Neural Residual Diffusion Models for Deep Scalable Vision Generation

Zhiyuan Ma[1], **Liangliang Zhao**[1,2], **Biqing Qi**[3], **Bowen Zhou**[1,3*]
[1]Department of Electronic Engineering, Tsinghua University, Beijing, China
[2]Frontis.AI, Beijing, China
[3]Shanghai AI Laboratory, Shanghai, China
{mzyth,zhoubowen}@tsinghua.edu.cn

Abstract

The most advanced diffusion models have recently adopted increasingly deep stacked networks (e.g., *U-Net* or *Transformer*) to promote the generative emergence capabilities of vision generation models similar to large language models (LLMs). However, progressively deeper stacked networks will intuitively cause numerical propagation errors and reduce noisy prediction capabilities on generative data, which hinders massively deep scalable training of vision generation models. In this paper, we first uncover the nature that neural networks being able to effectively perform generative denoising lies in the fact that the intrinsic residual unit has consistent dynamic property with the input signal's reverse diffusion process, thus supporting excellent generative abilities. Afterwards, we stand on the shoulders of two common types of deep stacked networks to propose a unified and massively scalable Neural Residual Diffusion Models framework (*Neural-RDM* for short), which is a simple yet meaningful change to the common architecture of deep generative networks by introducing a series of learnable gated residual parameters that conform to the generative dynamics. Experimental results on various generative tasks show that the proposed neural residual models obtain state-of-the-art scores on image's and video's generative benchmarks. Rigorous theoretical proofs and extensive experiments also demonstrate the advantages of this simple gated residual mechanism consistent with dynamic modeling in improving the fidelity and consistency of generated content and supporting large-scale scalable training.[1]

1 Introduction

Diffusion models (DMs) [1, 2, 3, 4] have emerged as a class of powerful generative models and have recently exhibited high quality samples in a wide variety of vision generation tasks such as image synthesis [5, 6, 7, 8, 9, 10, 11], video generation [12, 13, 14, 15, 16, 17, 18, 19] and 3D rendering and generation [20, 21, 22, 23, 24]. Relying on the advantage of iterative denoising and high-fidelity generation, DMs have gained enormous attention from the community and have been significantly improved in terms of sampling procedure [25, 26, 27, 28], conditional guidance [29, 30, 31, 32], likelihood maximization [33, 34, 35, 36] and generalization ability [37, 38, 39, 10] in previous efforts.

However, current diffusion models still face a scalability dilemma, which will play an important role in determining whether could support scalable deep generative training on large-scale vision data and give rise to emergent abilities [40] similar to large language models (LLMs) [41, 42]. Representatively, the recent emergence of Sora [43] has pushed the intelligent emergence capabilities of generative models to a climax by treating video models as world simulators. While unfortunately,

*Corresponding Author.
[1]Code is available at https://github.com/ponyzym/Neural-RDM.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Sora is still a closed-source system and the mechanism for the intelligence emergence is still not very clear, but the scalable architecture must be one of the most critical technologies, according to the latest investigation [44] on its reverse engineering.

To alleviate this dilemma and spark further research in the open source community beyond the realms of well established U-Net and Transformers, and enable DMs to be trained in new scalable deep generative architectures, we propose a unified and massively scalable *Residual-style Diffusion Models* framework (*Neural-RDM* for short) with a learnable gating residual mechanism, as shown in Figure 1. The proposed *Neural-RDM* framework aims to unify the current mainstream residual-style generative architecture (e.g., *U-Net* or *Transformer*) and guide the emergence of brand new scalable network architectures with emergent capabilities. To achieve this goal, we first introduce a continuous-time neural ordinary differential equation (ODE) to prove that the generative denoising ability of the diffusion models is closely related to the residual-style network structure, which almost reveals the essential reason why any network rich in residual structure can denoise well: Residual-style neural units implicitly build an ordinary differential equation that can well fit the reverse denoising process through ever-deepening neural units, thus supporting excellent generative abilities. Further, we also show that the gating-residual mechanism plays an important role in adaptively correcting the errors of network propagation and approximating the mean and variance of data, which avoids the adverse factors of network deepening. On this basis, we further present the theoretical advantages of the *Neural-RDM* in terms of stability and score prediction sensitivity when stacking this residual units to a very long depth by introducing another residual-sensitivity ODE. From a dynamic perspective, it reveals that deep stacked networks have the challenge of gradually losing sensitivity as the network progressively deepens, and our proposed gating weights have advantages in reverse suppression and error control.

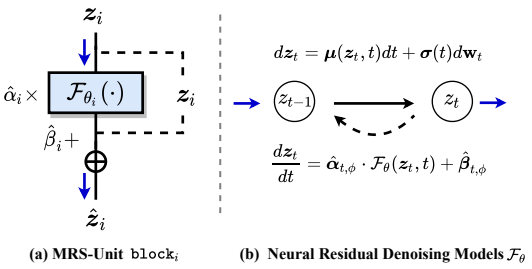


Figure 1: *Neural Residual-style Diffusion Models* framework with massively scalable gating-based **m**inimum **r**esidual **s**tacking **unit** (*mrs-unit*).

Our proposed framework has several theoretical and practical contributions:

Unified residual denoising framework: We unify the residual-style diffusion networks (e.g., *U-Net* and *Transformer*) by introducing a simple gating-residual mechanism and reveal the significance of the residual unit for effective denoising and generation from a brand new dynamics perspective.

Theoretically deep scalability: Thanks to the introduction of continuous-time ODE, we demonstrate that the dynamics equation expressed by deep residual networks possesses excellent dynamic consistency to the denoising probability flow ODE (PF-ODE) [45]. Based on this property, we achieve the simplest improvement to each mrs-unit by parameterizing a learnable mean-variance scheduler, which avoids to manually design and theoretically support massively deep scalable training.

Adaptive stability maintenance and error sensitivity control: When the mrs-units are infinitely stacked to express the dynamics of an overall network $\mathcal{F}_\theta$, the main technical difficulty is how to reduce the numerical errors caused by network propagation and ensure the stability of denoising. By introducing a sensitivity-related ODE in Sec. 2.3, we further demonstrate the theoretical advantages of the proposed gated residual networks in enabling stable denoising and effective sensitivity control. Qualitative and quantitative experimental results also consistently show their effectiveness.

2 Neural Residual Diffusion Models

We propose *Neural-RDM*, a simple yet meaningful change to the architecture of deep generative networks that facilitates effective denoising, dynamical isometry and enables the stable training of extremely deep networks. This framework is supported by three critical theories: **1) *Gating-Residual* ODE** (Sec. 2.1), which defines the dynamics of the **m**inimum **r**esidual **s**tacking **unit** (*mrs-unit* for short) that serves as the foundational denoising module, as shown in Figure 1 (a). Based on this gating-residual mechanism, we then introduce **2) *Denoising-Dynamics* ODE** (Sec. 2.2) to further stack the *mrs-units* to become a continuous-time deep score prediction network $\mathcal{F}_\theta$. Different from previous human-crafted *mean-variance* schedulers (e.g., variance exploding scheduler SMLD [46]

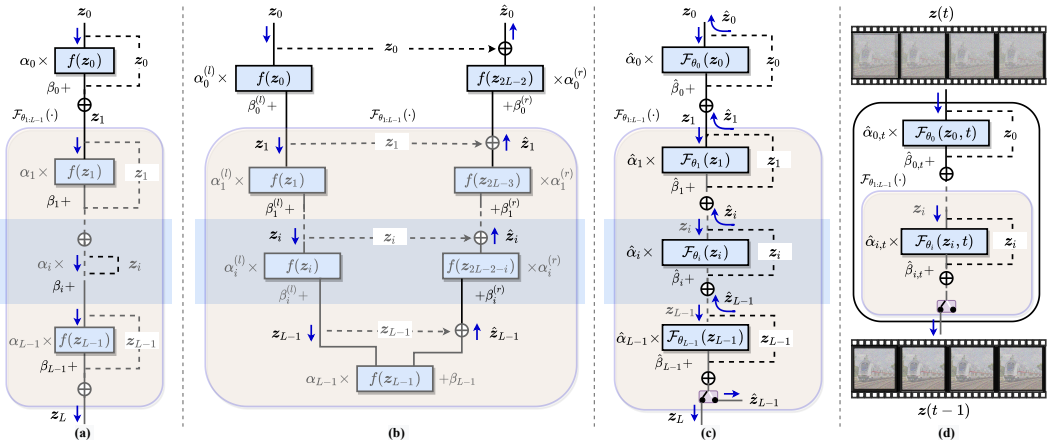


Figure 2: Overview. **(a)** Flow-shaped residual stacking networks. **(b)** U-shaped residual stacking networks. **(c)** Our proposed unified and massively scalable residual stacking architecture (i.e., *Neural-RDM*) with learnable gating-residual mechanism. **(d)** Residual denoising process via *Neural-RDM*.

and variance preserving scheduler DDPM [2]), which may cause concerns about instability in denoising efficiency and quality, we introduce a parametric method to implicitly learn the mean and variance distribution, which lowers the threshold of manual design and enhances the generalization ability of models. Last but not least, to maintain the stability of the deep stacked networks and verify the sensitivity of each residual unit $\mathcal{F}_{\theta_i}(\cdot)$ to the network $\mathcal{F}_\theta$, we stand on the shoulders of the *adjoint sensitivity* method [47, 48] to propose **3) *Residual-Sensitivity* ODE** (Sec. 2.3), which means the sensitivity-related dynamics of each latent state $\boldsymbol{z}_i$ from $\mathcal{F}_{\theta_i}(\cdot)$ to the deep network $\mathcal{F}_\theta$. Through rigorous derivation, we prove that the parameterized gating weights have a positive inhibitory effect on sensitivity decaying as network deepening. We will elaborate on them below.

2.1 Gating-Residual Mechanism

Let $\mathcal{F}_{\theta_i}$ represents the minimum residual unit $\texttt{block}_i$ (Figure 1 (a)), $f(\cdot)$ denotes any feature mapper wrapped by $\mathcal{F}_{\theta_i}$. Instead of propagating the signal $\boldsymbol{z}$ through each of vanilla neural transformation $\hat{\boldsymbol{z}} = f_\theta(\boldsymbol{z})$ directly, we introduce a gating-based residual connection for the signal $\boldsymbol{z}$, which relys on the two learnable gating weights $\hat{\alpha}$ and $\hat{\beta}$ to modulate the non-trivial transformation $\mathcal{F}_{\theta_i}(\boldsymbol{z}_i)$ as,

$$\hat{\boldsymbol{z}}_i = \boldsymbol{z}_i + \hat{\alpha}_i \cdot \mathcal{F}_{\theta_i}(\boldsymbol{z}_i) + \hat{\beta}_i. \tag{1}$$

For a deep neural network $\mathcal{F}_\theta(\cdot)$ with depth L, consider two common residual stacking fashions: Flow-shaped Stacking (**FS**) [49, 50] and U-shaped Stacking (**US**) [51, 52]. For the flow-based deep stacking networks as shown in Figure 2 (a), each residual unit $f(\cdot)$ accepts the output $\boldsymbol{z}_i$ of the previous *mrs-unit* as input, and obtains a new hidden state $\boldsymbol{z}_{i+1}$ through gating-residual connection,

$$\hat{\boldsymbol{z}}_i = \boldsymbol{z}_{i+1} = \boldsymbol{z}_i + [\alpha_i \cdot f_{\theta_i}(\boldsymbol{z}_i) + \beta_i]. \tag{2}$$

Note that Eq. 2 is a refined form of Eq. 1 in the case of flow-shaped stacking. In contrast, for the U-shaped deep stacking networks as in Figure 2 (b), each minimum residual unit contains two symmetrical branches, where the left branch receives the output $\boldsymbol{z}_i$ of the previous *mrs-unit*'s left branch as input (called the *read-in* branch), and the right branch performs the critical nonlinear residual transformation for readout (called the *read-out* branch), which can be formally described as:

$$\hat{\boldsymbol{z}}_i = \underbrace{\alpha_i^{(l)} \cdot f_{\theta_i^{(l)}}(\boldsymbol{z}_i) + \beta_i^{(l)}}_{\text{read-in branch}} \hookrightarrow \underbrace{\boldsymbol{z}_i + \alpha_i^{(r)} \cdot f_{\theta_i^{(r)}}(\boldsymbol{z}_{2L-2-i}) + \beta_i^{(r)}}_{\text{read-out branch}} = \boldsymbol{z}_i + \hat{\alpha}_i \cdot \mathcal{F}_{\theta_i}(\boldsymbol{z}_i) + \hat{\beta}_i. \tag{3}$$

Here Eq. 3 is a refined form of Eq. 1 in the case of U-shaped stacking, $\hat{\alpha}_i$ and $\hat{\beta}_i$ collectively denotes the gating weights from the left and right branches, $\mathcal{F}_{\theta_i}$ is the i-th minimum residual unit of the U-shaped networks, and "$\hookrightarrow$" denotes the skip-connection for "$\boldsymbol{z}_{i+1} \rightarrow \boldsymbol{z}_{2L-2-i}$", which is computed recursively via $\mathcal{F}_{\theta_{i+1:L-1}}$. To enable the networks to be infinitely stacked, we introduce a continuous-time *Gating-Residual* ordinary differential equation (ODE) to express the neural dynamics

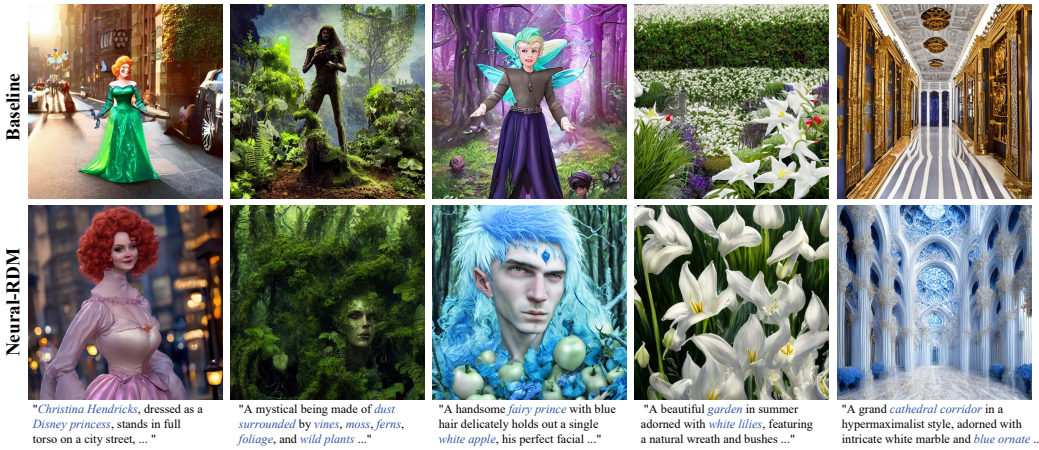


Figure 3: Compared with the latest baseline (SDXL-1.0 [7]), the samples produced by Neural-RDM (trained on JourneyDB [53]) exhibit exceptional quality, particularly in terms of fidelity and consistency in the details of the subjects in adhering to the provided textual prompts.

of these two types of deep stacking networks ($\delta = \frac{1}{L}$, $L \to \infty$ denotes the number of the *mrs-units*),

$$\frac{\boldsymbol{z}_{i+\delta} - \boldsymbol{z}_i}{\delta} = \hat{\boldsymbol{z}}_i - \boldsymbol{z}_i = \hat{\alpha}_i \cdot \mathcal{F}_{\theta_i}(\boldsymbol{z}_i) + \hat{\beta}_i \Longrightarrow \frac{d\boldsymbol{z}_t}{dt} = \hat{\alpha}_\phi \cdot \mathcal{F}_{\theta_t}(\boldsymbol{z}_t) + \hat{\beta}_\phi, \tag{4}$$

where ϕ represents the gating weights, which can be independently trained or fine-tuned without considering the parameters θ of the feature mapping network $\mathcal{F}_\theta(\cdot)$ itself.

2.2 Denoising Dynamics Parameterization

The above-mentioned gating-residual mechanism is utilized to modulate mainstream deep stacking networks and unify them into a residual-style massively scalable generative framework, as shown in Figure 2 (c). Next, we further explore the essential relationship between residual neural networks and score-based generative denoising models from a dynamic perspective.

First, inspired by the theory of continuous-time diffusion models [45, 54], the forward add-noising process can be expressed as a dynamic process with stochastic differential equation (SDE) as,

$$d\boldsymbol{z}_t = \boldsymbol{\mu}(\boldsymbol{z}_t, t)dt + \boldsymbol{\sigma}(t)d\mathbf{w}_t \Longrightarrow \frac{d\boldsymbol{z}_t}{dt} = \boldsymbol{\mu}(\boldsymbol{z}_t, t) + \boldsymbol{\sigma}(t) \cdot \epsilon_t, \epsilon_t \in \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5}$$

which describes a data perturbation process controlled by a *mean-variance* scheduler composed of $\boldsymbol{\mu}(\boldsymbol{z}_t, t)$ and $\boldsymbol{\sigma}(t)$ respectively, $\mathbf{w}_t$ denotes the standard Brownian motion. Compared with the forward process, the core of the diffusion model is to utilize a deep neural network (as deep and large as possible) for score-based reverse prediction [46, 55]. A remarkable property of this SDE is the existence of a reverse ODE (also dubbed as the *Probability Flow* (PF) ODE by [45]), which retain the same marginal probability densities as the SDE (See Appendix. A.2 for detailed proof) and could effectively guide the dynamics of the reverse denoising, it can be formally described as,

$$\frac{d\boldsymbol{z}_t}{dt} = \boldsymbol{\mu}(\boldsymbol{z}_t, t) - \frac{1}{2}\boldsymbol{\sigma}(t)^2 \cdot \Big[\nabla_z \log p_t(\boldsymbol{z}_t)\Big] = \hat{\boldsymbol{\alpha}}_{t,\phi} \cdot \mathcal{F}_\theta(\boldsymbol{z}_t, t) + \hat{\boldsymbol{\beta}}_{t,\phi}, \tag{6}$$

where $\nabla_z \log p_t(\boldsymbol{z}_t)$ denotes the gradient of the log-likelihood of $p_t(\boldsymbol{z}_t)$, which can be estimated by a score matching network $\mathcal{F}_\theta(\boldsymbol{z}_t, t)$. Here we re-parameterize the PF-ODE by utilizing gated weights to replace the manually designed mean-variance scheduler, in which $\hat{\boldsymbol{\alpha}}_{t,\phi}$ and $\hat{\boldsymbol{\beta}}_{t,\phi}$ denotes the time-dependent dynamics parameters, which is respectively parameterized to represent $-\frac{1}{2}\boldsymbol{\sigma}(t)^2$ and $\boldsymbol{\mu}(\boldsymbol{z}_t, t)$ by our proposed gating-residual mechanism. Note that $\mathcal{F}_\theta(\cdot)$ is a score estimation network composed of infinite *mrs-units* $\texttt{block}_i$ (i.e., $\mathcal{F}_{\theta_i}$), which enables massively scalable generative training on large-scale vision data, but also presents the challenge of numerical propagation errors.

2.3 Residual Sensitivity Control

To control the numerical errors in back-propagation and achieve steadily and massively scalable training, we stand on the shoulders of the *adjoint sensitivity* method [47, 48] to introduce another

Architecture	Method	Scalability	Class-to-Image			Text-to-Image		
			FID↓	sFID↓	IS↑	FID↓	sFID↓	IS↑
GAN	BigGAN-deep [56]	✗	6.95	7.36	171.4	-	-	-
	StyleGAN-XL [57]	✗	2.30	4.02	265.12	-	-	-
U-shaped	ADM [58]	✔	10.94	6.02	100.98	-	-	-
	ADM-U	✔	7.49	5.13	127.49	-	-	-
	ADM-G	✔	4.59	5.25	186.70	-	-	-
	LDM-8 [30]	✔	15.51	-	79.03	16.64	11.32	64.50
	LDM-8-G	✔	7.76	-	209.52	9.35	10.02	125.73
	LDM-4	✔	10.56	-	103.49	12.37	11.58	94.65
	LDM-4-G	✔	3.60	-	247.67	3.78	5.89	182.53
F-shaped	DiT-XL/2 [59]	✔	9.62	6.85	121.50	8.53	5.47	144.26
	DiT-XL/2-G	✔	2.27	4.60	278.24	3.53	5.48	175.63
	Latte-XL [60]	✔	2.35	5.17	224.75	2.74	5.35	195.03
Unified	**Neural-RDM-U (Ours)**	✔✔	3.47	5.08	256.55	**2.25**	**4.36**	**235.35**
	Neural-RDM-F (Ours)	✔✔	**2.12**	**3.75**	**295.32**	2.46	5.65	206.32

Table 1: The main results for image generation on ImageNet [61] (Class-to-Image) and JourneyDB [53] (Text-to-Image) with 256×256 image resolution. We highlight the best value in blue, and the second-best value in green. The *Scalability* column indicates the scaling capability of the parameter scale and architecture.

Residual-Sensitivity ODE, which is utilized to evaluate the sensitivity of each residual-state $\boldsymbol{z}_t$ of the *mrs-unit* $\mathcal{F}_{\theta_i}$ to the total loss $\mathcal{L}$ derived by score estimation network $\mathcal{F}_\theta(\cdot)$ (the sensitivity is denoted as $\boldsymbol{s}_t = \frac{d\mathcal{L}}{d\boldsymbol{z}_t}$, δ denotes an infinitesimal time interval) and can be formally described by the chain rule,

$$\boldsymbol{s}_t = \frac{d\mathcal{L}}{d\boldsymbol{z}_t} = \frac{d\mathcal{L}}{d\boldsymbol{z}_{t+\delta}} \cdot \frac{d\boldsymbol{z}_{t+\delta}}{d\boldsymbol{z}_t} = \boldsymbol{s}_{t+\delta} \cdot \frac{d\boldsymbol{z}_{t+\delta}}{d\boldsymbol{z}_t}. \tag{7}$$

On the basis of Eq. 7, we next continue to discuss the dynamic equation of sensitivity changing with time t. First, considering the trivial transformation $f_\theta(\cdot)$ without gating-residual mechanism,

$$d\boldsymbol{z}_{t+\delta} = d\boldsymbol{z}_t + \int_t^{t+\delta} f_\theta(\boldsymbol{z}_t, t)dt. \tag{8}$$

We can rewrite Eq. 7 based on Eq. 8 as:

$$\boldsymbol{s}_t = \boldsymbol{s}_{t+\delta} + \boldsymbol{s}_{t+\delta} \cdot \frac{\partial}{\partial \boldsymbol{z}_t}(\int_t^{t+\delta} f_\theta(\boldsymbol{z}_t, t)dt). \tag{9}$$

The *Residual-Sensitivity* ODE under vanilla situation then can be derived,

$$\frac{d\boldsymbol{s}_t}{dt} = \lim_{\delta \to 0^+} \frac{\boldsymbol{s}_{t+\delta} - \boldsymbol{s}_t}{\delta} = \lim_{\delta \to 0^+} \frac{-\boldsymbol{s}_{t+\delta} \cdot \frac{\partial}{\partial \boldsymbol{z}_t}(\int_t^{t+\delta} f_\theta(\boldsymbol{z}_t)dt)}{\delta} = -\boldsymbol{s}_t \cdot \frac{\partial f_\theta(\boldsymbol{z}_t, t)}{\partial \boldsymbol{z}_t}. \tag{10}$$

According to the derived residual-sensitivity equation in Eq. 10, we further use the Euler solver to obtain the sensitivity $\boldsymbol{s}_{t_0}$ of the starting state $\boldsymbol{z}_{t_0}$ to network $\mathcal{F}_\theta(\cdot)$ as,

$$\boldsymbol{s}_{t_0} = \boldsymbol{s}_{t_L} + \int_{t_L}^{t_0} \frac{d\boldsymbol{s}_t}{dt} dt = \boldsymbol{s}_{t_L} - \int_{t_L}^{t_0} \boldsymbol{s}_t \cdot \frac{\partial f_\theta(\boldsymbol{z}_t, t)}{\partial \boldsymbol{z}_t} dt. \tag{11}$$

Due to the non-negativity of the integral and the gradient $\frac{\partial f_\theta(\boldsymbol{z}_t, t)}{\partial \boldsymbol{z}_t}$ not equals to 0, we can obtain a gradually decaying sensitivity sequence: $\boldsymbol{s}_{t_L} > \boldsymbol{s}_{t_{L-1}} > \cdots > \boldsymbol{s}_{t_0}$. Similarly, when defining parameter-sensitivity $\boldsymbol{s}_\theta = \frac{d\mathcal{L}}{d\boldsymbol{\theta}}$, the same decaying results for $\boldsymbol{s}_{\theta_0}$ can be obtained:

$$\boldsymbol{s}_{\theta_0} = \boldsymbol{s}_{\theta_L} + \int_{t_L}^{t_0} \frac{d\boldsymbol{s}_\theta}{dt} dt = \boldsymbol{s}_{\theta_L} - \int_{t_L}^{t_0} \mathbf{s}_\theta \cdot \frac{\partial f_\theta(\boldsymbol{z}_t, t)}{\partial \theta} dt. \tag{12}$$

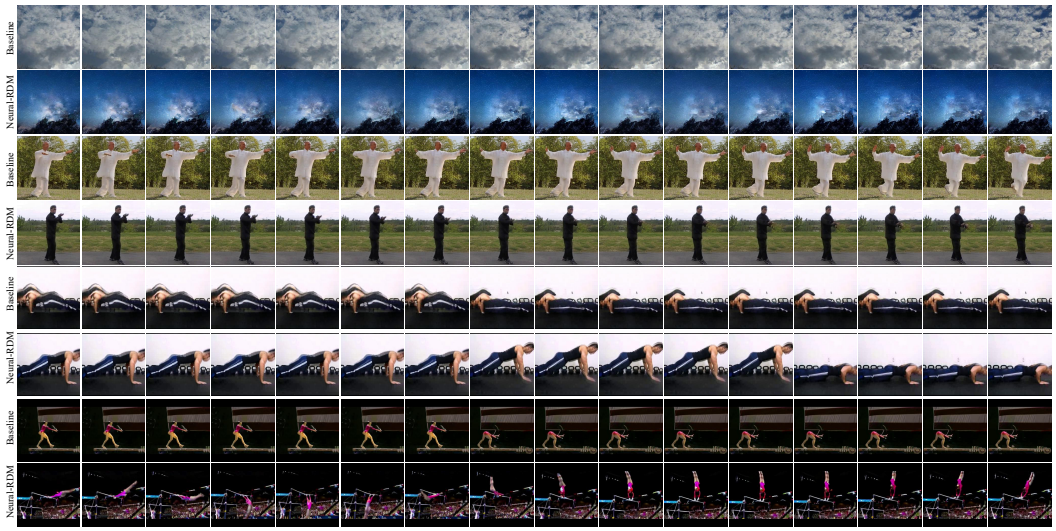


Figure 4: Compared with the latest baseline (Latte-XL [60]), the sample videos from SkyTime-lapse [62], Taichi-HD[63] and UCF101 [64] all exhibit better frame quality, temporal consistency and coherence.

To alleviate this problem, and enable stable training in massively deep scalable architecture, we introduce the following non-trivial solution with gating-residual transformation,

$$d\hat{\boldsymbol{z}}_{t+\delta} = d\hat{\boldsymbol{z}}_t + \int_t^{t+\delta} \Big[\alpha_{t,\phi} \cdot f_\theta(\hat{\boldsymbol{z}}_t) + \beta_{t,\phi}\Big] dt. \tag{13}$$

Substitute Eq. 13 into Eq. 7 to obtain the corrected sensitivity $\hat{\boldsymbol{s}}_t = \frac{d\mathcal{L}}{d\hat{\boldsymbol{z}}_t}$ as:

$$\hat{\boldsymbol{s}}_t = \hat{\boldsymbol{s}}_{t+\delta} + \hat{\boldsymbol{s}}_{t+\delta} \cdot \frac{\partial}{\partial \hat{\boldsymbol{z}}_t} \Big(\int_t^{t+\delta} \Big[\alpha_{t,\phi} \cdot f_\theta(\hat{\boldsymbol{z}}_t) + \beta_{t,\phi}\Big] dt \Big). \tag{14}$$

The non-trivial *Residual-Sensitivity* ODE can be derived as,

$$\frac{d\hat{\boldsymbol{s}}_t}{dt} = \lim_{\delta \to 0^+} \frac{\hat{\boldsymbol{s}}_{t+\delta} - \hat{\boldsymbol{s}}_t}{\delta} = -(\alpha_{t,\phi} \cdot \hat{\boldsymbol{s}}_t) \cdot \frac{\partial f_\theta(\hat{\boldsymbol{z}}_t, t)}{\partial \hat{\boldsymbol{z}}_t} - (\beta_{t,\phi} \cdot \hat{\boldsymbol{s}}_t). \tag{15}$$

Through the Euler solver, we can also obtain the sensitivity $\hat{\boldsymbol{s}}_{t_0}$ of the starting state adjusted by the gating-residual weights,

$$\hat{\boldsymbol{s}}_{t_0} = \hat{\boldsymbol{s}}_{t_L} + \int_{t_L}^{t_0} \frac{d\hat{\boldsymbol{s}}_t}{dt} dt = \hat{\boldsymbol{s}}_{t_L} - \int_{t_L}^{t_0} \Big[(\alpha_{t,\phi} \cdot \hat{\boldsymbol{s}}_t) \cdot \frac{\partial f_\theta(\hat{\boldsymbol{z}}_t, t)}{\partial \hat{\boldsymbol{z}}_t} + (\beta_{t,\phi} \cdot \hat{\boldsymbol{s}}_t)\Big] dt. \tag{16}$$

Where $\alpha_{t,\phi}$ and $\beta_{t,\phi}$ adaptively modulate and update the sensitivity of each *mrs-unit* to the final loss, which supports being trained through minimizing $\mathcal{L}_s = ||\mathcal{F}_\theta(\boldsymbol{z}_t, t) - \nabla_z \log p_t(\boldsymbol{z}_t)||_2^2 + \gamma \cdot \sum_L ||\alpha_{t,\phi} \cdot \frac{\partial f_\theta(\hat{\boldsymbol{z}}_t, t)}{\partial \hat{\boldsymbol{z}}_t} - \beta_{t,\phi}||_2^2$ in full-parameter training or model fine-tuning fashions.

3 Experiments

We present the main experimental settings in Sec. 3.1. To evaluate the generative performance of Neural-RDM, we compare it with state-of-the-art conditional/unconditional diffusion models for image synthesis and video generation in Sec. 3.2 and Sec. 3.3 respectively. We also visualize and analyze the effects of the proposed gated residuals and illustrate their advantages in enabling deep scalable training, which are presented in Sec. 3.4 and Sec. 3.5.

3.1 Experimental Settings

Datasets. For image synthesis tasks, we train and evaluate the **Class-to-Image** generation models on the ImageNet [61] dataset and train and evaluate the **Text-to-Image** generation models on the

Method	Scalability	Frame Evaluation		None-to-Video		Class-to-Video
		FID↓	IS↑	SkyTimelapse (FVD↓)	Taichi-HD (FVD↓)	UCF-101 (FVD↓)
MoCoGAN [71]	✗	23.97	10.09	206.6	-	2886.9
MoCoGAN-HD [72]	✗	7.12	23.39	164.1	128.1	1729.6
DIGAN [73]	✗	19.10	23.16	83.11	156.7	1630.2
StyleGAN-V [70]	✗	9.45	23.94	79.52	-	1431.0
MoStGAN-V [74]	✗	-	-	65.30	-	1380.3
PVDM [75]	✓	29.76	60.55	75.48	540.2	1141.9
LVDM [12]	✓	-	-	95.20	99.0	**372.0**
VideoGPT [76]	✓	22.70	12.61	222.7	-	2880.6
Latte-XL [60]	✓	5.02	68.53	59.82	159.60	477.97
Neural-RDM (Ours)	✓✓	3.35	85.97	**39.89**	**91.22**	**461.03**

Table 2: The main results for video generation on the SkyTimelapse [62], Taichi-HD [63] and UCF-101 [64] with 256×256 resolution of each frame. We highlight the best value in blue, and the second-best value in green.

MSCOCO [65] and JourneyDB [53] datasets. All images are resized to 256×256 resolution for training. For video generation tasks, we follow the previous works [12, 60] to train **None-to-Video** (*i.e.,* unconditional video generation) models on the SkyTimelapse [62] and Taichi [63] datasets, and train **Class-to-Video** models on the UCF-101 [64] dataset. Moreover, we follow previous works [12, 60] to sample 16-frame video clips from these video datasets and then resize all frames to 256×256 resolution for training and evaluation.

Implementation details. We implement our Neural-RDMs into Neural-RDM-U (U-shaped) and Neural-RDM-F (Flow-shaped) two versions on top of the current state-of-the-art diffusion models LDM [30] and Latte [60] for image generation, and further employ the Neural-RDM-F version for video generation. Specifically, we first load the corresponding pre-trained models and initialize gating parameters $\{\alpha = 1, \beta = 0\}$ of each layer, then perform full-parameter fine-tuning to implicitly learn the distribution of the data for acting as a parameterized mean-variance scheduler. During the training process, we adopt an explicit supervision strategy to enhance the sensitivity correction capabilities of α and β for deep scalable training, where the explicitly supervised hyper-parameter γ is set to 0.35. Eventually, we utilize the AdamW optimizer with a constant learning rate of 5×10^4 for all models and exploit an exponential moving average (EMA) strategy to obtain and report all results.

Evaluation metrics. Following the previous baselines [30, 58, 59, 60], we adopt Fréchet Inception Distance (FID) [66], sFID [67] and Inception Score (IS) [68] to evaluate the image generation quality and the video frame quality (except for sFID). Furthermore, we utilize a Fréchet Video Distance (FVD) [69] metric similar with FID to evaluate the unconditional and conditional video generation quality. Among these metrics, FVD is closer to human subjective judgment and thus better reflects the visual quality of the generated video content. Adhering to the evaluation guidelines proposed by StyleGAN-V [70], we calculate the FVD scores by analyzing 2048 generated video clips with each clip consists of 16 frames.

Baselines. We compare the proposed method with the recent state-of-the-art baselines, and categorize them into three groups: 1) **GAN-based.** BigGAN-deep [56] and StyleGAN-XL [57] for image task, MoCoGAN [71], MoCoGAN-HD [72], DIGAN [73], StyleGAN-V [70] and MoStGAN-V [74] for video task. 2) **U-shaped.** ADM [58] and LDM [30] for image task, PVDM [75] and LVDM [12] for video task. 3) **F-shaped.** DiT-XL/2 [59] and Latte-XL [60] for image task, VideoGPT [76] and Latte-XL [60] (with temporal attention learning) for video task.

3.2 Experiments on Image Synthesis with Deep Scalable Spatial Learning

For a more objective comparison, we maintain approximately the same model size to perform class-conditional and text-conditional image generation experiments, which are shown in Table 1. From Table 1, it can be observed that our Neural-RDMs have obtained state-of-the-art results. Specifically,

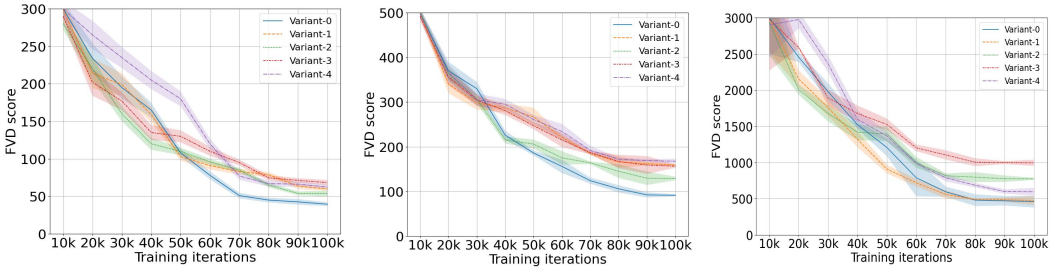


Figure 6: **(a)**, **(b)**, and **(c)** respectively illustrate the performance of the five residual structures variant models across the SkyTimelapsee [62], Taichi-HD[63], and UCF-101 [64].

the flow-based version (i.e., Neural-RDM-F) consistently outperforms all class-to-image baselines in all three image's generative benchmarks and meanwhile obtains relatively suboptimal results on another text-to-image evaluations. It is worth noting that another Neural-RDM-U version have made up for this shortcoming and achieved optimal results, which may benefit from the more powerful semantic guidance abilities of the cross-attention layer built into U-Net. To more clearly present the actual effects of the gated residuals, we further perform qualitative comparative experiments, which are shown in Figure 3. Compared with the latest baseline (SDXL-1.0 [7]), we can observe that the samples produced by Neural-RDM exhibit exceptional quality, particularly in terms of fidelity and consistency in the details of the subjects in adhering to the provided textual prompts, which consistently demonstrates the effectiveness of our proposed approach in deep scalable spatial learning.

3.3 Experiments on Video Generation with Deep Scalable Temporal Learning

To further explore the effectiveness and specific contributions of proposed gating-residual mechanism in temporal learning, we continue to perform the video generation evaluations, which are shown in Table 2. From Table 2, we find that our model (flow-shaped version) basically achieves the best results (except for the second-best results in **class-to-video** evaluation). Specifically, compare with Latte-XL [60], Neural-RDM respectively achieves an improvement of 33.3% and 42.8% in FVD scores on SkyTimelapse and Taichi-HD datasets, which hints the powerful potential of flow-based deep residual networks in promoting generative emergent capabilities of video models. Furthermore, we exhibit a number of visual comparison results of the 16-frames video produced by Neural-RDM and baseline (Latte-XL [60]), as shown in Figure 4. We can observe that some generated frames from the baseline partially exhibits poor quality and temporal inconsistency. Compare with the baseline, Neural-RDM maintains temporal coherence and consistency, resulting in smoother and more dynamic video frames, which further reflects the effectiveness of proposed method in both quantitative and qualitative evaluations.

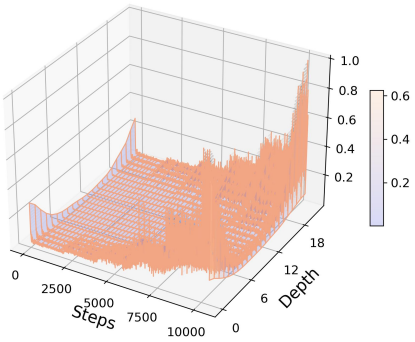


Figure 5: The sensitivity of α and β at different depths of the residual denoising network during the training process.

3.4 The Analyses of Gating Residual Sensitivity

To better illustrate the advantage of the gated residuals and understand the positive suppression effect for sensitivity attenuation as network deepening, we visualize the normalized sensitivity at different depths of our Neural-RDM during the training process, as shown in Figure 5. From Figure 5, we can observe that α and β can adaptively modulate the sensitivity of each *mrs-unit* to correct the denoising process as network deepening, which is consistent with Eq. 16. Moreover, we can also observe that at the beginning of training, the sensitivity scores are relatively low. As training advances, α and β are supervised to correct the sensitivity until obtaining relatively higher sensitivity scores.

3.5 Comparison Experiments of Gating Residual Variants and Deep Scalability

To explore the actual effects of different residual settings in deep training, we first perform the comparison experiments on 5 different residual variants: ***1)*** *Variant-0 (Ours)*: $z_{i+1} = z_i + \alpha f(z_i) + \beta$; ***2)*** *Variant-1 (AdaLN [77])*: $z_{i+1} = z_i + f(\alpha z_i + \beta)$; ***3)*** *Variant-2*: $z_{i+1} = \alpha z_i + f(z_i) + \beta$; ***4)*** *Variant-3 (ResNet [78])*: $z_{i+1} = z_i + f(z_i)$; ***5)*** *Variant-4 (ReZero [79])*: $z_{i+1} = z_i + \alpha f(z_i)$. We utilize Latte-XL as backbone to train each variant from scratch and then evaluate their performance for video generation. As depicted in Figure 6, as the number of training steps increases, almost all variants can converge effectively, but only *Variant-0* (Our approach) achieves the best FVD scores. We speculate that it may be because this post-processing gating-residual setting maintains excellent dynamic consistency with the reverse denoising process, thus achieving better performance.

Moreover, we further perform the deeep scalability experiments, which are shown in Figure 7. We can observe that as the depth of residual units increases, the performance of the model can be further improved, which illustrates the positive correlation between model performance and the depth of residual units and further highlights the deep scalability advantage of our Neural-RDM.

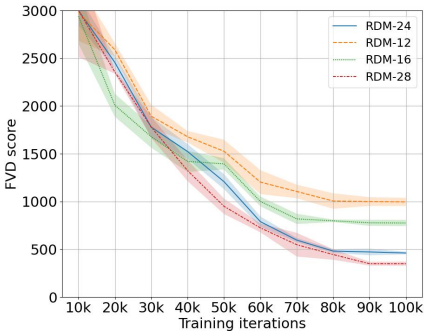


Figure 7: The performance of Neural-RDM with different network depths on the UCF-101 dataset [64].

4 Related Work

Deep Residual Networks. Most common deep residual networks can be divided into two types of architectures: flow-shaped stacking (FS) and u-shaped stacking (US) architectures. As a milestone of flow-based deep residual networks, ResNet [78] has led the research of visual understanding tasks [80]. In fact, the pratices [81, 82] and theories [83, 84, 85] that introducing the highway connections [86, 87] have been studied for a long time and have demonstrated excellent advantages in dealing with vanishing/exploding gradients and numerical propagation errors in deep stacked networks. Different from ResNet, U-Net [51] is a leader of u-shaped networks and almost dominated diffusion-based generative models [2, 30]. Though achieving remarkable success, both types of CNN-based models still face concerns about training efficiency. Recent years, Transformer [49] and ViT [50] have emerged as new state-of-the-art backbones in computer vision and multimodal [88, 89, 90, 91] and have also gained prominence in various diffusion models. Among them, DiT [59] and U-ViT [52] are two representative works by respectively adopting flow-shaped and u-shaped residual stacking fashions, which have enabled many studies on deep generative models [60, 92, 93, 94, 95]. In this work, we unify the above two types of residual stacking architectures from a dynamic perspective and propose a unified and deep scalable neural residual framework with a same gating-residual ODE.

Diffusion Models. Recent years has witnessed the remarkable success of diffusion models [2, 3, 4], due to their impressive generative capabilities. Previous efforts mainly focus on sampling procedure [25, 26, 27, 28], conditional guidance [31, 32, 96, 97, 98], likelihood maximization [33, 34, 35, 36] and generalization ability [37, 99, 39, 10] and have gained enormous attention. Recently, a major research topic on diffusion models is scalability. DiT [59] is one of the most representative models by exploiting a scalable Transformer to train latent diffusion models for image generation. Latte [60] stands on the shoulders of DiT to further perform temporal learning for video generation. However, both Latte and DiT adopt the residual structure of Transformer by default and utilize S-AdaLN to incorporate guidance information, they generally lack: 1) attention to the residual structure and 2) study the dynamic nature of the deep generative models, and 3) ignore the error propagation issues from deeper networks and therefore are still limited by the bottleneck of massively scalable training.

Overall, we practically unify u-shaped and flow-shaped stacking networks and to propose a unified and deep scalable neural residual diffusion model framework. Moreover, we theoretically parameterize the previous human-designed mean-variance scheduler and demonstrate excellent dynamics consistency.

5 Conclusion

In this paper, we have presented Neural-RDM, a simple yet meaningful change to the architecture of deep generative networks that facilitates effective denoising, dynamical isometry and enables the stable training of extremely deep networks. Further, we have explored the nature of two common types of neural networks that enable effective denoising estimation. On this basis, we introduce a parametric method to replace previous human-designed mean-variance schedulers into a series of learnable gating-residual weights. Experimental results on various generative tasks show that Neural-RDM obtains the best results, and extensive experiments also demonstrate the advantages in improving the fidelity, consistency of generated content and supporting large-scale scalable training.

Acknowledgments and Disclosure of Funding

This work is supported by the National Science and Technology Major Project (2023ZD0121403), National Natural Science Foundation of China (No. 62406161), China Postdoctoral Science Foundation (No. 2023M741950), and the Postdoctoral Fellowship Program of CPSF (No. GZB20230347).

References

[1] Zhiyuan Ma, Yuzhu Zhang, Guoli Jia, Liangliang Zhao, Yichao Ma, Mingjie Ma, Gaofeng Liu, Kaiyan Zhang, Jianjun Li, and Bowen Zhou. Efficient diffusion models: A comprehensive survey from principles to practices. *arXiv preprint arXiv:2410.11795*, 2024.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in NeurIPS*, 33:6840–6851, 2020.

[3] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171, 2021.

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

[5] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[6] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[7] 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.

[8] Zhiyuan Ma, Guoli Jia, Biqing Qi, and Bowen Zhou. Safe-sd: Safe and traceable stable diffusion with text prompt trigger for invisible generative watermarking. In *ACM Multimedia 2024*.

[9] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[10] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[11] Zhiyuan Ma, Guoli Jia, and Bowen Zhou. Adapedit: Spatio-temporal guided adaptive editing algorithm for text-based continuity-sensitive image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4154–4161, 2024.

[12] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022.

[13] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

[14] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[15] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[16] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. *arXiv preprint arXiv:2401.09047*, 2024.

[17] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023.

[18] Weimin Wang, Jiawei Liu, Zhijie Lin, Jiangqiao Yan, Shuo Chen, Chetwin Low, Tuyen Hoang, Jie Wu, Jun Hao Liew, Hanshu Yan, et al. Magicvideo-v2: Multi-stage high-aesthetic video generation. *arXiv preprint arXiv:2401.04468*, 2024.

[19] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

[20] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022.

[21] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2023.

[22] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[23] Joseph Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023.

[24] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. *arXiv preprint arXiv:2403.12008*, 2024.

[25] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, pages 42390–42402. PMLR, 2023.

[26] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021.

[27] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 32211–32252, 2023.

[28] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[29] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[30] 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*, pages 10684–10695, 2022.

[31] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

[32] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[33] Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang, and Il-Chul Moon. Maximum likelihood training of implicit nonlinear diffusion models. *arXiv preprint arXiv:2205.13699*, 2022.

[34] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023.

[35] Cheng Lu, Kaiwen Zheng, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *International Conference on Machine Learning*, pages 14429–14460. PMLR, 2022.

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

[37] Yuhan Li, Yishun Dou, Xuanhong Chen, Bingbing Ni, Yilin Sun, Yutian Liu, and Fuzhen Wang. Generalized deep 3d shape prior via part-discretized diffusion process. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16784–16794, 2023.

[38] Zhiyuan Ma, Jianjun Li, Bowen Zhou, et al. Lmd: Faster image reconstruction with latent masking diffusion. *arXiv preprint arXiv:2312.07971*, 2023.

[39] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4296–4304, 2024.

[40] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[41] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and Yasmine et al Babaei. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023.

[42] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[43] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[44] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

[45] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

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

[47] ML Chambers. The mathematical theory of optimal processes. *Journal of the Operational Research Society*, 16(4):493–494, 1965.

[48] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241, 2015.

[52] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[53] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Advances in Neural Information Processing Systems*, 36, 2024.

[54] 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.

[55] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.

[56] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[57] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[58] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in NeurIPS*, 34:8780–8794, 2021.

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

[60] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[62] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2018.

[63] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019.

[64] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[65] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[66] Mandi Luo, Jie Cao, Xin Ma, Xiaoyu Zhang, and Ran He. Fa-gan: Face augmentation gan for deformation-invariant face recognition. *IEEE Transactions on Information Forensics and Security*, 16:2341–2355, 2021.

[67] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, pages 7958–7968. PMLR, 2021.

[68] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017.

[69] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[70] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022.

[71] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[72] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *International Conference on Learning Representations*, 2020.

[73] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *International Conference on Learning Representations*, 2021.

[74] Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Mostgan-v: Video generation with temporal motion styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5661, 2023.

[75] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18456–18466, 2023.

[76] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[77] Yunhui Guo, Chaofeng Wang, Stella X Yu, Frank McKenna, and Kincho H Law. Adaln: a vision transformer for multidomain learning and predisaster building information extraction from images. *Journal of Computing in Civil Engineering*, 36(5):04022024, 2022.

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[79] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, pages 1352–1361. PMLR, 2021.

[80] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[81] Tapani Raiko, Harri Valpola, and Yann LeCun. Deep learning made easier by linear transformations in perceptrons. In *Artificial intelligence and statistics*, pages 924–932. PMLR, 2012.

[82] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[83] Nicol Schraudolph. Accelerated gradient descent by factor-centering decomposition. *Technical report/IDSIA*, 98, 1998.

[84] Nicol N Schraudolph. Centering neural network gradient factors. In *Neural Networks: Tricks of the Trade*, pages 207–226. Springer, 2002.

[85] Tommi Vatanen, Tapani Raiko, Harri Valpola, and Yann LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part I 20*, pages 442–449. Springer, 2013.

[86] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[87] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *Advances in neural information processing systems*, 28, 2015.

[88] Zhiyuan Ma, Jianjun Li, Guohui Li, and Kaiyan Huang. Cmal: A novel cross-modal associative learning framework for vision-language pre-training. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4515–4524, 2022.

[89] Zhiyuan Ma, Jianjun Li, Guohui Li, and Yongjing Cheng. Unitranser: A unified transformer semantic representation framework for multimodal task-oriented dialog system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 103–114, 2022.

[90] Zhiyuan Ma, Zhihuan Yu, Jianjun Li, and Guohui Li. Hybridprompt: bridging language models and human priors in prompt tuning for visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13371–13379, 2023.

[91] Xinwei Long, Jiali Zeng, Fandong Meng, Zhiyuan Ma, Kaiyan Zhang, Bowen Zhou, and Jie Zhou. Generative multi-modal knowledge retrieval with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18733–18741, 2024.

[92] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024.

[93] Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

[94] 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.

[95] Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2023.

[96] Yue Ma, Yingqing He, Hongfa Wang, Andong Wang, Chenyang Qi, Chengfei Cai, Xiu Li, Zhifeng Li, Heung-Yeung Shum, Wei Liu, et al. Follow-your-click: Open-domain regional image animation via short prompts. *arXiv preprint arXiv:2403.08268*, 2024.

[97] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Ying Shan, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. *arXiv preprint arXiv:2304.01186*, 2023.

[98] Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, et al. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. *arXiv preprint arXiv:2406.01900*, 2024.

[99] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, pages 10696–10706, 2022.

[100] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021.

[101] Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8):802, 2020.

[102] Bernt Øksendal and Bernt Øksendal. *Stochastic differential equations*. Springer, 2003.

A Appendix

A.1 Theoretical Interpretations

In this section, we provide mathematical intuitions for our Neural-RDMs.

Continuous-time Residual Networks. For a deep neural network $\mathcal{F}_\theta(\cdot)$ with depth L, let $\mathcal{F}_{\theta_i}$ represents the minimum residual unit $\texttt{block}_i$ (Figure 1 (a)). Instead of propagating the signal $\boldsymbol{z}$ through vanilla neural transformation $\hat{\boldsymbol{z}} = f_\theta(\boldsymbol{z})$, we introduce a gating-based skip-connection for the signal $\boldsymbol{z}$, which relys on the gating weights $\hat{\alpha}$ and $\hat{\beta}$ to modulate the non-trivial transformation $\mathcal{F}_\theta(\boldsymbol{z})$ as,

$$\hat{\boldsymbol{z}} = \boldsymbol{z} + \hat{\alpha} \cdot \mathcal{F}_\theta(\boldsymbol{z}) + \hat{\beta}. \tag{17}$$

In the case of continuous time, this dynamic equation describing the change process of the signal $\boldsymbol{z}$ is called the ***gating-residual*** **ODE**:

$$\frac{d\boldsymbol{z}_t}{dt} = \hat{\alpha}_\phi \cdot \mathcal{F}_{\theta_t}(\boldsymbol{z}_t) + \hat{\beta}_\phi, \tag{18}$$

Diffusion Probability Models. The diffusion probability models are modeled as: 1) a deterministic forward noising process $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$ from the original image $\mathbf{x}_0$ to a pure-Gaussian distribution $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, which can be formulated in an accumulated form:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{19}$$

2) a iteratively predictable reverse denoising process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$, which can be trained in a simplied denoising objective $\mathcal{L}_{\text{simple}}$ by merging $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ into predicting noise $\boldsymbol{\epsilon}_\theta$,

$$\mathcal{L}_{\text{simple}} = E_{\mathbf{x}_0, t, \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)||_2^2] \tag{20}$$

where $t \sim \mathcal{U}[1, T]$ is time parameters, $\mathcal{U}(\cdot)$ denotes uniform distribution. Moreover, in Stable Diffusion [30], the image $\mathbf{x}_t$ is compressed into a latent variable $\mathbf{z}_t$ by encoder $\mathcal{E}$ for more efficient training, i.e., $\mathbf{z}_t = \mathcal{E}(\mathbf{x}_t)$, thus this preliminary objective is usually defined as making $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ as close to $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ as possible.

Reverse Denoising ODE. A remarkable property of the SDE (Eq. 19) is the existence of a reverse ODE (also dubbed as the *Probability Flow* (PF) ODE by [45]), which retains the same marginal probability densities as the SDE (See Appendix A.2 for detailed proof) and could effectively guide the dynamics of the reverse denoising, it can be formally described as,

$$\frac{d\boldsymbol{z}_t}{dt} = \boldsymbol{\mu}(\boldsymbol{z}_t, t) - \frac{1}{2}\boldsymbol{\sigma}(t)^2 \cdot \Big[\nabla_z \log p_t(\boldsymbol{z}_t)\Big] = \hat{\boldsymbol{\alpha}}_{t,\phi} \cdot \mathcal{F}_\theta(\boldsymbol{z}_t, t) + \hat{\boldsymbol{\beta}}_{t,\phi}, \tag{21}$$

where $\nabla_z \log p_t(\boldsymbol{z}_t)$ denotes the gradient of the log-likelihood of $p_t(\boldsymbol{z}_t)$, which can be estimated by a score matching network $\mathcal{F}_\theta(\boldsymbol{z}_t, t)$.

Dynamics Consistency. Refer to Eq. 18 and Eq. 21, we define this dynamic consistency as: For any time-dependent signal $\boldsymbol{z}_t$, the different dynamics systems describe it with the same motion path (or the same change rate of data distribution). Note that in Eq. 21, we achieve this by using a re-parameterized approach.

Latent Space Projection. The latent space projection is proposed by [30] to compress the input images $\mathbf{x}_0$ into a perceptual high-dimensional space to obtain $\mathbf{z}_0$ by leveraging a pretrained VQ-VAE model [100]. The VQ-VAE is also used by our Neural-RDM, it consists of an encoder $\mathcal{E}$ and a decoder $\mathcal{G}$. The mathematical definition is: Given an input image $x \in \mathbb{R}^{H\times W\times 3}$, the VQ-VAE first compress the image x into a latent variable $\hat{z}$ by encoder $\mathcal{E}$, i.e., $\hat{z} = \mathcal{E}(x)$ and $\hat{z} \in \mathbb{R}^{h\times w\times d}$, where h and w respectively denote scaled height and width (scaled factor $f = H/h = W/w = 8$), and d is the dimensionality of the compressed latent variable. After going through the diffusion step described in Eq. 5 and Eq. 6, the latent variable $\hat{z}$ is updated and finally reconstructed into $\hat{x}$ by decoder $\mathcal{G}$,

$$\hat{x} = \mathcal{G}_\pi(\text{LDM}_{\mathcal{F}_\theta(\cdot)}(\mathcal{E}_\pi(x))), \tag{22}$$

where LDM$(\cdot)$ represents the latent diffusion models (including Unet-based or Transformer-based), θ denotes the parameters of LDM, and π denotes the parameters of the VQVAE that are frozen to train our Neural-RDM models.

A.2 Additional Proofs

Motivated by [101], we follow [45] to give a proof: A remarkable property of the SDE (Eq. 5) is the existence of a reverse ODE (PF-ODE [45]), which retain the same marginal probability densities as the SDE. We consider the SDE in Eq. 5, which possesses the following form:

$$d\boldsymbol{z}_t = \boldsymbol{\mu}(\boldsymbol{z}_t, t)dt + \boldsymbol{\sigma}(\boldsymbol{z}_t, t)d\mathbf{w}_t, \tag{23}$$

where $\boldsymbol{\mu}(\cdot, t) : R^d \to R^d$ and $\boldsymbol{\sigma}(\cdot, t) : R^d \to R^{d\times d}$. The marginal probability density $p_t(\boldsymbol{z}_t)$ evolves according to Kolmogorov's forward equation [102]

$$\begin{aligned}\frac{\partial p_t(\boldsymbol{z})}{\partial t} &= -\sum_{i=1}^{d}\frac{\partial}{\partial z_i}[\mu_i(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)] + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial^2}{\partial z_i \partial z_j}\Big[\sum_{k=1}^{d}\sigma_{ik}(\boldsymbol{z}_t, t)\sigma_{jk}(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)\Big]. \\ &= -\sum_{i=1}^{d}\frac{\partial}{\partial z_i}[\mu_i(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)] + \frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial z_i}\Big[\sum_{j=1}^{d}\frac{\partial}{\partial z_j}\Big[\sum_{k=1}^{d}\sigma_{ik}(\boldsymbol{z}_t, t)\sigma_{jk}(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)\Big]\Big].\end{aligned} \tag{24}$$

Since the sub-part of Eq. 24 can be written in the following form:

$$\begin{aligned}&\sum_{j=1}^{d}\frac{\partial}{\partial z_j}\Big[\sum_{k=1}^{d}\sigma_{ik}(\boldsymbol{z}_t, t)\sigma_{jk}(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)\Big] \\ =&\sum_{j=1}^{d}\frac{\partial}{\partial z_j}\Big[\sum_{k=1}^{d}\sigma_{ik}(\boldsymbol{z}_t, t)\sigma_{jk}(\boldsymbol{z}_t, t)\Big]p_t(\boldsymbol{z}_t) + \sum_{j=1}^{d}\sum_{k=1}^{d}\sigma_{ik}(\boldsymbol{z}_t, t)\sigma_{jk}(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)\frac{\partial}{\partial z_j}\log p_t(\boldsymbol{z}_t) \\ =&p_t(\boldsymbol{z}_t)\nabla\cdot[\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)] + p_t(\boldsymbol{z}_t)\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)\nabla_{\boldsymbol{z}_t}\log p_t(\boldsymbol{z}_t).\end{aligned} \tag{25}$$

Thus we can obtain:

$$\begin{aligned}\frac{\partial p_t(\boldsymbol{z}_t)}{\partial t} &= -\sum_{i=1}^{d}\frac{\partial}{\partial z_i}[\mu_i(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)] + \frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial z_i}\Big[\sum_{j=1}^{d}\frac{\partial}{\partial z_j}\Big[\sum_{k=1}^{d}\sigma_{ik}(\boldsymbol{z}_t, t)\sigma_{jk}(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)\Big]\Big] \\ &= -\sum_{i=1}^{d}\frac{\partial}{\partial z_i}[\mu_i(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)] \\ &\quad + \frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial z_i}\Big[p_t(\boldsymbol{z}_t)\nabla\cdot[\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)] + p_t(\boldsymbol{z}_t)\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)\nabla_{\boldsymbol{z}_t}\log p_t(\boldsymbol{z}_t)\Big] \\ &= -\sum_{i=1}^{d}\frac{\partial}{\partial z_i}\Big\{\mu_i(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t) \\ &\quad - \frac{1}{2}\Big[\nabla\cdot[\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)] + \boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)\nabla_{\boldsymbol{z}_t}\log p_t(\boldsymbol{z}_t)\Big]p_t(\boldsymbol{z}_t)\Big\} \\ &= -\sum_{i=1}^{d}\frac{\partial}{\partial z_i}[\tilde{\mu}_i(\boldsymbol{z}_t, t)p_t(\boldsymbol{z}_t)],\end{aligned} \tag{26}$$

where we define $\tilde{\boldsymbol{\mu}}_i(\cdot)$ as:

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{z}_t, t) \coloneqq \boldsymbol{\mu}(\boldsymbol{z}_t, t) - \frac{1}{2}\nabla\cdot[\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)] - \frac{1}{2}\boldsymbol{\sigma}(\boldsymbol{z}_t, t)\boldsymbol{\sigma}^\top(\boldsymbol{z}_t, t)\nabla_{\boldsymbol{z}_t}\log p_t(\boldsymbol{z}_t). \tag{27}$$

Combining Eq. 26 and Eq. 27, we can conclude that Equation Eq. 26 still describes a Kolmogorov's forward process but with $\tilde{\boldsymbol{\sigma}}(\boldsymbol{z}_t, t) \coloneqq \mathbf{0}$ as:

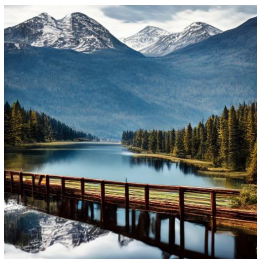
$$d\boldsymbol{z}_t = \tilde{\boldsymbol{\mu}}(\boldsymbol{z}_t, t)dt + \tilde{\boldsymbol{\sigma}}(\boldsymbol{z}_t, t)d\mathbf{w}_t, \quad \tilde{\boldsymbol{\sigma}}(\boldsymbol{z}_t, t) = \mathbf{0}. \tag{28}$$

Which proves that it is actually an ODE after reverse transformation $\tilde{\boldsymbol{\mu}}(\cdot)$:

$$d\boldsymbol{z}_t = \tilde{\boldsymbol{\mu}}(\boldsymbol{z}_t, t)dt, \tag{29}$$

which is essentially the same with our ***Denoising-Diffusion*-ODE** given by Eq. 6. Therefore, we demonstrate the existence of the reverse ODE and the practicality of parameterizing the mean-variance scheduler from the reverse ODE.

A.3 More Generated Images



"A *landscape* featuring a *wooden bridge* over a serene *lake* with a majestic *mountain* in the background."

"A cartoon-style *watercolor* cover illustration featuring *vibrant pink* and *cream hydrangeas* in a Disney-inspired setting, complemented by typography."

"A speckled *headscarf*, a swamp adder, and a *flag of India* depicted in an isometric illustration."

"A *castle* situated in the *mountains* with an array of very high thin *towers* adorned with numerous arrowslits."

"The *King* of Pentacles stands in a regal pose, surrounded by earthly riches, while a *mysterious UFO* hovers above, adding an element of otherworldly intrigue."

"A hyper-realistic portrait of a 17-year-old *English girl* with mismatched eyes, *blonde curly hair* adorned with flowers, holding a flute, radiating pure joy."

"*woman captain* wearing a *cocked hat* stands on a ship, gazing in awe and fear as a huge Cthulhu emerges from the water."

"A *cyberpunk man* with silver skin wearing a helmet featuring a *large glass visor,* holding a rocket launcher in a futuristic setting, depicted with hyper-realistic..."

Figure 8: The samples produced by Neural-RDM (trained on JourneyDB [53]) .

A.4 Limitations

Limitation Discussion. Although significant improvements have been achieved, Neural-RDM still has some limitations, the most important of which is that the gated residual mechanism only inhibits rather than completely avoids the sensitivity decrease and numerical propagation errors caused by the deepening of the network. If we want to completely avoid it, we may have to give up stacking-based deep network architectures, but that will lead to a significant reduction in performance. Therefore, our method chooses to continue to deepen the stacking of the network and suppress error propagation as much as possible in the trade-off between the two.

A.5 Social Impact

Potential Social Implications. We believe that Neural-RDM will bring new thinking about deep network architectures to the generative community, and hopefully promote the generative emergence capabilities of vision generation models in the open source community. In addition, we hope that more researchers can follow the powerful capabilities of residual denoising to build brand new scalable network architectures beyond the realms of well established U-Net and Transformers.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We believe that the main claims made in the abstract and introduction accurately reflect the contributions and scope of the paper, refer to theoretical proof in Sec. 2.1- 2.3 and experimental proof in Sec. 3.2- 3.5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Refer to Sec. A.4 for a discussion of limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We have provided all detailed assumptions and theoretical proofs in the main content and appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We are convinced that we have achieved this and promise to disclose all code details after the paper is accepted.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We are convinced that we have achieved this and promise to disclose all code details after the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We are convinced that we have achieved this, please refer to Sec. 3.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have presented the results of statistical significance in the main quantitative results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have reported the GPU power used in the experiments in Sec. 3.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we comply with all NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we have discussed social impacts in Sec. A.5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Yes, we discuss it in Social Impact but in reality our approach is largely irrelevant to safety precautions.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we have followed these license specifications and accurately stated contributions from previous work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.