)}, \mathbf{v} \rangle \Big)
\end{aligned}$$

where we use the law of total variance and denote $\text{Var}_{|\zeta} = \text{Var}(\cdot|\zeta)$.

Given the gradient direction only consists of $\mathbf{u}, \mathbf{w}_{r,t}^{(1),0}$ and $\mathbf{v}, \mathbf{w}_{r,t}^{(2),0}$ respectively for the two patches, we can decompose the weights $\mathbf{w}_{r,t}^{(1)}, \mathbf{w}_{r,t}^{(2)}$ into

$$\begin{aligned}
\mathbf{w}_{r,t}^{(1)} &= \phi_{r,t} \mathbf{w}_{r,t}^0 + \gamma_{r,t} \mathbf{u} \\
\mathbf{w}_{r,t}^{(2)} &= \varphi_{r,t} \mathbf{w}_{r,t}^0 + \varsigma_{r,t} \mathbf{u}
\end{aligned}$$

for $r \in [m]$. In addition, we decompose for each $p = 1, 2$

$$\boldsymbol{\epsilon}_t^{(p)} = \boldsymbol{\epsilon}_{t,-}^{(p)} + \boldsymbol{\epsilon}_{t,\perp}^{(p)} = \mathcal{P}_0^{(p)} \boldsymbol{\epsilon}_t^{(p)} + (I_d - \mathcal{P}_0^{(p)}) \boldsymbol{\epsilon}_t^{(p)}$$

where $\mathcal{P}_0^{(p)}$ denotes the projection onto the span of $\{\mathbf{w}_{1,t}^{(p),0}, ..., \mathbf{w}_{m,t}^{(p),0}\}$. Then we can write for the first patch

$$\begin{aligned}
&\sum_{r=1}^m \sigma(\langle \mathbf{w}_{r,t}^{(1)}, \mathbf{x}_t^{(1)} \rangle) \langle \mathbf{w}_{r,t}^{(1)}, \mathbf{u} \rangle \\
&= \sum_{r=1}^m \sigma(\langle \phi_{r,t} \mathbf{w}_{r,t}^0 + \gamma_{r,t} \mathbf{u}, \alpha_t \zeta \mathbf{u} + \beta_t (\boldsymbol{\epsilon}_{t,-}^{(1)} + \boldsymbol{\epsilon}_{t,\perp}^{(1)}) \rangle) \langle \phi_{r,t} \mathbf{w}_{r,t}^0 + \gamma_{r,t} \mathbf{u}, \mathbf{u} \rangle \\
&= \sum_{r=1}^m \sigma\Big(\phi_{r,t} \alpha_t \zeta \langle \mathbf{w}_{r,t}^0, \mathbf{u} \rangle + \gamma_{r,t} \alpha_t \zeta + \beta_t \langle \phi_{r,t} \mathbf{w}_{r,t}^0 + \gamma_{r,t} \mathbf{u}, \boldsymbol{\epsilon}_{t,-}^{(1)} \rangle + \beta_t \gamma_{r,t} \langle \mathbf{u}, \boldsymbol{\epsilon}_{t,\perp}^{(1)} \rangle \Big) \Big(\phi_{r,t} \langle \mathbf{w}_{r,t}^0, \mathbf{u} \rangle + \gamma_{r,t} \Big) \\
&= \widetilde{\sigma}^{(1)}(\langle \mathbf{u}, \boldsymbol{\epsilon}_{t,\perp}^{(1)} \rangle)
\end{aligned}$$

where $\widetilde{\sigma}^{(1)}(\cdot)$ is a polynomial with coefficients depending on $\gamma_{r,t}, \phi_{r,t}, \alpha_t, \beta_t, \zeta$. Similarly, we can write for the second patch that

$$\sum_{r=1}^m \sigma(\langle \mathbf{w}_{r,t}^{(2)}, \mathbf{x}_t^{(2)} \rangle) \langle \mathbf{w}_{r,t}^{(2)}, \mathbf{v} \rangle = \widetilde{\sigma}^{(2)}(\langle \mathbf{v}, \boldsymbol{\epsilon}_{t,\perp}^{(2)} \rangle)$$

where $\widetilde{\sigma}^{(2)}(\cdot)$ is a polynomial of the same form as $\widetilde{\sigma}^{(1)}(\cdot)$ except that $\phi_{r,t}, \gamma_{r,t}, \zeta$ is respectively replaced with $\varphi_{r,t}, \varsigma_{r,t}, 1-\zeta$. Then we can lower bound the variance by

$$\begin{aligned}\mathcal{E}_{\text{variance}} &\geq \mathbb{E}_{\zeta}\Big[\mathrm{Var}_{|\zeta}\Big(\sum_{r=1}^{m}\sigma(\langle \mathbf{w}_{r,t}^{(1)}, \mathbf{x}_t^{(1)}\rangle)\langle \mathbf{w}_{r,t}^{(1)}, \mathbf{u}\rangle\Big) + \mathrm{Var}_{|\zeta}\Big(\sum_{r=1}^{m}\sigma(\langle \mathbf{w}_{r,t}^{(2)}, \mathbf{x}_t^{(2)}\rangle)\langle \mathbf{w}_{r,t}^{(2)}, \mathbf{v}\rangle\Big)\Big] \\ &\geq \mathbb{E}_{\zeta, \boldsymbol{\epsilon}_{t,-}^{(1)}, \boldsymbol{\epsilon}_{t,-}^{(2)}}\Big[\mathrm{Var}_{|\zeta, \boldsymbol{\epsilon}_{t,-}^{(1)}}\big(\widetilde{\sigma}^{(1)}(\langle \mathbf{u}, \boldsymbol{\epsilon}_{t,\perp}^{(1)}\rangle)\big) + \mathrm{Var}_{|\zeta, \boldsymbol{\epsilon}_{t,-}^{(2)}}\big(\widetilde{\sigma}^{(2)}(\langle \mathbf{v}, \boldsymbol{\epsilon}_{t,\perp}^{(2)}\rangle)\big)\Big]\end{aligned}$$

where we use law of total variance. □

Lemma F.1 ((Cao et al., 2022)). *If* $\mathbf{w}_t^0 \sim \mathcal{N}(0, \sigma_0^2\mathbf{I})$*, we have with probability at least* $1-\delta$

$$\begin{aligned}&\sigma_0^2 d(1-\widetilde{O}(d^{-1/2})) \leq \|\mathbf{w}_t^0\|^2 \leq \sigma_0^2 d(1+\widetilde{O}(d^{-1/2})) \\ &|\langle \mathbf{w}_t^0, \mathbf{u}\rangle| \leq \sqrt{2\log(8/\delta)}\sigma_0, \\ &|\langle \mathbf{w}_t^0, \mathbf{v}\rangle| \leq \sqrt{2\log(8/\delta)}\sigma_0,\end{aligned}$$

Proof of Theorem 3.5. Let $L^{(p)}(\mathbf{W}_t) = \mathbb{E}_{\boldsymbol{\epsilon}_t^{(p)}, \mathbf{x}_0^{(p)}}\|s_w^{(p)}(\mathbf{x}_t^{(p)}) - \boldsymbol{\epsilon}_t^{(p)}\|^2$. Then the loss can be written as

$$L(\mathbf{W}_t) = \sum_{p=1}^{P} L^{(p)}(\mathbf{W}_t) = \sum_{p=1}^{P}\mathbb{E}_{\boldsymbol{\epsilon}_t^{(p)}, \mathbf{x}_0^{(p)}}\Big\|s_w^{(p)}(\mathbf{x}_t^{(p)}) - \boldsymbol{\epsilon}_t^{(p)}\Big\|^2 = \sum_{p=1}^{P}\mathbb{E}_{\boldsymbol{\epsilon}_t^{(p)}, \mathbf{x}_t^{(p)}}\Big\|\langle \mathbf{w}_t^{(p)}, \mathbf{x}_t^{(p)}\rangle \mathbf{w}_t^{(p)} - \frac{1}{\beta_t^2}\mathbf{x}_t^{(p)} - \boldsymbol{\epsilon}_t^{(p)}\Big\|^2$$

We first simplify the loss as follows, where we omit the superscript and consider a single patch due to that each patch is independent and weights are separated.

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\Big\|\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle \mathbf{w}_t - \frac{1}{\beta_t^2}\mathbf{x}_{t,i} - \boldsymbol{\epsilon}_{t,i}\Big\|^2 \\ &= \underbrace{\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\Big\|\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle \mathbf{w}_t\Big\|^2}_{I_1} + \underbrace{\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\Big\|\frac{1}{\beta_t^2}\mathbf{x}_{t,i} + \boldsymbol{\epsilon}_{t,i}\Big\|^2}_{I_2} - 2\underbrace{\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\big[\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle\langle \mathbf{w}_t, \frac{1}{\beta_t^2}\mathbf{x}_{t,i} + \boldsymbol{\epsilon}_{t,i}\rangle\big]}_{I_3}\end{aligned}$$

where we can compute each term following (Han et al., 2024a) as

$$\begin{aligned}I_1 &= \mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}[\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle^2]\|\mathbf{w}_t\|^2 = \Big(\alpha_t^2\langle \mathbf{w}_t, \mathbf{x}_{0,i}\rangle^2 + \beta_t^2\|\mathbf{w}_t\|^2\Big)\|\mathbf{w}_t\|^2 \\ I_2 &= \mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\Big[\frac{\alpha_t^2}{\beta_t^4}\|\mathbf{x}_{0,i}\|^2 + \big(1+\frac{1}{\beta_t}\big)^2\|\boldsymbol{\epsilon}_{t,i}\|^2\Big] = \frac{\alpha_t^2}{\beta_t^4}\|\mathbf{x}_{0,i}\|^2 + \Big(1+\frac{1}{\beta_t}\Big)^2 d \\ I_3 &= \frac{\alpha_t}{\beta_t^2}\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\Big[\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle\Big]\langle \mathbf{w}_t, \mathbf{x}_{0,i}\rangle + \Big(\frac{1}{\beta_t}+1\Big)\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\big[\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle\langle \mathbf{w}_t, \boldsymbol{\epsilon}_{t,i}\rangle\big] = \frac{\alpha_t^2}{\beta_t^2}\langle \mathbf{w}_t, \mathbf{x}_{0,i}\rangle^2 + (1+\beta_t)\|\mathbf{w}_t\|^2.\end{aligned}$$

This suggests

$$\mathbb{E}_{\boldsymbol{\epsilon}_{t,i}}\Big\|\langle \mathbf{w}_t, \mathbf{x}_{t,i}\rangle \mathbf{w}_t - \frac{1}{\beta_t^2}\mathbf{x}_{t,i} - \boldsymbol{\epsilon}_{t,i}\Big\|^2 = \Big(\alpha_t^2\langle \mathbf{w}_t, \mathbf{x}_{0,i}\rangle^2 + \beta_t^2\|\mathbf{w}_t\|^2\Big)\|\mathbf{w}_t\|^2 - \frac{2\alpha_t^2}{\beta_t^2}\langle \mathbf{w}_t, \mathbf{x}_{0,i}\rangle^2 - 2(1+\beta_t)\|\mathbf{w}_t\|^2 + I_2$$

where I_2 is a constant independent of $\mathbf{w}_t$. Then we obtain the loss for the first two patches as

$$\begin{aligned}L^{(1)}(\mathbf{w}_t^{(1)}) &= \big(\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t^{(1)}\|^2\big)\|\mathbf{w}_t^{(1)}\|^2 - \frac{2\alpha_t^2}{\beta_t^2}\mathbb{E}[\zeta^2]\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 - 2(1+\beta_t)\|\mathbf{w}_t^{(1)}\|^2 + I_2 \\ L^{(2)}(\mathbf{w}_t^{(2)}) &= \big(\alpha_t^2\mathbb{E}[(1-\zeta)^2]\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 + \beta_t^2\|\mathbf{w}_t^{(2)}\|^2\big)\|\mathbf{w}_t^{(2)}\|^2 - \frac{2\alpha_t^2}{\beta_t^2}\mathbb{E}[(1-\zeta)^2]\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 - 2(1+\beta_t)\|\mathbf{w}_t^{(2)}\|^2 + I_2.\end{aligned}$$

We next analyze the training dynamics of the gradient descent on the first patch. The second patch follows from similar analysis. For notation clarity, we omit the superscript.

The gradient for the first patch can be computed as

$$\begin{aligned}\nabla L^{(1)}(\mathbf{w}_t) = \|\mathbf{w}_t\|^2(2\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle\mathbf{u} + 2\beta_t^2\mathbf{w}_t) + 2\Big(\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t\|^2\Big)\mathbf{w}_t \\ - \frac{2\alpha_t^2}{\beta_t^2}\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle\mathbf{u} - 2(1+\beta_t)\mathbf{w}_t.\end{aligned}$$

It is noticed that the gradient only consists of directions of $\mathbf{w}_t^0$ and $\mathbf{u}$. It suffices to track the gradient descent dynamics projected to the two directions $\mathbf{u}$ and $\widetilde{\mathbf{w}}_t^0$, where $\widetilde{\mathbf{w}}_t^0 = \mathbf{w}_t^0 - \langle \mathbf{w}_t^0, \mathbf{u}\rangle \mathbf{u}$, i.e.,

$$\begin{aligned}
\langle \mathbf{w}_t^{k+1}, \mathbf{u}\rangle &= \langle \mathbf{w}_t^k, \mathbf{u}\rangle - \eta \langle \nabla^{(1)} L(\mathbf{w}_t), \mathbf{u}\rangle \\
&= \Big(1 + \eta\Big(2\alpha_t^2\beta_t^{-2}\mathbb{E}[\zeta^2] + 2(1+\beta_t) - 2\alpha_t^2\mathbb{E}[\zeta^2]\|\mathbf{w}_t^k\|^2 - 4\beta_t^2\|\mathbf{w}_t^k\|^2 - 2\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t^k, \mathbf{u}\rangle^2\Big)\Big)\langle \mathbf{w}_t^k, \mathbf{u}\rangle \\
\langle \mathbf{w}_t^{k+1}, \widetilde{\mathbf{w}}_t^0\rangle \|\widetilde{\mathbf{w}}_t^0\|^{-1} &= \langle \mathbf{w}_t^k, \widetilde{\mathbf{w}}_t^0\rangle \|\widetilde{\mathbf{w}}_t^0\|^{-1} - \eta \langle \nabla^{(1)} L(\mathbf{w}_t), \widetilde{\mathbf{w}}_t^0\rangle \|\widetilde{\mathbf{w}}_t^0\|^{-1} \\
&= \Big(1 + \eta\Big(2(1+\beta_t) - 4\beta_t^2\|\mathbf{w}_t^k\|^2 - 2\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t^k, \mathbf{u}\rangle^2\Big)\Big)\langle \mathbf{w}_t^k, \widetilde{\mathbf{w}}_t^0\rangle \|\widetilde{\mathbf{w}}_t^0\|^{-1}
\end{aligned}$$

It is clear that gradients become zero only when $\Theta(\|\mathbf{w}_t^k\|^2 + \langle \mathbf{w}_t^k, \mathbf{u}\rangle^2) = \Theta(1)$. This suggests that before convergence, $\|\mathbf{w}_t^k\|^2 = o(1)$ given the initialization is small, i.e., $\sigma_0 = O(d^{-1/2})$. We can then verify that $\langle \nabla^{(1)} L(\mathbf{w}_t), \mathbf{u}\rangle \geq \langle \nabla^{(1)} L(\mathbf{w}_t), \widetilde{\mathbf{w}}_t^0\rangle + C$ for some constant C.

In addition, suppose we decompose $\mathbf{w}_t^k = \phi_t^k \widetilde{\mathbf{w}}_t^0 + \gamma_t^k \mathbf{u}$, we can see

$$\phi_t^k = \langle \mathbf{w}_t^k, \widetilde{\mathbf{w}}_t^0\rangle \|\mathbf{w}_t^0\|^{-2}, \quad \gamma_t^k = \langle \mathbf{w}_t^k, \mathbf{u}\rangle$$

which then implies

$$\|\mathbf{w}_t^k\|^2 = \langle \mathbf{w}_t^k, \widetilde{\mathbf{w}}_t^0\rangle^2 \|\widetilde{\mathbf{w}}_t^0\|^{-2} + \langle \mathbf{w}_t^k, \mathbf{u}\rangle^2.$$

This combined with the fact that $\langle \nabla^{(1)} L(\mathbf{w}_t), \mathbf{u}\rangle \geq \langle \nabla^{(1)} L(\mathbf{w}_t), \widetilde{\mathbf{w}}_t^0\rangle + C$ suggests that $\langle \mathbf{w}_t^{k+1}, \widetilde{\mathbf{w}}_t^0\rangle \|\widetilde{\mathbf{w}}_t^0\|^{-1}$ cannot increase to $\Theta(1)$ without $\langle \mathbf{w}_t^k, \mathbf{u}\rangle$ reaching $\Theta(1)$. Thus at stationary point, we must have both $\langle \mathbf{w}_t^k, \mathbf{u}\rangle, \|\mathbf{w}_t^k\|^2 = \Theta(1)$.

Next, we analyze the stationary point. Given the gradient only consists of directions $\mathbf{w}_t^k$ and $\mathbf{u}$, we have for any stationary point $\mathbf{w}_t$, it satisfies

$$\begin{aligned}
\langle \nabla L^{(1)}(\mathbf{w}_t), \mathbf{w}_t\rangle &= \|\mathbf{w}_t\|^2(2\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle^2 + 2\beta_t^2\|\mathbf{w}_t\|^2) + 2\Big(\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t\|^2\Big)\|\mathbf{w}_t\|^2 \\
&\quad - \frac{2\alpha_t^2}{\beta_t^2}\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle^2 - 2(1+\beta_t)\|\mathbf{w}_t\|^2 = 0 \\
\langle \nabla L^{(1)}(\mathbf{w}_t), \mathbf{u}\rangle &= \|\mathbf{w}_t\|^2(2\alpha_t^2\mathbb{E}[\zeta^2]\mu^2\langle \mathbf{w}_t, \mathbf{u}\rangle + 2\beta_t^2\langle \mathbf{w}_t, \mathbf{u}\rangle) + 2\Big(\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t\|^2\Big)\langle \mathbf{w}_t, \mathbf{u}\rangle \\
&\quad - \frac{2\alpha_t^2}{\beta_t^2}\mathbb{E}[\zeta^2]\mu^2\langle \mathbf{w}_t, \mathbf{u}\rangle - 2(1+\beta_t)\langle \mathbf{w}_t, \mathbf{u}\rangle = 0
\end{aligned}$$

We solve the stationary equalities as

$$\begin{aligned}
\|\mathbf{w}_t^{(1)}\|^2 &= \frac{6\alpha_t^2\beta_t^{-2}\mathbb{E}[\zeta^2] + 6 + 2\beta_t \pm \sqrt{4\alpha_t^4\beta_t^{-4}(\mathbb{E}[\zeta^2])^2 + (56+16\beta_t)\alpha_t^2\beta_t^{-2}\mathbb{E}[\zeta^2] + 28 + 16\beta_t + 4\beta_t^2}}{8\alpha_t^2\mathbb{E}[\zeta^2] + 8\beta_t^2} \\
\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 &= \frac{\|\mathbf{w}_t^{(1)}\|^2}{\alpha_t^2\mathbb{E}[\zeta^2]}\frac{1+\beta_t - 2\beta_t^2\|\mathbf{w}_t^{(1)}\|^2}{2\|\mathbf{w}_t^{(1)}\|^2 - \beta_t^{-2}}
\end{aligned}$$

Similarly, we can compute and solve the stationary point for the second patch where $\mathbb{E}[\zeta^2]$ is replaced with $\mathbb{E}[(1-\zeta)^2]$.

We then compute the bias error as

$$\begin{aligned}
\mathcal{E}_{\text{bias}} &= \mathbb{E}_{\boldsymbol{\epsilon}_t, \mathbf{x}_0}\Big[\langle \mathbf{w}_t^{(1)}, \mathbf{x}_t^{(1)}\rangle \langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle + \langle \mathbf{w}_t^{(2)}, \mathbf{x}_t^{(2)}\rangle \langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle\Big] - \frac{\alpha_t}{\beta_t^2} \\
&= \Big|\alpha_t\mathbb{E}[\zeta]\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \alpha_t\mathbb{E}[1-\zeta]\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 - \frac{\alpha_t}{\beta_t^2}\Big| \\
&= \left|\frac{\mathbb{E}[\zeta]\|\mathbf{w}_t^{(1)}\|^2}{\alpha_t\mathbb{E}[\zeta^2]}\frac{1+\beta_t - 2\beta_t^2\|\mathbf{w}_t^{(1)}\|^2}{2\|\mathbf{w}_t^{(1)}\|^2 - \beta_t^{-2}} + \frac{\mathbb{E}[1-\zeta]\|\mathbf{w}_t^{(2)}\|^2}{\alpha_t\mathbb{E}[(1-\zeta)^2]}\frac{1+\beta_t - 2\beta_t^2\|\mathbf{w}_t^{(2)}\|^2}{2\|\mathbf{w}_t^{(2)}\|^2 - \beta_t^{-2}} - \frac{\alpha_t}{\beta_t^2}\right| = C_0(\mathbb{E}[\zeta], \mathbb{E}[\zeta^2], \alpha_t, \beta_t)
\end{aligned}$$

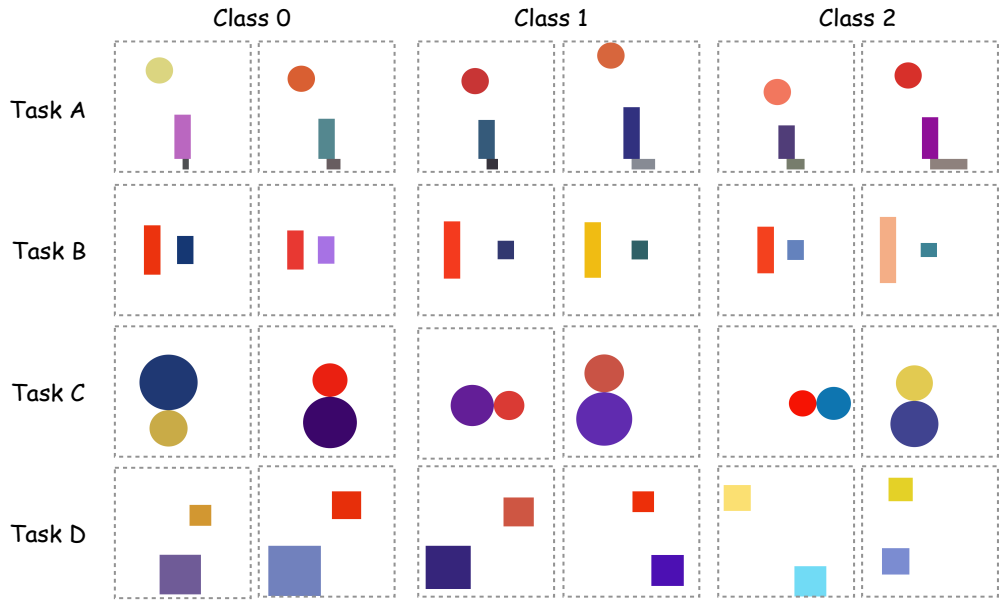


Figure 15: **Constructed contrastive data** includes three classes, which differ in fine-grained rules.

It can be easily verified that there exists a constant bias C_0 that depends on $\mathbb{E}[\zeta], \mathbb{E}[\zeta^2], \alpha_t, \beta_t$. In addition, we compute the variance as

$$\begin{aligned}
\mathcal{E}_{\text{variance}} &= \Big(\alpha_t^2\mathbb{E}[\zeta^2]\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t^{(1)}\|^2\Big)\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \Big(\alpha_t^2\mathbb{E}[(1-\zeta)^2]\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 + \beta_t^2\|\mathbf{w}_t^{(2)}\|^2\Big)\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 \\
&\quad + 2\alpha_t^2\mathbb{E}[\zeta(1-\zeta)]\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 - \Big(\alpha_t\mathbb{E}[\zeta]\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \alpha_t\mathbb{E}[1-\zeta]\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2\Big)^2 \\
&= \alpha_t^2\mathrm{Var}(\zeta)\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^4 + \alpha_t^2\mathrm{Var}(1-\zeta)\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^4 - 2\alpha_t^2\mathrm{Cov}(\zeta, 1-\zeta)\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 \\
&\quad + \beta_t^2\|\mathbf{w}_t^{(1)}\|^2\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t^{(2)}\|^2\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 \\
&= \alpha_t^2\mathrm{Var}\Big(\zeta\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 - (1-\zeta)\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2\Big) + \beta_t^2\|\mathbf{w}_t^{(1)}\|^2\langle \mathbf{w}_t^{(1)}, \mathbf{u}\rangle^2 + \beta_t^2\|\mathbf{w}_t^{(2)}\|^2\langle \mathbf{w}_t^{(2)}, \mathbf{v}\rangle^2 \\
&= C_1(\mathbb{E}[\zeta], \mathbb{E}[\zeta^2], \alpha_t, \beta_t) > 0
\end{aligned}$$

where we see $\mathbb{E}[A^2] - \mathbb{E}[A]^2 = \mathrm{Var}(A) \geq 0$ for arbitrary random variable A. □

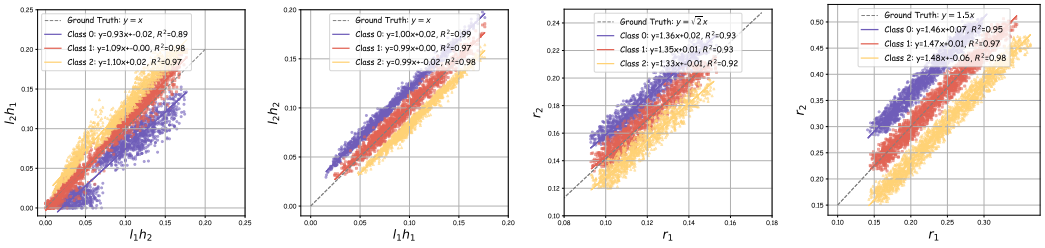
G Experiment Details on Synthetic Data with Two-layer Diffusion Model

In order to verify the theoretical claims on DMs failing to precisely recover the inter-feature rule equation $*$ (in Section 3), we conduct numerical experiments on a two-layer diffusion model on a two-patch data distribution.

Specifically we set $\mathbf{x} = [\mathbf{x}^{(1)\top}, \mathbf{x}^{(2)\top}]$ where $\mathbf{x}^{(1)} = \zeta\mathbf{u}$, $\mathbf{x}^{(2)\top} = (1-\zeta)\mathbf{v}$. Here we set $\mathbf{u} = [1, 0, \cdots 0] \in \mathbb{R}^d$, $\mathbf{v} = [0, 1, 0, \cdots 0] \in \mathbb{R}^d$ with $d = 100$. The score network follows the structure in equation 1 where we consider $\sigma(\cdot)$ to be ReLU, linear, quadratic and cubic activation functions. We set network width $m = 20$. To simulate the DDPM loss in expectation, for each epoch, we sample $n = 1000$ input data $\mathbf{x}_{0,i}, i \in [n]$ and for each data we sample $n_\epsilon = 1000$ standard Gaussian noise $\boldsymbol{\epsilon}_{t,i,j}, i, j \in [1000]$, and consider minimizing the empirical loss

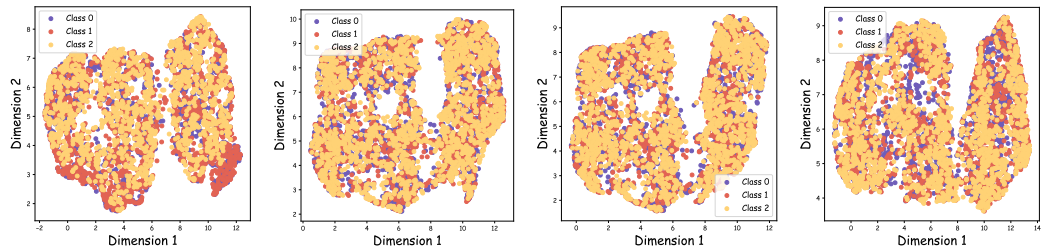
$$L(\mathbf{W}_t) = \frac{1}{nn_\epsilon}\sum_{i=1}^{n}\sum_{j=1}^{n_\epsilon}\sum_{p=1}^{2}\|s_w(\mathbf{x}_{t,i,j}^{(p)}) - \boldsymbol{\epsilon}_{t,i,j}^{(p)}\|^2$$

where $\mathbf{x}_{t,i,j}^{(p)} = \alpha_t\mathbf{x}_{0,i}^{(p)} + \beta_t\boldsymbol{\epsilon}_{t,i,j}^{(p)}, p = 1, 2$. We use gradient descent to train the score network for 5000 epochs. We consider $\alpha_t = \exp(-t)$ and $\beta_t = \sqrt{1-\exp(-2t)}$ where we set $t = 0.2, 0.4, 0.6, 0.8$.



(a) Task A (b) Task B (c) Task C (d) Task D

Figure 16: **Construction of contrastive training data.** For each task, we build a three-class dataset where Class 1 represents samples satisfying fine-grained rules, while Classes 0 and 2 represent samples that only satisfy coarse-grained rules. Based on these constructed contrastive datasets, we train classifiers as additional guidance to improve DDPM's generation.



(a) Task A (b) Task B (c) Task C (d) Task D

Figure 17: **CLIP representation of contrastive training data**. For each task, we use the CLIP model to extract its representations and apply UMAP for dimensionality reduction. We observe that the contrastive data is nearly inseparable, which presents a challenge for training the classifier.

We then check whether learned diffusion models learn the ground-truth rule equation $*$ by plotting the distribution of $\psi_t(\mathbf{x}_t)$ against α_t/β_t^2. The distribution of $\psi_t(\mathbf{x}_t)$ is estimated with 5000 samples $\mathbf{x}_t$.

H Details of Mitigation Strategies

H.1 Details of Guided Diffusion

Guided Diffusion is a common strategy that trains an additional classifier to guide DDPM generation towards desired samples during the sampling process.

Training Details and Results. This section includes the details of training classifiers with contrastive learning as guidance. We use a U-Net classifier $f_\theta(\mathbf{x}, t)$ with guidance weight $\lambda = 1$. Fig.16 visualizes the contrastive datasets constructed for each of the four synthetic tasks.Fig. fig. 15 visualizes the constructed contrastive data, where each dataset includes three sample types that differ only in fine-grained rules and appear nearly identical at a glance. The classifier training for each task is treated as a three-class classification problem with 2000 positive samples (class 1) and 2000 samples per negative class (classes 0 and 2). We use U-Net as the classifier architecture, trained for 20000 iterations with a learning rate of $3e-4$, and a contrastive learning weight $\lambda = 1$. Beyond standard guided diffusion, we dynamically adjust guidance weights (gradient scales) with a piecewise strategy where guidance is activated only in the final 20 denoising steps. The weight linearly increases from 0 to predefined gradient scale factors (7 for Tasks A/C, 10 for Tasks B/D). Through comparation of constant versus piecewise weighting, we report optimal strategies: Task A,B,D for standard sweighting method and Task C employs the piecewise weighting method. As noted in Section 4.2, training high-accuracy classifiers is not easy in our problem, as evidenced by the accuracy of the training data for Tasks A, B, C, and D being 0.57, 0.51, 0.55, and 0.63, respectively.

NT-Xent Loss. NT-Xent Loss (Normalized Temperature-scaled Cross Entropy Loss) (Sohn, 2016) is commonly used in contrastive learning to measure the similarity between positive pairs (similar

samples) and distinguish them from negative pairs (dissimilar samples).

$$\mathcal{L}_{\text{NT-Xent}}(i,j) = -\log\left(\frac{\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N}\mathbf{1}_{[k\neq i]}\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_k)/\tau)}\right), \tag{9}$$

Where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the embeddings of the i-th and j-th samples, $\text{sim}(\mathbf{z}_i,\mathbf{z}_j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|\|\mathbf{z}_j\|}$ is the cosine similarity and τ is the temperature parameter that scales the similarity which we set $\tau = 0.5$ in our experiments.

H.2 Details of Filtered DDPM

Filtered DDPM is a more straightforward strategy that uses a classifier trained on raw images to filter DDPM generations, keeping only samples predicted to satisfy fine-grained rules.

Training Details and Results. Based on the contrastive data constructed in fig. 16, we split the training and test data in an 80:20 ratio and directly train a three-way classifier in the raw image space, using MLP, ResNet-8, and U-Net architectures. These models are trained for 100 epochs with a learning rate of $3e-4$. As shown in fig. 18, the classifiers achieve accuracy between 60% and 80%. While they outperform classifiers trained for guided diffusion due to the noise-free setting, they still fail to achieve 100% accuracy, even for these simple synthesis tasks. Additionally, fig. 17 shows the representations extracted by CLIP (Radford et al., 2021) for each synthetic task, followed by dimensionality reduction using UMAP (McInnes et al., 2018). We observe that the data from different categories in the contrastive data is difficult to distinguish, which presents a challenge for training the classifier. Based on test accuracy, we use the trained MLP model to filter DDPM generations for Tasks A and B, keeping only samples predicted as Class 1 (satisfying fine-grained rules). For Tasks C and D, we use the U-Net model for filtering.

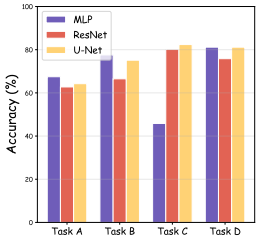


Figure 18: Test accuracy of different architectures.