

---

# Hierarchical Koopman Diffusion: Fast Generation with Interpretable Diffusion Trajectory

---

**Hanru Bai**

Fudan University  
hrbai23@m.fudan.edu.cn

**Weiyang Ding\***

Fudan University  
dingwy@fudan.edu.cn

**Difan Zou**

The University of Hong Kong  
dzou@cs.hku.hk

## Abstract

Diffusion models have achieved impressive success in high-fidelity image generation but suffer from slow sampling due to their inherently iterative denoising process. While recent one-step methods accelerate inference by learning direct noise-to-image mappings, they sacrifice the interpretability and fine-grained control intrinsic to diffusion dynamics, key advantages that enable applications like editable generation. To resolve this dichotomy, we introduce **Hierarchical Koopman Diffusion**, a novel framework that achieves both one-step sampling and interpretable generative trajectories. Grounded in Koopman operator theory, our method lifts the nonlinear diffusion dynamics into a latent space where evolution is governed by globally linear operators, enabling closed-form trajectory solutions. This formulation not only eliminates iterative sampling but also provides full access to intermediate states, allowing manual intervention during generation. To model the multi-scale nature of images, we design a hierarchical architecture that disentangles generative dynamics across spatial resolutions via scale-specific Koopman subspaces, capturing coarse-to-fine details systematically. We empirically show that the Hierarchical Koopman Diffusion not only achieves competitive one-step generation performance but also provides a principled mechanism for interpreting and manipulating the generative process through spectral analysis. Our framework bridges the gap between fast sampling and interpretability in diffusion models, paving the way for explainable image synthesis in generative modeling.

## 1 Introduction

Diffusion models [29, 7, 30] have achieved remarkable success in image generative tasks [3]. However, despite producing high-fidelity samples, the sampling process of diffusion models typically requires an expensive iterative procedure, which limits their applicability in real-world scenarios [18]. Accelerating sampling in diffusion models thus remains a critical challenge. To overcome this limitation, recent work has focused on efficient one-step inference for diffusion models to replace the costly iterative sampling process.

Existing approaches toward this goal fall into several distinct paradigms. A widely adopted paradigm is distillation-based methods [39, 42], which aim to distill pre-trained diffusion models into efficient one-step generators. Among these, Knowledge Distillation (KD) [17], Progressive Distillation (PD) [27], and (distilled) Rectified Flow [14, 13] are considered classical approaches, laying the foundation

---

\*Corresponding author

for recent advances in one-step generation. Another influential direction involves consistency models [32, 31], which learn time-consistent mappings from noisy inputs to clean data.

Although these methods achieve strong performance in one-step image generation, they fundamentally rely on learning direct noise-to-image mappings, bypassing the temporally coherent denoising trajectory upon which diffusion models are built. As a result, they lack access to intermediate generative states of the diffusion trajectory that are crucial for interpreting and controlling sample evolution from noise to image, thus leading to limited interpretability of the generation process. This lack of interpretability not only obscures how semantic and structural information gradually emerges during sampling but also limits the ability to intervene at specific stages along the generative trajectory—an ability that enables controllable image synthesis at inference time.<sup>2</sup>

Our goal is to develop an explicitly interpretable one-step generation framework, which is not merely to achieve high generation quality in a single step, but to provide a principled understanding of the generative process and thus enable fine-grained control along the diffusion trajectory. Motivated by Koopman operator theory [20, 24], which projects the nonlinear dynamics into an observable space where evolution is linear via function space lifting, we propose an *explicit* one-step generation paradigm that mapping the entire deterministic sampling trajectory of diffusion models into a latent space where closed-form ODE solutions exist. This latent space, theoretically grounded in Koopman theory, is referred to as the Koopman space. In this space, the dynamics of the diffusion process are governed by a linear operator, thus revealing an explicit, analytically tractable form for the evolution to enable an interpretable mapping from noise to data. We realize this framework by jointly learning the mapping and its associated linear operator. In our explicit framework, all intermediate states along trajectories are analytically accessible, thus naturally allowing for introducing additional supervision on these states to guide the noise-to-image mapping more precisely and enabling fine-grained control of the generation dynamics, a capability not afforded by implicit methods.

While the Koopman-based modeling provides a principled way to enable explicit one-step sampling, its standard formulations operate in a single latent space, implicitly assuming uniform dynamics across all spatial and semantic scales. However, generative processes, particularly in visual domains, exhibit inherently multi-scale behaviors: global structures emerge early and evolve slowly, while fine-grained textures form later with more rapid variations [34]. Ignoring this scale-specific nature limits the model’s capacity to represent the full complexity of image generation. To address this limitation, we reformulate a *hierarchical* Koopman modeling framework that explicitly decomposes the generative dynamics across multiple spatial resolutions. Features at different scales are extracted via a U-Net encoder and projected into separate Koopman subspaces, each governed by its own linear operator. This design allows the model to track the evolution of visual content from coarse layout to fine detail, aligning with the hierarchical nature of visual perception and offering improved generation fidelity. Our key contributions are summarized as follows:

- We propose a novel interpretable one-step generation paradigm, referred to as Hierarchical Koopman Diffusion (HKD), via hierarchical Koopman dynamics. This paradigm uniquely integrates explicit intermediate generative states into the process, enabling control along the diffusion trajectory. Moreover, our framework provides a novel analytical aspect for diffusion models using spectral tools in dynamical systems theory to analyze the underlying generative mechanisms.
- We provide a theoretical justification that our Koopman explicit formulation is provably more expressive than directly learning a black-box mapping from noise to image using standard neural networks for learning the diffusion process.
- We conduct experiments on the CIFAR-10 and FFHQ datasets to demonstrate the competitive one-step generation performance of our proposed framework. Beyond generation quality, we interpret the underlying generative dynamics via principled spectral analysis, revealing a quantitative correspondence between spectral components and semantic image attributes. Moreover, we further justify the interpretability of our framework through an image editing experiment that targets frequency-specific intervention at the intermediate stage of the diffusion trajectory.

---

<sup>2</sup>Some distillation-based fast generation methods have explored multi-step editing for text-to-image, which is beyond the scope of this work. We focus on trajectory-level intervention during one-step generation.

## 2 Background

**Diffusion Models.** We consider the continuous-time diffusion models [33], where data is generated by reversing a stochastic process that gradually adds Gaussian noise. Let  $p_{\text{data}}(\mathbf{x})$  denote the data distribution. The diffusion process is described by the stochastic differential equation (SDE):  $d\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t, t)dt + \sigma(t)d\mathbf{w}_t$ , where  $t \in [0, T]$ ,  $\boldsymbol{\mu}(\cdot, \cdot)$  and  $\sigma(\cdot)$  are the drift and diffusion coefficients, respectively.  $\{\mathbf{w}_t\}_{t \in [0, T]}$  represents standard Brownian motion. The distribution of  $\mathbf{x}_t$  is  $p_t(\mathbf{x})$ , with  $p_0(\mathbf{x}) \equiv p_{\text{data}}(\mathbf{x})$ . Notably, there exists an ordinary differential equation (ODE), called the Probability Flow ODE, whose solution trajectories sampled at  $t$  follow  $p_t(\mathbf{x})$ :

$$d\mathbf{x}_t = [\boldsymbol{\mu}(\mathbf{x}_t, t) - \frac{1}{2}\sigma(t)^2 \nabla \log p_t(\mathbf{x}_t)] dt, \quad (1)$$

where  $\nabla \log p_t(\mathbf{x}_t)$  is the score function of  $p_t(\mathbf{x}_t)$ .

**Koopman Operators.** Koopman theory provides a framework for analyzing nonlinear dynamical systems by transforming them into a linear form. The formal definition of the Koopman operator is given below.

**Definition 2.1** (Koopman Operator [21]). *Consider a finite-dimensional state space  $\mathcal{X} \subseteq \mathbb{R}^n$  with state evolution described by  $\mathbf{x}_{t+\Delta t} = \Phi(\mathbf{x}_t)$ ,  $t \in [t_0, t_1]$ , where  $\Phi : \mathcal{X} \rightarrow \mathcal{X}$  is the state transition operator. The Koopman operator  $\mathcal{K}$  is a linear operator that acts on an infinite-dimensional space of observable function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$ , such that*

$$(\mathcal{K} \circ g)(\mathbf{x}) = g(\Phi(\mathbf{x})). \quad (2)$$

Instead of modeling the state  $\mathbf{x}_t$  directly, the Koopman operator captures the evolution of observables  $\mathbf{z}_t = \mathbf{g}(\mathbf{x}) \triangleq [g_1(\mathbf{x}), \dots, g_m(\mathbf{x})]^\top$  in a lifted space, where each  $g_i$  is a component of the observable function  $\mathbf{g}$ . In continuous time, this yields a linear system  $d\mathbf{z}_t/dt = \mathbf{A}\mathbf{z}_t$ , enabling spectral analysis of nonlinear dynamics. Formal details on approximating with finite observables are in App. A.2.

## 3 Hierarchical Koopman Diffusion for Fast Generation

We propose a novel denoising generative model called Hierarchical Koopman Diffusion (HKD) that enables one-step deterministic sampling for diffusion models via Koopman operator theory. By leveraging the Koopman operator’s ability to use tools of linear dynamics such as spectral analysis, our formulation not only enables efficient one-step ODE sampling but also offers a principled dynamical interpretation of generative modeling. This section is organized as follows: we first introduce the theoretical formulation of our framework (Sec. 3.1), then describe the corresponding supervision strategy (Sec. 3.2), and conclude with implementation details (Sec. 3.3). We subsequently present theoretical results that justify the superiority of the proposed method under ideal conditions (Sec. 3.4).

### 3.1 Framework Formulation

**Encoder.** Given the trajectory  $\{\mathbf{x}_t\}_{t \in [\epsilon, T]}$  satisfying ODE in Eq. (1), we construct a hierarchical Koopman representation by applying an encoder

$$\mathcal{E}_\theta : \mathbf{x}_t \in \mathbb{R}^{C \times H \times W} \mapsto \{\mathbf{z}_t^{(l)} \in \mathbb{R}^{d_l \times h_l \times w_l}\}_{l=1}^L.$$

$\mathcal{E}_\theta$  adopts a U-Net style downsampling architecture, just as the right part of Fig. 1. Here,  $d_l$  is the number of latent channels and  $(h_l, w_l)$  is the image resolution at level  $l$ . The encoder approximates independent Koopman observable functions  $\mathbf{g}^{(l)} : \mathbb{R}^{C \times H \times W} \rightarrow \mathbb{R}^{d_l \times h_l \times w_l}$  at each level by obtaining  $\mathbf{z}_t^{(l)} \approx \mathbf{g}^{(l)}(\mathbf{x}_t)$ , where  $\mathbf{g}^{(l)}$  captures the dynamics corresponding to its spatial scale.

**Hierarchical Koopman Subspace.** Each latent feature  $\mathbf{z}_t^{(l)} \in \mathbb{R}^{d_l \times h_l \times w_l}$  is extracted at level  $l$ , with  $\mathbf{z}_t^{(l)}(i, j) \in \mathbb{R}^{d_l}$  denoting the feature vector at spatial position  $(i, j)$ . We model  $\mathbf{z}_t^{(l)}(i, j)$  evolves linearly under a spatially-varying linear operator  $\mathbf{A}^{(l)}(i, j) \in \mathbb{R}^{d_l \times d_l}$ , which is specific to each spatial location  $(i, j)$ , just as the middle part of Fig. 1:

$$\frac{d\mathbf{z}_t^{(l)}(i, j)}{dt} = \mathbf{A}^{(l)}(i, j) \mathbf{z}_t^{(l)}(i, j), \quad \forall(i, j), \quad t \in [\epsilon, T]. \quad (3)$$

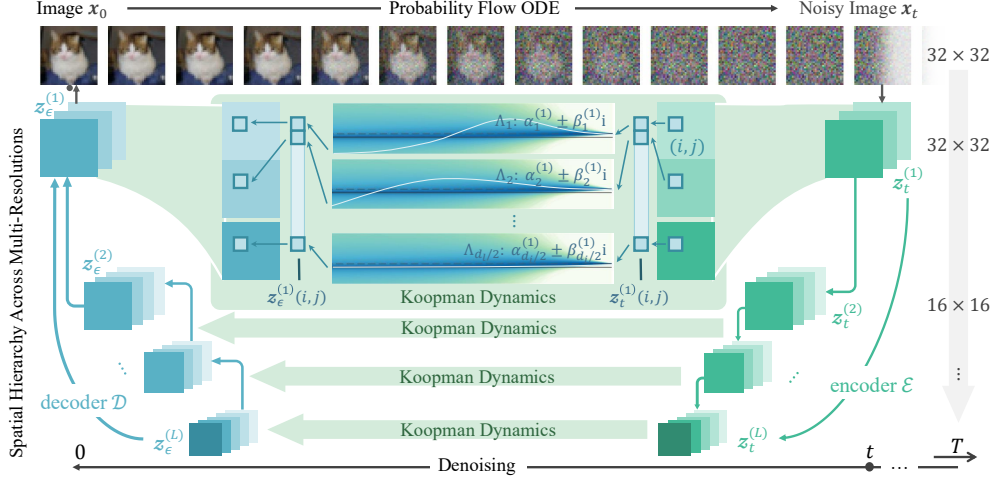


Figure 1: The framework of the proposed method. The HKD model first hierarchically extracts different-level features from the given noisy image at any time  $t$  by **encoder  $\mathcal{E}$** . Secondly, the **Koopman dynamics** model is applied for each level to the skips and the bottleneck. Last, a **uniform decoder  $\mathcal{D}$**  performs the mapping from the Koopman spaces back to the image space.

The latent evolution at each position thus follows an independent linear dynamical system governed by its local Koopman operator. Allowing  $\mathbf{A}^{(l)}(i, j)$  to vary across spatial locations enables the model to capture heterogeneous temporal-frequency behaviors at different regions, thus leading to finer-grained and spatially adaptive dynamics modeling.

Without loss of generality,  $\mathbf{A}^{(l)}(i, j)$  could be modeled in the form of block-diagonalizable, i.e.,

$$\mathbf{A}^{(l)}(i, j) = \text{diag}\{\mathbf{\Lambda}_1^{(l)}(i, j), \mathbf{\Lambda}_2^{(l)}(i, j), \dots, \mathbf{\Lambda}_k^{(l)}(i, j), \dots, \mathbf{\Lambda}_{d_l/2}^{(l)}(i, j)\} \quad (4)$$

with  $d_l$  even and each block  $\mathbf{\Lambda}_k^{(l)}(i, j) \in \mathbb{R}^{2 \times 2}$  corresponding to a pair of complex conjugate eigenvalues  $\alpha_k^{(l)} \pm i\beta_k^{(l)}$  of  $\mathbf{A}^{(l)}(i, j)$ , given explicitly by  $\mathbf{\Lambda}_k^{(l)}(i, j) = \begin{bmatrix} \alpha_k^{(l)}(i, j) & \beta_k^{(l)}(i, j) \\ -\beta_k^{(l)}(i, j) & \alpha_k^{(l)}(i, j) \end{bmatrix}$ . This reduces the number of parameters compared to modeling a full matrix. The rationality of this modeling can be guaranteed by the Prop. C.1, App. C.1. Due to the linear evolution in the Koopman space, the mapping between latent states at time  $s$  and  $t$  for level  $l$  and location  $(i, j)$  can be explicitly expressed as

$$\mathbf{z}_t^{(l)}(i, j) = e^{\mathbf{A}^{(l)}(i, j)(t-s)} \mathbf{z}_s^{(l)}(i, j). \quad (5)$$

The explicit formulation enables the network to learn sufficient one-step mappings from  $\mathbf{z}_t$ ,  $\forall t \in [\epsilon, T]$  to  $\mathbf{z}_\epsilon$  during the training process. Consequently, it allows direct mapping from  $\mathbf{w}_T$  to  $\mathbf{w}_\epsilon$  in the inference time, thus achieving efficient one-step sampling without the need for iterative integration.

**Decoder.** Finally, at the target time step  $\epsilon$ , the evolved set of evolved latent features  $\{\mathbf{z}_\epsilon^{(l)}\}_{l=1}^L$  is decoded to generate the sample  $\mathbf{x}_\epsilon$ . The decoding process is performed by a decoder  $\mathcal{D}\phi$ , which implements the mapping

$$\mathcal{D}\phi : \{\mathbf{z}_\epsilon^{(l)}\}_{l=1}^L \mapsto \mathbf{x}_\epsilon \in \mathbb{R}^{C \times H \times W}.$$

At each decoding level, the decoder upsamples the latent features from the lower-resolution level and integrates them with skip features obtained by evolving the encoder outputs through Koopman dynamics, just as the left part of Fig. 1.

### 3.2 Intermediate State Supervision: Trajectory Consistency Loss

A key advantage of our explicit framework over implicit mappings is the ability to supervise intermediate states during generation. Unlike conventional methods that supervise only the endpoint, our approach explicitly constrains the trajectory from noise to image by enforcing consistency with



Figure 2: Image reconstruction from noisy inputs at time  $t$  highlights the consistency enforced by our framework. The left-most image is obtained from  $x_T$  and the right-most is from  $x_0$ .

expected denoising dynamics along the path. Hence, we introduce a new trajectory consistency loss for the supervision of intermediate states. Fig. 2 presents the reconstructed images from noisy inputs at all time steps, showcasing the consistency properties promoted by our proposed loss function.

**Definition 3.1** (Trajectory Consistency Loss). *The trajectory consistency loss is defined as*

$$\mathcal{L}_{t\text{-consist}} \triangleq \mathbb{E}_{t \sim \mathcal{U}[\epsilon, T]} \left[ d \left( \mathcal{D}_\phi \left[ \{ e^{(\epsilon-t)\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(x_t) \}_{l=1}^L \right], x_\epsilon \right) \right], \quad (6)$$

where the expectation is taken with respect to  $t \sim \mathcal{U}[\epsilon, T]$ , and  $x_\epsilon$  is given by Eq. (1). Distance  $d(\cdot, \cdot) \geq 0$  is in the image space and equals 0 if and only if the argument variables are the same.

This trajectory consistency loss enforces that any intermediate state  $x_t$ , after being encoded and evolved through Koopman dynamics to time  $\epsilon$ , yields a decoded prediction that matches the ground truth image  $x_\epsilon$  obtained by Eq. (1). However, the most direct supervision would be to enforce alignment between the encoder-derived Koopman representation at time  $t$  and its analytically evolved counterpart from time  $T$  within the Koopman space. Although conceptually straightforward, enforcing consistency directly in the latent space may lead to suboptimal results in practice, as it reduces training flexibility and introduces gradients that poorly reflect perceptual discrepancies. Therefore, we instead adopt an alternative image-space formulation in Eq. (6), which enables the use of perceptually meaningful distance metrics, such as the Learned Perceptual Image Patch Similarity (LPIPS), to better capture semantic fidelity. Importantly, we show in App. C.2 that, under structural assumptions, minimizing the trajectory consistency loss in image space is theoretically equivalent to minimizing its latent-space counterpart.

### 3.3 Implementation

**Training.** We train the model using a total loss  $\mathcal{L} = \mathcal{L}_{t\text{-consist}} + \mathcal{L}_{\text{recon}}$ , where  $\mathcal{L}_{t\text{-consist}}$  enforces consistency along the diffusion trajectory, and  $\mathcal{L}_{\text{recon}} = d \left( \mathcal{D}_\psi \left( \{ e^{(\epsilon-T)\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(x_T) \}_{l=1}^L \right), x_\epsilon \right)$  supervises accurate one-step mapping from noise to clean image. The distance function is defined as  $d(x, y) = \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \mathcal{L}_{\text{LPIPS}}$ , with  $\lambda_2 = 1$  and  $\lambda_1$  annealed to shift from coarse alignment to perceptual refinement. The full model,  $\mathcal{E}_\theta$ ,  $\mathcal{D}_\phi$ , and  $\{\mathbf{A}^{(l)}\}_{l=1}^L$ , is trained end-to-end. The training algorithm is presented in Alg. 1. More training details are provided in App. B.1.

**One-Step Sampling.** At inference time, we could obtain the final generated sample  $\hat{x}_\epsilon$  from noise  $x_T$  by

$$\mathcal{D}_\phi \left( \{ e^{(\epsilon-T)\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(x_T) \}_{l=1}^L \right).$$

Specifically, Given a noise sample, we project it into the Koopman space, perform one-step evolution via Koopman dynamics, and decode the result back to the data space. The detailed sampling algorithm of sampling is presented in Alg. 2.

**Koopman Spectral Analysis.** Our framework offers a distinct perspective on diffusion generation through Koopman spectral analysis. Each block  $\mathbf{A}_k^{(l)}(i, j)$  in Eq. (4) represents a local dynamical mode, with eigenvalues  $\alpha_k^{(l)}(i, j) \pm i\beta_k^{(l)}(i, j)$  encoding the growth/decay and oscillation of the trajectory. This explicit spectral interpretability further allows for controllable image generation by selectively modifying certain frequency components. More details of Koopman spectral analysis of our framework can be found in App. B.2.

---

**Algorithm 1** The training algorithm.

---

- 1: **Inputs:** Trajectory  $\{\mathbf{x}_t\}_{t \in [0, T]}$  from a well-trained diffusion model; The number of samples  $s$
  - 2: **Outputs:** Network parameters  $\theta$  and  $\phi$ ; Koopman matrices  $\{\mathbf{A}^{(l)}\}_{l=1}^L$
  - 3: Initialize  $\theta$  and  $\phi$  by a pre-trained U-Net
  - 4: Initialize  $\mathbf{A}^{(l)} \leftarrow \mathbf{O}$ ,  $l = 1, \dots, L$
  - 5: **for**  $i = 0$  **to** num\_iter-1 **do**
  - 6:   Sample  $S_t = \{t_i \mid t_i \sim \mathcal{U}[0, T]\}_{i=1}^{s-1} \cup \{T\}$  ▷ Uniformly sample the intermediate time
  - 7:   Reconstruct  $\hat{\mathbf{x}}_\varepsilon = \mathcal{D}_\phi(\{e^{(\epsilon-t)\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(\mathbf{x}_t)\}_{l=1}^L)$  for  $t \in S_t$  ▷ Apply the HKD model
  - 8:   Let  $\mathcal{L} = \sum_{t \in S_t} [\|\hat{\mathbf{x}}_\varepsilon - \mathbf{x}_0\| + \|\mathcal{F}(\hat{\mathbf{x}}_\varepsilon) - \mathcal{F}(\mathbf{x}_0)\|]$  ▷  $\mathcal{F}$  is the feature extractor in LPIPS
  - 9:   Update  $\theta$ ,  $\phi$  and  $\mathbf{A}^{(l)}$  by the gradients of  $\mathcal{L}$
  - 10: **end for**
  - 11: **Return**  $\theta$ ,  $\phi$  and  $\{\mathbf{A}^{(l)}\}_{l=1}^L$
- 

---

**Algorithm 2** The sampling algorithm.

---

- 1: **Inputs:** Trained HKD including network parameters  $\theta$ ,  $\phi$  and Koopman matrices  $\{\mathbf{A}^{(l)}\}_{l=1}^L$
  - 2: **Outputs:** Predicted image  $\hat{\mathbf{x}}_\varepsilon = \mathcal{D}_\phi(\{e^{(\epsilon-T)\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(\mathbf{x}_T)\}_{l=1}^L)$
- 

### 3.4 Theoretical Results

In order to illustrate the feasibility of our proposed method, we theoretically analyze the error bounds caused by the proposed method, namely  $err_{\text{HKD}}$ , and by conventional one-step methods  $err_{\text{one-step}}$ . Intuitively, neural networks of the same size are faced with larger errors when estimating more complex functions. In the proposed formulation, the encoder and decoder are commonly simple due to the elementary observable function, hence, the HKD generation, serving as a composition of three functions, is overall simpler than end-to-end one-step methods. Algebraically speaking, we have  $err_{\text{HKD}} \leq err_{\text{one-step}} + O(\kappa)$  where  $\kappa$  is a reasonably small error caused by the Koopman process.

This inequality holds mainly because the proposed framework first transfers the complicated data space to the simpler Koopman space at a small cost and then controls the Koopman trajectory error by the explicit formulation in the Koopman theory. Thm. 3.1 provides an informal description of the conclusion where  $\kappa = O(N^{-\frac{1}{2}}) + o(m^{-\frac{r}{3}})$  is small and that substantiates the common knowledge with  $N$  and  $m$  being the size of the dataset and the dimension of the Koopman space respectively. A formal description and proof of the theorem are given in App. C.3.

**Theorem 3.1** (Comparison of Error Bounds for One-Step Diffusions). *Let  $err_{\text{HKD}}$  and  $err_{\text{one-step}}$  be the ideal estimation errors of our proposed HKD model and an end-to-end one-step model with the same total number of activation functions we have,*

$$err_{\text{HKD}} \leq err_{\text{one-step}} + O(\kappa),$$

where  $\kappa = \max \left\{ \frac{2\sqrt{3}\sigma m^2 \rho_{\text{sup}}}{\sqrt{N(\frac{\delta}{3} - 2m^{-\frac{r}{3}}) - \sqrt{3}\sigma m^2}}, \frac{2\sqrt{3}\sigma m^2 \rho_{\text{inf}}}{\sqrt{N(\frac{\delta}{3} - 2m^{-\frac{r}{3}}) \cdot \rho_{\text{inf}} - \sqrt{3}\sigma m^2}} \right\} + o(m^{-\frac{r}{3}})$  with  $N$  and  $m$  dominating  $\kappa$  being the size of the dataset and the dimension of the Koopman space respectively. The parameter  $\sigma$  is the standard deviation of the observables,  $\delta$  is 0 when the network is perfectly trained,  $\rho_{\text{inf}}$  and  $\rho_{\text{sup}}$  are the true spectral range of the Koopman operator, and  $r \rightarrow -\infty$  when the  $m$ -dimensional Koopman process adequately model the Koopman space.

To formally prove the above result, we first proposed to evaluate the network size by the number of activation functions it uses, as they are the ones causing non-linearity of the network. Secondly, we introduced the simplicial error as the deviation in the simplicial complex estimation. Fig. 3 empirically shows that the metric of same-density triangulations reveals the complexity of 2D surfaces, demonstrating our idea. Consequently, the simplicial error quantitatively evaluates the complexity of the data and noise spaces and the mappings between them. Finally, using the ideas above, we analytically prove Thm. 3.1.

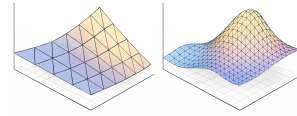


Figure 3: More complex surfaces need denser triangulation for the same error.

## 4 Experiments

We conducted experiments to present the promising one-step generation capability of our framework compared with other one-step baseline methods (Sec. 4.1). Notably, beyond generating high-quality images, we further provide empirical insights toward understanding the underlying dynamics of diffusion models through Koopman spectral analysis (Sec. 4.2). This analysis demonstrates the dynamical interpretability of our approach and further enables controllable image editing (Sec. 4.3). An ablation study was finally performed to show the key contributions of our framework (Sec. 4.4). Extensive results highlight both the generation quality and interpretability advantages of our framework, distinguishing it from distillation- and consistency-model-based paradigms.

### 4.1 Compare with Prior One-step Generation Methods

In this section, we evaluated our proposed framework on the CIFAR-10 [12] and FFHQ datasets [9], focusing exclusively on one-step generation methods within the diffusion model family. We used the standard Fréchet Inception Distance (FID) [6] metric and report FID-50k scores, following prior work [42]. Our model employs the U-Net [26] architecture from EDM [8] as the backbone for both the encoder and decoder, initialized with pretrained weights. All models were trained using the Adam optimizer [11] with a constant learning rate of  $1 \times 10^{-3}$  and a weight decay of 0.95. App. D.4 provides additional implementation details of the experiments.

Table 1: Sample quality on CIFAR-10 dataset.

| Methods                            | NFE(↓) | FID(↓) |
|------------------------------------|--------|--------|
| <b>Multi-Step Diffusion Models</b> |        |        |
| DDPM [7]                           | 1000   | 3.17   |
| Score SDE [33]                     | 2000   | 2.38   |
| DDIM [30]                          | 100    | 4.16   |
| DDIM [30]                          | 10     | 13.36  |
| EDM [8]                            | 35     | 1.97   |
| EDM [8]                            | 15     | 5.62   |
| <b>Diffusion Distillation</b>      |        |        |
| KD [17]                            | 1      | 9.36   |
| PD [27]                            | 1      | 8.34   |
| CD (LPIPS) [32]                    | 1      | 3.55   |
| DMD [39]                           | 1      | 3.77   |
| 1-Rectified flow [14]              | 1      | 6.18   |
| 2-Rectified flow [14]              | 1      | 4.85   |
| 3-Rectified flow [14]              | 1      | 5.21   |
| 2-Rectified flow++ [13]            | 1      | 3.38   |
| <b>Consistency Model</b>           |        |        |
| CT (LPIPS) [32]                    | 1      | 8.70   |
| CD (LPIPS) [32]                    | 1      | 3.55   |
| iCT [31]                           | 1      | 2.83   |
| iCT-deep [31]                      | 1      | 2.51   |
| ECM [4]                            | 1      | 3.60   |
| HKD                                | 1      | 3.30   |

improved versions. (2) **ECM** [4]: Easy Consistency Models, which initializes the network weights with the ones from a pretrained score model. More details of comparison methods are in App. D.5.

**Results.** While enhanced methods such as iCT-deep [31] have pushed performance boundaries, they suffer from high sensitivity to hyperparameters. Moreover, state-of-the-art consistency training typically requires nearly a week of training on 8 GPUs [4], and remains unstable in practice [16]. Retraining with consistency distillation (CD) even results in a degraded FID of 10.53, as reported

**Comparison to Prior Work.** We evaluated our framework against two dominant one-step generation approaches: distillation-based methods and consistency models, with comparisons to multi-step diffusion baselines for context. The multi-step diffusion baselines include: (1) **DDPM** [7]: The foundational denoising diffusion model; (2) **Score SDE** [33]: Continuous-time diffusion via reverse-time SDEs; (3) **DDIM** [30]: Efficient deterministic sampler from diffusion ODEs; (4) **EDM** [8]: State-of-the-art diffusion backbone (used in our framework).

For the distillation-based one-step paradigm, the comparison extends to traditional leading methods: (1) **KD** [17]: Traditional knowledge distillation method with direct teacher-student mimicry; (2) **PD** [27]: Progressive distillation with step reduction through sequential student-teacher training; (3) **CD (LPIPS)** [32]: Consistency distillation with LPIPS loss; (4) **DMD** [39]: Distribution matching distillation with the distribution-matching loss; (5) and **(distilled) ReFlow** series: including 1-Rectified flow, 2-Rectified flow, 3-Rectified flow [14] and 2-Rectified flow++ [13] with reflow and distillation. For the consistency-model-based method, we included: (1) **CT** [32]: Consistency training including CT (LPIPS), iCT [31], and iCT-deep [31], which provides a comparison to the end-to-end traditional consistency model, and its im-



Figure 4: Visualization of Spectral Contributions in Generated Images Across Koopman Modes. (a) visualizes the contribution of Koopman spectrum with the smallest real-part magnitudes to the reconstructed state over time. It corresponds to the noisy part of the image. (b), (c) and (d) illustrate progressively larger spectral components and their reconstructed image components.

in [10]. In contrast, our method achieved comparable performance within just 2–3 days on 8×V100 GPUs.

Moreover, we significantly improved the training stability thanks to two design choices: (1) the exponential formulation in the Koopman space ensures sufficiently large gradients for spectra with magnitudes near 1, mitigating the high-variance gradient issues commonly seen in consistency models [28]; and (2) supervising the Koopman trajectory at multiple time points enables more stable and averaged spectrum estimation, leading to more reliable mode evolution modeling. Experimental evidence of the improved training stability on the CIFAR-10 dataset is provided in App. D.1.

Table 2: Sample quality on FFHQ.

| Methods   | NFE(↓) | FID(↓) |
|-----------|--------|--------|
| DDIM [30] | 10     | 18.30  |
| EDM [8]   | 79     | 2.47   |
| EDM [8]   | 15     | 9.85   |
| ECM [4]   | 1      | 5.99   |
| HKD       | 1      | 5.70   |

Additional experiments on FFHQ 64x64 (Tab. 2) validated the effectiveness of our method on the high-resolution and structurally complex dataset. See App. D.1 for additional visualizations, per-image wall-clock timings for various generation methods, and conditional generation results.

## 4.2 Koopman Spectral Analysis for Generative Process

We conducted an analysis of  $\mathbf{A}$  to further investigate how individual spectral components influence the generative trajectory. We first tracked the contribution of each Koopman spectrum (cumulative effect, namely CE) to the reconstruction over time. Then, at each resolution level  $l$ , we sorted the eigenvalue pairs across spatial locations by the real parts and applied spectral masking by retaining only those within a target range (smallest, intermediate, or largest), zeroing out the rest. The masked representations are decoded to visualize the contribution of selected spectral modes.

Fig. 4 summarizes the results of the two spectral analyses: the left shows the contribution of grouped Koopman modes to reconstruction over time, while the right links spectral ranges to corresponding image structures. By selectively activating spectral bands and decoding the latent representations, we observe a clear semantic hierarchy: low-range modes capture global structure, mid-range modes recover overall shape and pose, and high-range modes refine local details. These results demonstrate the semantic interpretability of Koopman spectra and their potential for controllable image synthesis. More Koopman spectral analysis results are provided in App. D.2.

## 4.3 One-Step Image Editing: A Case of Model Interpretability

We evaluated the interpretability of our framework through an image editing experiment that targets frequency-specific interventions at the intermediate state of the diffusion trajectory. We performed editing by injecting high-frequency features from a reference image into the generated image at mixing ratios of 10%, 20%, 50%, 80%, and 90%, applied at the middle time step to demonstrate

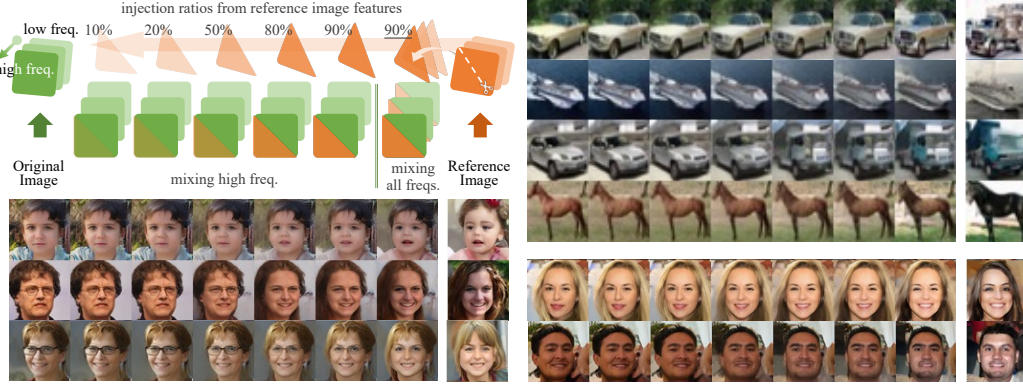


Figure 5: One-step image editing via frequency-aware interventions along the diffusion trajectory. We controlled the image generation by the high-frequency features from a reference image through injecting them into the lower-left half of the generating image at different mixing ratios (10%, 20%, 50%, 80%, 90%). The modifications were performed at the midpoint of the Koopman trajectory. Columns 2-6 showcase the frequency-aware editing, where only high-frequency components were mixed, preserving the low-frequency structure of the original image. Column 7 is for frequency-agnostic editing, where all-frequency features of reference images are mixed with all frequency bands of the original image at a mixing ratio of 90%. We exhibit the results from both datasets.

temporal editability. For comparison, frequency-agnostic editing mixes the full-spectrum content of the reference and generated images at the same step.

As shown in Fig. 5, increasing the injection ratio of reference image features from 10% to 90% gradually reveals more facial details from the reference, demonstrating that our frequency-aware editing establishes meaningful correspondences through interpretable frequency decomposition. In contrast, frequency-agnostic editing disrupts global structures, indicating a lack of disentangled control. These results further validate the interpretability of our framework.

We further perform an image editing experiment of image recuperation in the inpainting and coloring tasks on CIFAR-10 dataset following Algorithm 4 from [32], which iteratively mixes a reference image with the generated image along the generative trajectory by adding and removing noise at each time step  $t$ . The results are presented in Fig. 6.

#### 4.4 Ablation Study

We present an ablation study on the CIFAR-10 dataset to evaluate the contributions of key components in our framework: Koopman evolution (Koop.), the trajectory consistency loss ( $\mathcal{L}_{t\text{-consist}}$ ), and hierarchical design (hierar.). Tab. 3 shows that the one-step baseline model removing the Koopman dynamics results in an FID of 5.72. Upon it, additional Koopman evolutions at the skips and bottleneck paths, and introducing trajectory consistency loss further improve the results. In addition, a hierarchical Koopman dynamics design also contributes to better generation quality.

Table 3: Ablation study on CIFAR-10.

| Koop.        | Settings                         |              | FID ( $\downarrow$ ) |
|--------------|----------------------------------|--------------|----------------------|
|              | $\mathcal{L}_{t\text{-consist}}$ | hierar.      |                      |
| $\times$     | $\times$                         | $\checkmark$ | 5.72                 |
| $\checkmark$ | $\times$                         | $\checkmark$ | 5.57                 |
| $\checkmark$ | $\checkmark$                     | $\times$     | 4.78                 |
| $\checkmark$ | $\checkmark$                     | $\checkmark$ | 3.30                 |

## 5 Related Work

**One-step Generation for Diffusion Models.** Prior works have distilled pretrained diffusion models into efficient one-step generators. Knowledge distillation [17] trains a student model to replicate the denoising behavior of the teacher, rectified flows [14] reformulate diffusion as ODEs for distillation, all considered as classical approaches. Recent works further improve distillation-based generation, such as DMD [39], which aligns one-step generators with diffusion models via KL divergence, and SlimFlow, which [42] enhances rectified flows through compression of model size. Consistency



Figure 6: The visualization of image recuperation in the inpainting and coloring tasks. The experiment is performed by Algorithm 4 from [32], which iteratively mixes the image with a reference image created by adding and removing noise at time  $t$  with  $t$  decreasing from  $T$  to 0.

models [32] take a different route, suffering from instability and inefficiency. Improvements include fine-tuning from diffusion models [4] and adversarial enhancements [5]. However, these methods prioritize performance at the cost of losing the inherent interpretability and controllability offered by diffusion models.

**Koopman Operators.** Koopman operator theory has enabled advances in system analysis [19], control [23], and optimization [25] by extracting globally linear structures from nonlinear dynamics. Recent work extends its use to time series modeling [22, 40, 2]. In addition, many other works have proposed approximating the Koopman operators using neural networks. For instance, KNF [35] leverages DNNs to learn the linear Koopman space and the coefficients of chosen measurement functions. To our knowledge, we are the first to introduce Koopman theory into the field of image generation. See App. A.1 for more related work.

## 6 Discussion

HKD is an interpretable one-step generation framework that retains access to intermediate states and fine-grained control—while enabling one-step sampling. It is the first to introduce interpretability and intermediate-state intervention into one-step generation. By bridging fast sampling and interpretability, HKD opens new possibilities for explainable image synthesis. While our current formulation already demonstrates competitive one-step generation performance and enables spectral interventions, several directions remain open for future exploration. First, although we adopt a standard training paradigm without relying on adversarial optimization or heavy tuning, integrating advanced training techniques (e.g., adversarial learning) may further improve generation quality. In addition, while HKD performs well on standard-resolution datasets, its potential for high-resolution generation, afforded by its hierarchical design, remains underexplored. Moreover, the explicit spectral decomposition in our framework naturally supports interpretable interventions, making it well-suited for a range of semantic editing tasks. Beyond the frequency-aware manipulations we demonstrate, HKD opens up possibilities for text-guided editing, attribute-specific control. While these directions are currently unexplored, they underscore the broader applicability of our framework beyond pure generation. Our framework provides a solid foundation for these future advancements.

## Acknowledgments

We would like to thank the anonymous reviewers and area chairs for their helpful comments. This work was partially supported by the National Natural Science Foundation of China (No. 12471481, U24A2001), the Science and Technology Commission of Shanghai Municipality (No. 23ZR1403000), and the Open Foundation of Key Laboratory Advanced Manufacturing for Optical Systems, CAS (No. KLMSKF202403). We also acknowledge the support (to D. Zou) from NSFC 62306252, Hong Kong ECS award 27309624, Guangdong NSF 2024A1515012444, and the central fund from HKU IDS.

## References

- [1] Julius Aka, Johannes Brunnemann, Jörg Eiden, Arne Speerforck, and Lars Mikelsons. Balanced neural odes: nonlinear model order reduction and koopman operator approximations. *arXiv preprint arXiv:2410.10174*, 2024.
- [2] Hanru Bai and Weiyang Ding. KoNODE: Koopman-driven neural ordinary differential equations with evolving parameters for time series analysis. In *Forty-second International Conference on Machine Learning*, 2025.
- [3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [4] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.
- [5] Shelly Golan, Roy Ganz, and Michael Elad. Enhancing consistency-based image generation via adversarially-trained classification and energy-based discrimination. *arXiv preprint arXiv:2405.16260*, 2024.
- [6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [8] 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.
- [9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [10] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.
- [11] Diederik Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International conference on learning representations (ICLR)*, volume 5. California, 2015.
- [12] Alex Krizhevsky, Vinod Nair, Geoffrey Hinton, et al. The CIFAR-10 dataset. *online: <http://www.cs.toronto.edu/kriz/cifar.html>*, 55(5):2, 2014.
- [13] Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in Neural Information Processing Systems*, 37:63082–63109, 2024.
- [14] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.
- [15] Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with Koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.
- [16] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.
- [17] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.
- [18] Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *Advances in Neural Information Processing Systems*, 37:115377–115408, 2024.

- [19] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.
- [20] Alexandre Mauroy, Y Susuki, and Igor Mezic. *Koopman operator in systems and control*, volume 484. Springer, 2020.
- [21] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.
- [22] Ilan Naiman, N Benjamin Erichson, Pu Ren, Michael W Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via Koopman VAEs. *arXiv preprint arXiv:2310.02619*, 2023.
- [23] Abhinav Narasingam, Sang Hwan Son, and Joseph Sang-Il Kwon. Data-driven feedback stabilisation of nonlinear systems: Koopman-based model predictive control. *International Journal of Control*, 96(3):770–781, 2023.
- [24] Samuel E Otto and Clarence W Rowley. Koopman operators for estimation and control of dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):59–87, 2021.
- [25] William T Redman, Maria Fonoberova, Ryan Mohr, Ioannis G Kevrekidis, and Igor Mezic. An operator theoretic view on pruning deep neural networks. *arXiv preprint arXiv:2110.14856*, 2021.
- [26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [27] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- [28] Gianluigi Silvestri, Luca Ambrogioni, Chieh-Hsin Lai, Yuhta Takida, and Yuki Mitsufuji. Training consistency models with variational noise coupling. *arXiv preprint arXiv:2502.18197*, 2025.
- [29] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [31] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.
- [32] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023.
- [33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [34] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.
- [35] Rui Wang, Yihe Dong, Sercan O Arik, and Rose Yu. Koopman neural operator forecaster for time-series with temporal distributional shifts. In *The Eleventh International Conference on Learning Representations*, 2023.
- [36] Yuanchao Xu, Jing Liu, Zhongwei Shen, and Isao Ishikawa. Reinforced data-driven estimation for spectral properties of Koopman semigroup in stochastic dynamical systems. *arXiv preprint arXiv:2509.04265*, 2025.

- [37] Yuanchao Xu, Kaidi Shao, Isao Ishikawa, Yuka Hashimoto, Nikos Logothetis, and Zhongwei Shen. A data-driven framework for Koopman semigroup estimation in stochastic dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(10):103123, 10 2025.
- [38] Yuanchao Xu, Kaidi Shao, Nikos Logothetis, and Zhongwei Shen. ResKoopNet: Learning Koopman representations for complex dynamics with spectral residuals. *arXiv preprint arXiv:2501.00701*, 2025.
- [39] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.
- [40] Ronghua Zheng, Hanru Bai, and Weiyang Ding. KooNPro: A variance-aware Koopman probabilistic model enhanced by neural process for time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [41] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.
- [42] Yuanzhi Zhu, Xingchao Liu, and Qiang Liu. Slimflow: Training smaller one-step diffusion models with rectified flow. In *European Conference on Computer Vision*, pages 342–359. Springer, 2024.

## 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: Main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

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: In Section 6, we discuss the limitations and possible future works for this paper.

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 provide proofs for every theory and proposition in the 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 provide experimental settings and training details in Appendix D.

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: After organizing the code, we will release the code to support full reproducibility. Details for reproducing results are described in the Appendix.

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 provide experimental settings and training details in Appendix D.

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: Statistical significance of the spectrum is analyzed and reported in the Koopman spectral analysis section.

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 provide experimental settings and training details in Appendix D. We also give the information on computer resources in Sec. 4.1.

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: The research conducted in the paper conforms, in every respect, with the 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: We provide the broader impacts in the appendix.

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: [NA]

Justification: The models and datasets used are standard and pose no specific high-risk misuse scenario. No new models with high dual-use potential are released.

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: 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.

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](https://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: We do 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: The 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: The 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.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## Appendix Overview

The following lists the structure of the appendix, with links to the respective sections.

### A Background Knowledge Supplement:

[A.1 Related Work](#)

[A.2 Koopman Operator: From Infinite-Dimensional to Finite-Dimensional Approximation](#)

### B Framework Supplement:

[B.1 Training Details](#)

[B.2 Koopman Spectral Analysis](#)

### C Theoretical Supplement:

[C.1 Proposition: Rationality of the Block Diagonalization of  \$\mathbf{A}\$](#)

[C.2 Proposition: Image-Space Trajectory Consistency Loss and its Latent-Space Counterpart](#)

[C.3 Formal Statement and Proof of Theorem 3.1](#)

### D Experimental Supplement:

[D.1 Additional Experimental Results for Sec. 4.1](#)

[D.1 Conditional Generation on CIFAR-10](#)

[D.1 Visual Results](#)

[D.1 Training Stability Results](#)

[D.1 The Wall-clock Time Per Image for Generation Methods](#)

[D.2 Additional Experimental Results for Sec. 4.2](#)

[D.3 Additional Experimental Results for Sec. 4.3](#)

[D.4 More Implementation Details about Experiments](#)

[D.5 More Details about Comparison Methods](#)

### E Broader Impact

## A Background Knowledge Supplement

### A.1 Related Work

**One-step Generation for Diffusion Models.** Several prior works have explored distilling pretrained diffusion models into efficient one-step generators. Knowledge distillation [17] trains a student model to mimic the full sampling trajectory of a pretrained diffusion model. Rectified flows [14] reformulate diffusion sampling as a continuous normalizing flow, allowing distillation via ODE simulation. Progressive distillation [27] gradually reduces the number of inference steps during training, enabling faster sampling with minimal performance loss. Recent works further improve distillation-based generation, such as SiD [41], which reformulates forward diffusion processes as semi-implicit distributions, DMD [39], which enforces the one-step image generator to match the diffusion model at the distribution level by minimizing an approximate KL divergence, and SIM [18], which computes the gradients for a wide class of score-based divergences between a diffusion model and a generator. In parallel, SlimFlow [42] builds on rectified flows by exploring joint compression of inference steps and model size to enhance efficiency. Consistency models [32] offer an alternative approach, learning to map noisy inputs directly to clean outputs. These models can either be trained end-to-end or used as distillation students. Due to their training instability and inefficiency, various improvements have been proposed, such as [4], which fine-tunes a consistency model starting from a pretrained diffusion model, [28], which trains consistency models with variational noise coupling, and [5] by enhancing consistency models via adversarially-trained classification and energy-based discrimination. However, most of these works ignore the interpretability of the generation process.

**Koopman Operators.** Over the past two decades, Koopman operator theory has attracted growing interest, enabling progress in dynamical system analysis [36, 38, 37], control [23], optimization [25], and forecasting [15]. These methods utilize Koopman-based representations to extract globally linearizable structures from nonlinear dynamics, improving interpretability and control. More recently, Koopman operators have been increasingly applied to time series modeling [22, 40]. In addition, several works have leveraged neural networks to approximate Koopman operators. For example, [1] proposes balanced neural ODEs to approximate Koopman operators without predefined dimensionality, while [?] uses DNNs to learn both the Koopman space and coefficients of selected observables. In our work, we similarly employ neural networks to learn observable functions for end-to-end training. To the best of our knowledge, we are the first to introduce Koopman theory into image generation.

### A.2 Koopman Operator: From Infinite-Dimensional to Finite-Dimensional Approximation

In theory, the Koopman operator  $\mathcal{K}$  is an infinite-dimensional linear operator that governs the evolution of observable functions under a nonlinear dynamical system. That is, for a dynamical map  $\Phi : \mathcal{X} \rightarrow \mathcal{X}$  and any observable function  $g : \mathcal{X} \rightarrow \mathbb{R}$ , the Koopman operator acts as

$$(\mathcal{K}g)(x) = g(\Phi(x)). \quad (7)$$

However, this operator acts on an infinite-dimensional function space, which is not directly tractable in practice. To enable computation, the Koopman operator is commonly approximated in a finite-dimensional subspace. This is achieved by selecting a finite set of basis functions  $\mathbf{u} = [u_1, u_2, \dots, u_m]^\top$  to represent the space of observables. We then approximate the action of  $\mathcal{K}$  on this basis as

$$\mathcal{K}\mathbf{u}(x) \approx \mathbf{K}\mathbf{u}(x), \quad (8)$$

where  $\mathbf{K} \in \mathbb{R}^{m \times m}$  is a finite-dimensional matrix known as the Koopman matrix. Given an observable  $g$  projected onto the subspace as  $\boldsymbol{\xi} \triangleq [\langle g, u_1 \rangle, \langle g, u_2 \rangle, \dots, \langle g, u_m \rangle]^\top$ , its evolution under  $\Phi$  can be approximated by  $g(\Phi(x)) \approx \boldsymbol{\xi}^\top \mathbf{K}\mathbf{u}(x)$ . Furthermore, if we consider a vector of observables  $\mathbf{g} \approx \mathbf{P}\mathbf{u}$  that is linearly related to the basis via some invertible matrix  $\mathbf{P}$ , the Koopman dynamics in this new coordinate system are given by the similarity transform:

$$\mathbf{g}(\Phi(x)) \approx \mathbf{P}\mathbf{K}\mathbf{P}^{-1}\mathbf{g}(x). \quad (9)$$

Several numerical algorithms have been developed to compute the Koopman matrix  $\mathbf{K}$  from data:

(1) Dynamic Mode Decomposition (DMD): DMD is a widely used data-driven technique that approximates the Koopman operator using linear regression on snapshots of system states. It assumes

a linear relationship between time-shifted observables and solves for the best-fit linear operator  $\mathbf{K}$  such that  $\mathbf{x}_{t+1} \approx \mathbf{K} \mathbf{x}_t$  in the observable space. Variants include Extended DMD (EDMD) and Hankel-DMD for richer function spaces.

(2) Extended Dynamic Mode Decomposition (EDMD): EDMD generalizes DMD by applying the method in a lifted feature space defined by nonlinear basis functions (e.g., polynomials, radial basis functions). This corresponds to choosing a specific  $\mathbf{u}$  and solving for the best Koopman matrix in this subspace.

(3) Neural Approximations: More recently, neural networks have been used to learn either the Koopman embedding (i.e., the basis functions  $\mathbf{u}$ ) or the Neural Approximations use trainable architectures to discover a latent space where linear evolution via a Koopman operator holds approximately. They enable scalable and adaptive modeling of complex, high-dimensional dynamics. In our work, we adopt this method to approximate the Koopman operators.

## B Framework Supplement

### B.1 Training Details

We optimize our model using a total loss that combines the trajectory consistency loss  $\mathcal{L}_{t\text{-consist}}$  that supervises intermediate states along the diffusion trajectory with a reconstruction loss

$$\mathcal{L}_{\text{recon}} = d\left(\mathcal{D}_{\psi}(\{\mathbf{e}^{(\epsilon-T)\mathbf{A}^{(l)}} \mathcal{E}_{\theta}^{(l)}(\mathbf{x}_T)\}_{l=1}^L), \mathbf{x}_{\epsilon}\right)$$

from the noise  $\mathbf{x}_T$  to enforce the accurate one-step mapping from noise to clean image, i.e.,

$$\mathcal{L} = L_{t\text{-consist}} + L_{\text{recon}}.$$

Define that  $\mathbf{x}_{\epsilon}^t \triangleq \mathcal{D}_{\psi}(\{\mathbf{e}^{(\epsilon-t)\mathbf{A}^{(l)}} \mathcal{E}_{\theta}^{(l)}(\mathbf{x}_T)\}_{l=1}^L)$ , then  $\mathcal{L}_{\text{total}} = \sum_{t \in \{T\} \cup \mathcal{T}} d(\mathbf{x}_{\epsilon}^t, \mathbf{x}_{\epsilon})$ , where the elements of  $\mathcal{T}$  are sampled from  $U[\epsilon, T)$  and  $d(\mathbf{x}, \mathbf{y})$  is a metric composed of  $l_2$ -distance, i.e., MSE loss, and LPIPS loss in our model:

$$d(\mathbf{x}, \mathbf{y}) = \lambda_1 \mathcal{L}_{\text{MSE}}(\mathbf{x}, \mathbf{y}) + \lambda_2 \mathcal{L}_{\text{LPIPS}}(\mathbf{x}, \mathbf{y}).$$

To balance perceptual fidelity and optimization stability, we fix  $\lambda_2 = 1$  and apply an annealing schedule to  $\lambda_1$ , allowing the model to initially focus on coarse alignment before gradually emphasizing perceptual refinement.

For network architecture, we adopt the mature U-Net encoder and decoder backbone used in diffusion models for both the encoder and decoder modules in our formulation to leverage existing diffusion models' design efficacy. To improve training efficiency and reduce mode collapse, we further initialize the encoder and decoder with pre-trained weights from diffusion models, which provide structured latent-to-output mappings, and retain rich hierarchical features acquired during large-scale training. The entire model, including  $\mathcal{E}_{\theta}$ ,  $\mathcal{D}_{\phi}$ , and  $\{\mathbf{A}^{(l)}\}_{l=1}^L$ , is trained end-to-end using the loss  $\mathcal{L}_{\text{total}}$ .

### B.2 Koopman Spectral Analysis: A New Lens on Interpreting Diffusion Dynamics

Our framework offers a fundamentally different viewpoint on the analysis of the diffusion generation process via Koopman spectral analysis. Beyond enabling one-step sampling, the proposed formulation naturally provides a novel dynamical perspective of diffusion-based generative models through the lens of Koopman spectral analysis. By explicitly modeling the latent evolution as a linear dynamical system, our approach allows for the application of spectral tools from dynamical systems theory to reveal the underlying spectral structure of generative dynamics and provide new theoretical tools for analyzing and interpreting dynamical modes during sampling.

Specifically, each block  $\Lambda_k^{(l)}(i, j)$  directly characterizes a local dynamical mode, with eigenvalues  $\alpha_k^{(l)}(i, j) \pm i\beta_k^{(l)}(i, j)$  encoding the growth/decay rate and oscillatory frequency of the diffusion trajectories. The eigenvalues of Koopman matrix  $\mathbf{K}^{(l)}(i, j)$  associated with each mode can be computed as

$$\lambda_k\left(\mathbf{K}^{(l)}(i, j)\right) = e^{\alpha_k^{(l)}(i, j)\Delta t} \left[ \cos\left(\beta_k^{(l)}(i, j)\Delta t\right) \pm i \sin\left(\beta_k^{(l)}(i, j)\Delta t\right) \right], \quad (10)$$

providing a principled spectral description of the generative dynamics. The magnitude  $|\lambda_k(\mathbf{K}^{(l)}(i, j))| = e^{\alpha_k^{(l)}(i, j)\Delta t}$  determines the stability of each mode, while the imaginary part of  $\lambda_k(\mathbf{K}^{(l)}(i, j))$  captures the frequency modes of oscillations. Benefiting from the block-diagonal structure of  $\mathbf{A}^{(l)}(i, j)$ , our framework enables direct analysis of its spectral components to interpret the generative behavior of the model by directly observing the values of  $\alpha_k^{(l)}(i, j)$  and  $\beta_k^{(l)}(i, j)$ . Specifically, the high real parts  $\alpha_k^{(l)}(i, j)$  and imaginary parts  $\beta_k^{(l)}(i, j)$  and low  $\alpha_k^{(l)}(i, j)$  and  $\beta_k^{(l)}(i, j)$  of each block correspond to low- and high-frequency modes in the generation process, respectively.

This explicit spectral interpretability further allows for controllable image editing by selectively modifying certain frequency components, enabling targeted modification of structural or fine-grained details in the generated images.

## C Theoretical Supplement

### C.1 Proposition: Rationality of the Block Diagonalization of $\mathbf{A}$

We present the following proposition to ensure the rationality of the block diagonalization of the linear operator  $\mathbf{A}$ .

**Proposition C.1.** *We define the set for the  $l$ -layer encoders  $S_{\mathcal{E}} \triangleq \{\mathcal{E} = \mathcal{E}_l \circ \dots \circ \mathcal{E}_2 \circ \mathcal{E}_1 \mid \mathcal{E}_l : \mathbf{f} \mapsto \mathbf{W}_l^{(\mathcal{E})} \mathbf{f} + \mathbf{b}_l^{(\mathcal{E})}\}$  and the set for the decoders  $S_{\mathcal{D}} \triangleq \{\mathcal{D} = \mathcal{D}_l \circ \dots \circ \mathcal{D}_2 \circ \mathcal{D}_1 \mid \mathcal{D}_1 : \mathbf{x} \mapsto \mathbf{W}_1^{(\mathcal{D})} \mathbf{x} + \mathbf{b}_1^{(\mathcal{D})}\}$ . Then for  $\mathcal{E} \in S_{\mathcal{E}}, S_{\mathcal{D}} \in S_{\mathcal{D}}$  and the Koopman matrix  $\mathbf{K}$ , there exists another set of  $\tilde{\mathcal{E}} \in S_{\mathcal{E}}, \tilde{\mathcal{D}} \in S_{\mathcal{D}}$  and  $\tilde{\mathbf{K}} \in \mathbb{R}^{m \times m}$  that  $\tilde{\mathbf{K}} = \mathbf{I} + \text{diag}\{\Lambda_1, \Lambda_2, \dots, \Lambda_k\}$  and  $\Lambda_i = \begin{bmatrix} \alpha_i & \beta_i \\ -\beta_i & \alpha_i \end{bmatrix}$ , such that,*

$$\mathcal{D}e^{-t(\mathbf{K}-\mathbf{I})}\mathcal{E} \equiv \tilde{\mathcal{D}}e^{-t(\tilde{\mathbf{K}}-\mathbf{I})}\tilde{\mathcal{E}}. \quad (11)$$

*Proof.* We assume that the eigenvalue decomposition of matrix  $\mathbf{K}$  in the complex field is  $\mathbf{K} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$  where  $\mathbf{\Lambda} \triangleq \text{diag}\{\lambda_1, \bar{\lambda}_1, \dots, \lambda_r, \bar{\lambda}_r, \lambda_{r+1}, \lambda_{r+2}, \dots, \lambda_{m-r}\}$  with  $\lambda_i, \bar{\lambda}_i, 1 \leq i \leq r$  the conjugated imaginary eigenvalues and  $\lambda_i, r+1 \leq i \leq m-r$  the real eigenvalues.

Let  $\Lambda_i = \begin{bmatrix} \text{Re } \lambda_i - 1 & \text{Im } \lambda_i \\ -\text{Im } \lambda_i & \text{Re } \lambda_i - 1 \end{bmatrix}, 1 \leq i \leq r$  and  $\Lambda_i = \begin{bmatrix} \lambda_i - 1 & 0 \\ 0 & \lambda_i - 1 \end{bmatrix}, r+1 \leq i \leq m-r$ , we use them to define  $\tilde{\mathbf{K}}$ . Furthermore, let  $\tilde{\mathbf{P}} \triangleq [\text{Re } p_1, \text{Im } p_1, \dots, \text{Re } p_r, \text{Im } p_r, \frac{p_{r+1}}{2}, \frac{p_{r+1}}{2}, \dots, \frac{p_{m-r}}{2}, \frac{p_{m-r}}{2}]$  where  $\mathbf{P} = [p_1, \bar{p}_1, \dots, p_r, \bar{p}_r, p_{r+1}, p_{r+2}, \dots, p_{m-r}]$  and  $\tilde{\mathbf{Q}} = \sqrt{2} [\text{Re } p_1^\dagger, -\text{Im } p_1^\dagger, \dots, \text{Re } p_r^\dagger, -\text{Im } p_r^\dagger, \frac{p_{r+1}^\dagger}{2}, \frac{p_{r+1}^\dagger}{2}, \dots, \frac{p_{m-r}^\dagger}{2}, \frac{p_{m-r}^\dagger}{2}]^\top$  where  $\mathbf{P}^{-1} = [p_1^\dagger, \bar{p}_1^\dagger, \dots, p_r^\dagger, \bar{p}_r^\dagger, p_{r+1}^\dagger, p_{r+2}^\dagger, \dots, p_{m-r}^\dagger]^\top$ . Note that  $p_i^\dagger$ 's are row vectors and  $\tilde{\mathbf{P}}\tilde{\mathbf{Q}}$  is NOT necessarily identity.

Last but not least, let  $\tilde{\mathbf{W}}_l^{(\mathcal{E})} \triangleq \tilde{\mathbf{Q}}\mathbf{W}_l^{(\mathcal{E})}$  and  $\tilde{\mathbf{b}}_l^{(\mathcal{E})} \triangleq \tilde{\mathbf{Q}}\mathbf{b}_l^{(\mathcal{E})}$  in  $\tilde{\mathcal{E}}$  and  $\tilde{\mathbf{W}}_1^{(\mathcal{D})} \triangleq \mathbf{W}_1^{(\mathcal{D})}\tilde{\mathbf{P}}$  in  $\tilde{\mathcal{D}}$  then

$$\begin{aligned} & \tilde{\mathcal{D}}e^{-t(\tilde{\mathbf{K}}-\mathbf{I})}\tilde{\mathcal{E}}(\mathbf{x}), \\ &= \tilde{\mathcal{D}}_l \circ \dots \circ \tilde{\mathcal{D}}_2(\tilde{\mathbf{W}}_1^{(\mathcal{D})}e^{-t(\tilde{\mathbf{K}}-\mathbf{I})}(\tilde{\mathbf{W}}_l^{(\mathcal{E})}\tilde{\mathcal{E}}_{l-1} \circ \dots \circ \tilde{\mathcal{E}}_1(\mathbf{x}) + \tilde{\mathbf{b}}_l^{(\mathcal{E})}) + \tilde{\mathbf{b}}_1^{(\mathcal{D})}), \\ &= \mathcal{D}_l \circ \dots \circ \mathcal{D}_2(\mathbf{W}_1^{(\mathcal{D})}\tilde{\mathbf{P}}e^{-t(\tilde{\mathbf{K}}-\mathbf{I})}\tilde{\mathbf{Q}}(\mathbf{W}_l^{(\mathcal{E})}\mathcal{E}_{l-1} \circ \dots \circ \mathcal{E}_1(\mathbf{x}) + \mathbf{b}_l^{(\mathcal{E})}) + \mathbf{b}_1^{(\mathcal{D})}), \\ &= \mathcal{D}[\tilde{\mathbf{P}}e^{-t(\tilde{\mathbf{K}}-\mathbf{I})}\tilde{\mathbf{Q}}\mathcal{E}(\mathbf{x})], \\ &\stackrel{*}{=} \mathcal{D}e^{-t(\mathbf{K}-\mathbf{I})}\mathcal{E}(\mathbf{x}), \end{aligned}$$

where  $\stackrel{*}{=}$  holds due to  $\begin{bmatrix} \frac{p_i}{\sqrt{2}} & \frac{p_i}{\sqrt{2}} \end{bmatrix} e^{-t\Lambda_i} \begin{bmatrix} p_i^\dagger/\sqrt{2} \\ p_i^\dagger/\sqrt{2} \end{bmatrix} = e^{-t(\lambda_i-1)}p_i p_i^\dagger, r+1 \leq i \leq m-r$  and

$$\begin{bmatrix} \sqrt{2}\text{Re } p_i & \sqrt{2}\text{Im } p_i \end{bmatrix} e^{-t\Lambda_i} \begin{bmatrix} \sqrt{2}\text{Re } p_i^\dagger \\ -\sqrt{2}\text{Im } p_i^\dagger \end{bmatrix}, \quad (12)$$

$$\frac{\lambda_i = \alpha_i + i\beta_i}{2} 2 \begin{bmatrix} \text{Re } p_i & \text{Im } p_i \end{bmatrix} \begin{bmatrix} e^{-t(\alpha_i-1)} \cos(-t\beta_i) & e^{-t(\alpha_i-1)} \sin(-t\beta_i) \\ -e^{-t(\alpha_i-1)} \sin(-t\beta_i) & e^{-t(\alpha_i-1)} \cos(-t\beta_i) \end{bmatrix} \begin{bmatrix} \text{Re } p_i^\dagger \\ -\text{Im } p_i^\dagger \end{bmatrix}, \quad (13)$$

$$= 2e^{-t(\alpha_i-1)} \begin{bmatrix} \text{Re } p_i & \text{Im } p_i \end{bmatrix} \begin{bmatrix} \cos(t\beta_i) & -\sin(t\beta_i) \\ \sin(t\beta_i) & \cos(t\beta_i) \end{bmatrix} \begin{bmatrix} \text{Re } p_i^\dagger \\ -\text{Im } p_i^\dagger \end{bmatrix}, \quad (14)$$

$$= e^{-t(\alpha_i-1)} \begin{bmatrix} \text{Re } p_i & \text{Im } p_i \end{bmatrix} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} e^{-t\beta_i i} & 0 \\ 0 & e^{t\beta_i i} \end{bmatrix} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} \begin{bmatrix} \text{Re } p_i^\dagger \\ -\text{Im } p_i^\dagger \end{bmatrix}, \quad (15)$$

$$= e^{-t(\alpha_i-1)} \begin{bmatrix} p_i & \bar{p}_i \end{bmatrix} \begin{bmatrix} e^{-t\beta_i i} & 0 \\ 0 & e^{t\beta_i i} \end{bmatrix} \begin{bmatrix} p_i^\dagger \\ \bar{p}_i^\dagger \end{bmatrix} = \begin{bmatrix} p_i & \bar{p}_i \end{bmatrix} e^{-t \left( \begin{bmatrix} \lambda_i & 0 \\ 0 & \bar{\lambda}_i \end{bmatrix} - I \right)} \begin{bmatrix} p_i^\dagger \\ \bar{p}_i^\dagger \end{bmatrix}, \quad (16)$$

for  $1 \leq i \leq r$ .  $\square$

## C.2 Proposition: Image-Space Trajectory Consistency Loss and its Latent-Space Counterpart

In this part, we first propose the informal version of the equivalence between image-space trajectory consistency loss and its latent-space counterpart under specific assumptions. Then we present the formal version.

**Proposition C.2** (Equivalence of Trajectory Consistency Losses under Structural Assumptions, Informal). *Assume that the decoder  $\mathcal{H} = \mathcal{F} \circ \mathcal{D}_\phi$ ,  $\forall \mathcal{F}$  is differentiable with respect to each Koopman state  $\mathbf{w}^{(l)}$ , and satisfies isotropy and level disentanglement properties with respect to the Koopman decomposition. Then, the image-space trajectory consistency loss:*

$$\mathcal{L}_{t-\text{consist}} = \mathbb{E}_{t \sim \mathcal{U}[\epsilon, T]} \left[ d \left( \mathcal{D}_\phi(\{e^{(\epsilon-t)\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(\mathbf{x}_t)\}_{l=1}^L), \mathbf{x}_\epsilon \right) \right], \quad (17)$$

is approximately equivalent to the following latent-space loss:

$$\mathcal{L}_{t-\text{consist}} \approx \mathbb{E}_{t \sim \mathcal{U}[\epsilon, T]} \sum_{l=1}^L \left\| \frac{\partial \mathcal{F} \circ \mathcal{D}}{\partial \mathbf{w}^{(l)}} \right\| \cdot \left\| e^{-t\mathbf{A}^{(l)}} \right\| \cdot \left\| \mathcal{E}_\theta^{(l)}(\mathbf{x}_t), e^{(t-T)\mathbf{A}^{(l)}} \mathbf{w}_T^{(l)} \right\|_2^2, \quad (18)$$

where  $d(\cdot, \cdot)$  denotes suitable distance metric such as squared  $L_2$ -norm or perceptual distances like LPIPS with a corresponding feature extractor  $\mathcal{F}$ , i.e.,  $d(\mathbf{x}, \mathbf{y}) \triangleq \|\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y})\|_2^2$ . The approximation holds up to small residual terms governed by the assumptions as detailed in the derivation.

*Proof.* Our proof relies on three key assumptions. The assumptions are (1) the level-based Koopman states  $\mathbf{w}_t^{(l)}$  are disentangled across levels labeled by  $l$ , (2) the mapping  $\mathcal{H}$  is isotropic w.r.t. the Koopman states, and (3) the spectrums of the Koopman observables are close to 1.

Intuitively, it requires (1) Koopman states in different levels depict different features, (2) each Koopman state has the same unit and contributes equally to the loss function, and (3) the Koopman space sufficiently models the latent evolution. A detailed version of the derivation, with the divergence from the assumptions included, is

$$\mathbb{E}_{t \sim \mathcal{U}[\epsilon, T], \mathbf{x}_t \sim p_{\text{data}}} \left\| \mathcal{F} \circ \mathcal{D} \left[ \{e^{-t\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(\mathbf{x}_t)\}_{l=1}^L \right] - \mathcal{F} \circ \mathcal{D} \left[ \{\mathbf{w}_0^{(l)}\}_{l=1}^L \right] \right\|_2^2, \quad (19)$$

$$\stackrel{(*)}{=} \sum_{l=1}^L \mathbb{E} \left\| \mathcal{H}(\mathbf{w}_l) - \mathcal{H}(\mathbf{w}_{l-1}) \right\|_2^2 + o(\rho), \quad (20)$$

$$\stackrel{(**)}{=} \sum_{l=1}^L \mathbb{E} \left\| \frac{\partial \mathcal{H}}{\partial \mathbf{w}^{(l)}} \right\| \cdot \left\| e^{-t\mathbf{A}^{(l)}} \mathcal{E}_\theta^{(l)}(\mathbf{x}_t) - \mathbf{w}_0^{(l)} \right\|_2^2 + o(\rho + \epsilon), \quad (21)$$

$$\stackrel{(***)}{=} \sum_{l=1}^L \mathbb{E} \left\| \frac{\partial \mathcal{H}}{\partial \mathbf{w}^{(l)}} \right\| \cdot \left\| \mathcal{E}_\theta^{(l)}(\mathbf{x}_t) - \mathbf{w}_t^{(l)} \right\|_2^2 + o(\rho + \epsilon + \delta), \quad (22)$$

where parameters  $\mathbf{w}_l \triangleq \{\mathbf{e}^{-t\mathbf{A}^{(l)}} \mathcal{E}_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}_t)\}_{k=1}^l \cup \{\mathbf{w}_0^{(l)}\}_{k=l+1}^L$ , and the equations  $(*)$ ,  $(**)$ , and  $(***)$  correspond to the assumptions above respectively. The error term consists of three small-value terms where  $\rho$  is the extent of entanglement among levels,  $\varepsilon$  is the amount of anisotropy of  $\mathcal{H}$ , and  $\delta$  is the range of Koopman spectrum.

□

**Proposition C.3** (Equivalence of Trajectory Consistency Losses under Structural Assumptions, Formal). *Under the assumptions that (1) the level-based Koopman states  $\mathbf{w}_t^{(l)}$  are strongly disentangled across levels with an upper bound of correlations  $\rho$ , (2) the mapping  $\mathcal{H}$  is isotropic w.r.t. the Koopman states, with a lower bound of  $\eta^{(l)}$  with non-zero weights for sufficient models, and (3) the spectrums of the Koopman observables are close to 1, which leads to the fact that the maximal spectrum  $\alpha$  of matrix  $\mathbf{A}$  has a small positive value. Following the notations  $\mathbf{w}^{(l)}$ ,  $\mathcal{H}$ ,  $\mathcal{F}$ , and  $d$  in Proposition C.2., we have*

$$\mathcal{L}_{t-\text{consist}} = \mathbb{E}_{t \sim \mathcal{U}[\varepsilon, T]} \left[ d \left( \mathcal{D}_{\phi}(\{\mathbf{e}^{(\varepsilon-t)\mathbf{A}^{(l)}} \mathcal{E}_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}_t)\}_{l=1}^L), \mathbf{x}_{\varepsilon} \right) \right], \quad (23)$$

$$\geq \left(1 - \frac{\rho^2 L}{4}\right) \cdot e^{-t\alpha} \cdot \sum_{l=1}^L \eta^{(l)} \cdot \mathbb{E} \|\mathcal{E}_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}_t) - e^{(t-T)\mathbf{A}^{(l)}} \mathbf{w}_T^{(l)}\|_2^2, \quad (24)$$

which showcases that the minimization of  $\mathcal{L}_{t-\text{consist}}$  enforces the minimization of estimation error in the Koopman space.

*Proof.* Under the given assumptions, we have

$$\mathbb{E}_{t \sim \mathcal{U}[\varepsilon, T], \mathbf{x}_t \sim p_{\text{data}}} \|\mathcal{F} \circ \mathcal{D}[\mathbf{w}_L] - \mathcal{F} \circ \mathcal{D}[\mathbf{w}_0]\|_2^2 = \mathbb{E} \left\| \sum_{l=1}^L [\mathcal{H}(\mathbf{w}_l) - \mathcal{H}(\mathbf{w}_{l-1})] \right\|_2^2, \quad (25)$$

$$= \sum_{l=1}^L \mathbb{E} \|\mathcal{H}(\mathbf{w}_l) - \mathcal{H}(\mathbf{w}_{l-1})\|_2^2 + \sum_{\substack{l,k=1 \\ l \neq k}}^L \mathbb{E} [\mathcal{H}(\mathbf{w}_l) - \mathcal{H}(\mathbf{w}_{l-1})]^\top [\mathcal{H}(\mathbf{w}_k) - \mathcal{H}(\mathbf{w}_{k-1})], \quad (26)$$

$$\stackrel{(*)}{\geq} \left(1 - \frac{\rho^2 L}{4}\right) \sum_{l=1}^L \mathbb{E} \|\mathcal{H}(\mathbf{w}_l) - \mathcal{H}(\mathbf{w}_{l-1})\|_2^2, \quad (27)$$

$$\stackrel{(**)}{\geq} \left(1 - \frac{\rho^2 L}{4}\right) \sum_{l=1}^L \eta^{(l)} \cdot \mathbb{E} \|\mathbf{e}^{-t\mathbf{A}^{(l)}} \mathcal{E}_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}_t) - \mathbf{w}_0^{(l)}\|_2^2, \quad (28)$$

$$\stackrel{(***)}{\geq} \left(1 - \frac{\rho^2 L}{4}\right) \cdot e^{-t\alpha} \cdot \sum_{l=1}^L \eta^{(l)} \cdot \mathbb{E} \|\mathcal{E}_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}_t) - \mathbf{e}^{(t-T)\mathbf{A}^{(l)}} \mathbf{w}_T^{(l)}\|_2^2, \quad (29)$$

where parameters  $\mathbf{w}_l \triangleq \{\mathbf{e}^{-t\mathbf{A}^{(l)}} \mathcal{E}_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}_t)\}_{k=1}^l \cup \{\mathbf{w}_0^{(l)}\}_{k=l+1}^L$ . In addition, the inequality  $\stackrel{(*)}{\geq}$  holds under the assumption (1) where the maximal entanglement of Koopman observables at different levels  $\rho \triangleq \max_{l,k=1, \dots, L} \left| \text{Corr}[\mathcal{H}(\mathbf{w}_l) - \mathcal{H}(\mathbf{w}_{l-1}), \mathcal{H}(\mathbf{w}_k) - \mathcal{H}(\mathbf{w}_{k-1})] \right|$  is small. On the other hand,

the inequality  $\stackrel{(**)}{\geq}$  and  $\stackrel{(***)}{\geq}$  holds when  $\eta^{(l)} \triangleq \inf_{\Delta \mathbf{w}} \|\mathcal{H}(\mathbf{w}^{(l)} + \Delta \mathbf{w}) - \mathcal{H}(\mathbf{w}^{(l)})\|_2^2 / \|\Delta \mathbf{w}\|_2^2$  is the minimal spectrum of  $\mathcal{H}$ , and  $\alpha \triangleq r(\mathbf{A}^{(l)})$  is the spectral radius of matrix  $\mathbf{A}^{(l)}$ .

Under the assumptions, there holds  $\rho \rightarrow 0^+$  and  $\alpha \rightarrow 0^+$ . Consequently, minimizing the left-hand side (as in the proposed algorithm) is equivalent to the minimization of Koopman observables to the right-hand side. □

### C.3 Formal Statement and Proof of Theorem 3.1

In this section, we provide the formal statement and proof of Theorem 3.1. Before that, we first give Definition C.1, Proposition C.3, and Lemma C.1 to introduce Theorem 3.1.

In order to illustrate the feasibility of our proposed method, we propose to measure the complexity of functions by the estimation accuracy using the simplicial complex created by Delaunay triangulation.

Def. C.1 first introduces the simplicial error, which reflects the complexity of the subset  $\Omega$  as  $\varepsilon_N(\Omega) = 0$  if and only if  $\dim \Omega = m$  and  $\Omega$  is convex.

**Definition C.1** (The Simplicial Error). *For a compact subset of an  $m$ -dimensional manifold  $\Omega \subset \mathcal{M}^m \subset \mathbb{R}^d$ , the simplicial error of it is  $\varepsilon_N(\Omega) \triangleq \inf_{\Xi \subset \Omega, |S_m(\Xi)|=N} \varepsilon_\Xi(\Omega)$  where  $\Xi$  is a set of  $N$  generic point samples in  $\Omega$ ,  $\varepsilon_\Xi(\Omega) \triangleq \sup_{x \in \Omega} \min_{s \in S_m(\Xi)} d(x, s)$  is the error of the  $m$ -dimensional Delaunay triangulation  $S_m(\Xi)$  of the point set  $\Xi$ , and  $d(x, s) \triangleq \inf_{y \in s} \|x - y\|$  is the minimal distance from  $x$  to simplex  $s$ . The norm  $\|\cdot\|$  is commonly  $L_2$ -norm.*

We then propose to measure the theoretical error bounds for network estimation by the error between its graph and a closest simplicial complex. Firstly, Prop. C.4 defines the function complexity, which is intuitively influenced by the complexity of its domain and range sets. The conclusion in Lem. C.1 further shows a bound for the estimation error of the function by a network using ReLU activations. The error is closely related to the function complexity we defined, which is coherent with the common knowledge, as the network creates a simplicial complex with each activation creating a surface between simplicials.

**Proposition C.4.** *The function simplicial error is  $\varepsilon_N(f) \triangleq \varepsilon_N(\{(x, f(x)) | x \in \Omega\})$  for  $f : \Omega \rightarrow \Gamma$ . We then have  $\max\{\varepsilon_N(\Omega), \varepsilon_N(\Gamma)\} \leq \varepsilon_N(f) \leq \inf_{\Xi \subset \Omega, |S_m(\Xi)|=N} \|\varepsilon_\Xi(\Omega), \varepsilon_\Xi(\Gamma)\|$ .*

*Proof.* We first prove the left side.

$$\begin{aligned} \varepsilon_N(f) &= \inf_{\substack{\Xi \subset \Omega \\ |S_m(\Xi)|=N}} \left[ \sup_{\theta \in \Theta} \min_{s \in S_m(\Xi)} d(\theta, s) \right] \geq \inf_{\substack{\Xi \subset \Theta \\ |S_m(\Xi)|=N}} \left[ \sup_{\theta \in \Theta} \min_{s \in S_m(\Xi)} d(P_\Omega \theta, P_\Omega s) \right], \\ &= \inf_{\substack{P_\Omega \Xi \subset P_\Omega \Theta \\ |S_m(P_\Omega \Xi)|=N}} \left[ \sup_{P_\Omega \theta \in P_\Omega \Theta} \min_{P_\Omega s \in S_m(P_\Omega \Xi)} d(P_\Omega \theta, P_\Omega s) \right] = \varepsilon_N(P_\Omega \Theta) = \varepsilon_N(\Omega), \end{aligned}$$

where  $P_\Omega$  is the projector to  $\Omega$ . Similarly,

$$\begin{aligned} \varepsilon_N(f) &= \inf_{\substack{\Xi \subset \Theta \\ |S_m(\Xi)|=N}} \left[ \sup_{\theta \in \Theta} \min_{s \in S_m(\Xi)} d(\theta, s) \right] \geq \inf_{\substack{\Xi \subset \Theta \\ |S_m(\Xi)|=N}} \left[ \sup_{\theta \in \Theta} \min_{s \in S_m(\Xi)} d(P_\Gamma \theta, P_\Gamma s) \right], \\ &= \inf_{\substack{P_\Gamma \Xi \subset \Gamma \\ |S_m(P_\Gamma \Xi)|=N}} \left[ \sup_{P_\Gamma \theta \in P_\Gamma \Theta} \min_{P_\Gamma s \in S_m(P_\Gamma \Xi)} d[P_\Gamma \theta, P_\Gamma s] \right], \\ &= \inf_{\substack{\Xi \subset \Gamma \\ |S_m(\Xi)|=N}} \varepsilon_{f(\Xi)}(\Gamma) \geq \varepsilon_N(\Gamma). \end{aligned}$$

On the other hand, the right-hand side holds due to,

$$\begin{aligned} \varepsilon_N(f) &= \inf_{\substack{\Xi \subset \Theta \\ |S_m(\Xi)|=N}} \left[ \sup_{\theta \in \Theta} \min_{s \in S_m(\Xi)} \inf_{\varphi \in s} \left\| \left[ \|P_\Omega(\theta - \varphi)\|, \|P_\Gamma(\theta - \varphi)\| \right] \right\| \right], \\ &\leq \inf_{\substack{\Xi \subset \Theta \\ |S_m(\Xi)|=N}} \left[ \sup_{\theta \in \Theta} \min_{s \in S_m(\Xi)} \inf_{\varphi \in s} \left\| [\varepsilon_\Xi(P_\Omega \Theta), \varepsilon_\Xi(P_\Gamma \Theta)] \right\| \right], \\ &\stackrel{*}{=} \inf_{\substack{\Xi \subset \Omega \\ |S_m(\Xi)|=N}} \left\| [\varepsilon_\Xi(\Omega), \varepsilon_{f(\Xi)}(\Gamma)] \right\|, \end{aligned}$$

where  $\stackrel{*}{=}$  holds as the norm inside inf is a constant w.r.t.  $\theta, \varphi$  and  $s$ .  $\square$

**Lemma C.1.** *The estimation of function  $f : \Omega^m \rightarrow \Gamma$  by a network  $\mathcal{F}$  with  $n_a$  ReLU activations satisfies  $\mathbb{P}(\|\mathcal{F}(x) - f(x)\| \leq \varepsilon_{\lceil 2n_a/(m+1) \rceil}(f)) \geq 1 - \delta$  where  $\delta$  indicates the amount undertrained.*

*Proof.* With reference to the universal approximation theorem, each  $(m-1)$ -dimensional face of the simplex matches a cuspidal point of a ReLU activation. For a network with  $n_a$  activations, the maximal number of faces is  $n_a$  which means there are at most  $\lceil \frac{2n_a}{m+1} \rceil$  simplices in the simplicial complex, as each simplex has  $m+1$  faces and all faces are clamped between (exactly) two simplices.

For activations other than ReLU, one may use intrinsic Delaunay triangulation to replace the Delaunay triangulation.  $\square$

**Theorem C.1** (Formal Statement of Theorem 3.1). *If the noise space  $\mathbf{x}_T \in \Xi \subset \mathbb{R}^{n \times n}$  and the Koopman space  $\Psi \subset \mathbb{R}^D$  are compact, the ideal estimation error of our proposed HKD model with encoder  $\mathcal{E}$  and decoder  $\mathcal{D}$  networks of  $n_a/2$  activation functions is smaller than that of an end-to-end one-step model  $\mathcal{F}$  with  $n_a$  activation functions, i.e.,  $err_{HKD} \leq err_{one-step} + O(\kappa)$  holds for any one-step diffusion where*

$$err_{HKD} = \varepsilon_{\lceil n_a/(n^2+1) \rceil}(\mathcal{E}) + \kappa + \varepsilon_{\lceil n_a/(D+1) \rceil}(\mathcal{D}) \text{ and } err_{one-step} = \varepsilon_{\lceil 2n_a/(n^2+1) \rceil}(\mathcal{F})$$

for the one-step mapping  $\mathcal{F}$ , the encoder  $\mathcal{E}$  and decoder  $\mathcal{D}$  in the proposed framework. The additional Koopman error  $\kappa$  is

$$\kappa \triangleq \max \left\{ \frac{2\sqrt{3}\sigma m^2 \rho_{\sup}}{\sqrt{N(\frac{\delta}{3} - 2m^{-\frac{r}{3}}) - \sqrt{3}\sigma m^2}}, \frac{2\sqrt{3}\sigma m^2 \rho_{\inf}}{\sqrt{N(\frac{\delta}{3} - 2m^{-\frac{r}{3}}) \cdot \rho_{\inf} - \sqrt{3}\sigma m^2}} \right\} + o(m^{-\frac{r}{3}}),$$

which is small when the Koopman dimension  $m$  is sufficient and the dataset size  $N$  is large enough. Here, parameter  $\sigma$  is the standard deviation of the observables,  $\delta$  is 0 when the network is perfectly trained,  $(\rho_{\inf}, \rho_{\sup})$  is the true spectral range of the Koopman operator and  $r \rightarrow -\infty$  when the  $m$ -dimensional Koopman process adequately model the Koopman space.

*Proof.* Under the assumption of AE-based data compression methods, we assume that the data are approximately on an  $m$ -dimensional manifold  $\mathcal{M}^m \subset \mathbb{R}^{n \times n}$  defined by a bijection  $\mathfrak{F} : \mathcal{M} \leftrightarrow \mathbb{R}^m$ , and that the projection of the dataset on the manifold is  $\Omega \subset \mathcal{M}$ . Let  $\Psi = \mathfrak{F}(\Omega)$  be a compact subset of  $\mathbb{R}^m$  that encodes the dataset.

We consider  $\mathbf{x}_t(p)$  as a function w.r.t. spatial position  $p$  and define operators  $\mathcal{F}_t : \mathbf{x}_t(\cdot) \mapsto f(\mathbf{x}_t(\cdot), t)$ ,  $\mathcal{G}_t : \mathbf{x}_t(\cdot) \mapsto g^2(t) \cdot \nabla \log p_t(\mathbf{x}_t(\cdot))$ , hence we have,

$$\delta(\mathfrak{F}\mathbf{x}_t) = [\mathfrak{F}\mathcal{F}_t\mathbf{x}_t - \mathfrak{F}\mathcal{G}_t\mathbf{x}_t] \delta t = \mathfrak{F}(\mathcal{F}_t - \mathcal{G}_t)\mathfrak{F}^{-1}(\mathfrak{F}\mathbf{x}_t)\delta t,$$

which satisfies the definition of a Koopman process where the operator is  $\mathcal{K}_t = \mathcal{I} + \mathfrak{F}(\mathcal{F}_t - \mathcal{G}_t)\mathfrak{F}^{-1}$ .

Letting  $\boldsymbol{\xi} = (\mathcal{K}_t - \hat{\mathcal{K}}_t)\mathbf{y}_t$  for the observables. We have the bound that

$$\mathbb{P} \left( \|(\mathcal{K}_t - \hat{\mathcal{K}}_t)\mathbf{y}_t\| \leq \kappa_t \|\mathbf{y}_t\| \right) \geq 1 - \delta,$$

where  $\kappa_t = 2\sqrt{3}\sigma m^2 r(\mathcal{K}_t) / \left[ \sqrt{N(\delta - 2m^{-\frac{r}{3}}) \cdot \min\{1, r(\mathcal{K}_t)\} - \sqrt{3}\sigma m^2} \right] + o(m^{-\frac{r}{3}})$ .

Here, the bound factor  $\kappa_t$  is related to time  $t$  while it has a uniform bound factor,

$$\kappa^{(\delta)} \triangleq \sup_{t \in [0, T]} \kappa_t = \max \left\{ \frac{2\sqrt{3}\sigma m^2 \rho_{\sup}}{\sqrt{N(\delta - 2m^{-\frac{r}{3}}) - \sqrt{3}\sigma m^2}}, \frac{2\sqrt{3}\sigma m^2 \rho_{\inf}}{\sqrt{N(\delta - 2m^{-\frac{r}{3}}) \cdot \rho_{\inf} - \sqrt{3}\sigma m^2}} \right\} + o(m^{-\frac{r}{3}}),$$

where  $\rho_{\inf} \triangleq \inf_t r(\mathcal{K}_t)$  and  $\rho_{\sup} \triangleq \sup_t r(\mathcal{K}_t)$ .

The estimated Koopman model with an encoder  $\mathcal{E}$  estimating mapping  $\mathfrak{F}$  and a decoder  $\mathcal{D}$  estimating  $\mathfrak{F}^{-1}$  (with each of them having  $n_a/2$  activations) has a probabilistic error bound,

$$\begin{aligned} & \mathbb{P}(\|\mathcal{D}\mathbf{z}_0 - \mathbf{x}_0\| \leq \varepsilon), \text{ where } \mathbf{z}_t = \mathbf{z}_T + \int_T^t (\hat{\mathcal{K}}_s - \mathcal{I})\mathbf{z}_s \, ds, \mathbf{z}_T = \mathcal{E}(\mathbf{x}_T), \\ & \text{and } \mathbf{x}_0 = \mathfrak{F}^{-1}\mathbf{y}_0, \mathbf{y}_t = \mathbf{y}_T + \int_T^t (\mathcal{K}_s - \mathcal{I})\mathbf{y}_s \, ds, \mathbf{y}_T = \mathfrak{F}\mathbf{x}_T, \\ & \geq \mathbb{P}(\|\mathcal{D}\mathbf{z}_0 - \mathfrak{F}^{-1}\mathbf{z}_0\| + \|\mathfrak{F}^{-1}\mathbf{z}_0 - \mathfrak{F}^{-1}\mathbf{y}_0\| \leq \varepsilon) \\ & \stackrel{*}{\geq} \mathbb{P}(\|\mathcal{D}\mathbf{z}_0 - \mathfrak{F}^{-1}\mathbf{z}_0\| \leq \varepsilon_{\lceil \frac{n_a}{D+1} \rceil}(\mathfrak{F}^{-1})) \cdot \mathbb{P}(\|\mathbf{z}_0 - \mathbf{y}_0\| \leq \kappa^{(\delta/3)} \zeta(e^{\rho T} - 1)/\rho) \\ & \quad \cdot \mathbb{P}(\|\mathbf{z}_T - \mathbf{y}_T\| \leq \varepsilon_{\lceil \frac{n_a}{n^2+1} \rceil}(\mathfrak{F})), \\ & \geq (1 - \delta/3)^3 \geq 1 - \delta. \end{aligned}$$

Here,  $\stackrel{*}{\geq}$  holds due to Lemma 1 and  $\stackrel{*}{\geq}$  holds when

$$\varepsilon = \varepsilon_{\lceil \frac{n_a}{D+1} \rceil}(\mathfrak{F}^{-1}) + r(\mathfrak{F}^{-1}) \cdot \kappa^{(\delta/3)} \zeta(e^{\rho T} - 1)/\rho + r(\mathfrak{F}^{-1}) \cdot e^{\rho T} \cdot \varepsilon_{\lceil \frac{n_a}{n^2+1} \rceil}(\mathfrak{F}),$$

where  $r(\mathfrak{F}^{-1})$  is the spectrum radius of  $\mathfrak{F}^{-1}$  and the bound for  $\|z_0 - y_0\|$  is given by

$$\begin{aligned}\|z_t - y_t\| &\leq \|z_T - y_T\| + \left\| \int_T^t (\mathcal{K}_s - \mathcal{I})(z_s - y_s) ds \right\| + \int_T^t \|(\hat{\mathcal{K}}_s - \mathcal{K}_s)z_s\| ds, \\ &\leq \|z_T - y_T\| + \rho \int_t^T \|z_s - y_s\| ds + \kappa \zeta (T - t),\end{aligned}$$

where  $\rho \triangleq \max\{1 - \rho_{\inf}, \rho_{\sup} - 1\}$  and  $\zeta$  is a bound of  $\|z_t\|$ . Let  $D_t$  satisfies  $D_T = \|z_T - y_T\|$  and  $\frac{dD_t}{dt} = -\rho D_t - \kappa \zeta$  which leads to  $d_t \triangleq \|z_t - \tilde{z}_t\| \leq D_t$  as they are positive trajectories. The solution to  $D_t$  is  $D_t = \frac{\kappa \zeta}{\rho} [e^{\rho(T-t)} - 1] + D_T e^{\rho(T-t)}$ .

For ideal  $\mathcal{K}_t$  and  $\mathfrak{F}$ ,  $\varepsilon$  is dominated by  $\varepsilon_{\lceil \frac{n_a}{D+1} \rceil}(\mathfrak{F}^{-1}) + \varepsilon_{\lceil \frac{n_a}{n^2+1} \rceil}(\mathfrak{F})$  as  $\rho$  is small and  $r(\mathfrak{F}^{-1}) \approx 1$ .

On the other hand, the one-step methods aim at estimating  $\hat{x}_0 = \mathcal{F}(x_T) \triangleq x_T + \int_T^0 (\mathcal{F}_t - \mathcal{G}_t)x_t dt$  by a network  $\mathcal{F}$  with  $n_a$  activation functions. The error satisfies, according to Lemma 1,

$$\mathbb{P}(\|\mathcal{F}(x_T) - x_0\| \leq \varepsilon_{\lceil 2n_a/(n^2+1) \rceil}(\mathcal{F})) \geq 1 - \delta.$$

Note that ideally,  $x_T \in \Xi \subset \mathbb{R}^{n \times n}$ ,  $x_0 \in \Omega \subset \mathcal{M}^m$  and  $\mathfrak{F}x_0 \in \Psi \subset \mathbb{R}^m$ , where  $\Xi$  and  $\Lambda$  are compact, hence

$$\varepsilon_{\lceil \frac{2n_a}{n^2+1} \rceil}(\mathcal{F}), \tag{30}$$

$$\geq \max \left\{ \varepsilon_{\lceil \frac{2n_a}{n^2+1} \rceil}(\Xi), \varepsilon_{\lceil \frac{2n_a}{n^2+1} \rceil}(\Omega) \right\}, \quad \Leftarrow \text{Proposition 1} \tag{31}$$

$$= \varepsilon_{\lceil \frac{2n_a}{n^2+1} \rceil}(\Omega), \quad \Leftarrow \Omega \text{ is compact} \tag{32}$$

$$= \|\varepsilon_{\mathfrak{F}(\Delta)}(\Psi), \varepsilon_{\Delta}(\Omega)\| + \|\varepsilon_{\cdot}(\Xi), \varepsilon_{\mathfrak{F}(\cdot)}(\Psi)\|, \quad \Leftarrow \Xi \text{ and } \Psi \text{ are compact} \tag{33}$$

$$\text{where } \Delta \text{ satisfies } \varepsilon_{\Delta}(\Omega) = \varepsilon_{\lceil \frac{2n_a}{n^2+1} \rceil}(\Omega), \quad \varepsilon_{\cdot}(\Xi) = \varepsilon_{\cdot}(\Psi) = 0 \tag{34}$$

$$\geq \varepsilon_{\lceil \frac{n_a}{D+1} \rceil}(\mathfrak{F}^{-1}) + \varepsilon_{\lceil \frac{n_a}{n^2+1} \rceil}(\mathfrak{F}). \quad \Leftarrow \text{Proposition 1} \tag{35}$$

$$\tag{36}$$

Using Lem. C.1, we may reach the conclusion in Thm. C.1, which shows that if the neural network models are trained properly, the theoretical error bound of the proposed method is smaller than that of any other end-to-end one-step method.  $\square$

## D Experimental Supplement

### D.1 Additional Experimental Results for Sec. 4.1

**Conditional Generation on CIFAR-10.** In this part, we provide the conditional generation results on CIFAR-10 to show the generalization ability of our framework on class-conditional tasks. The results are provided in Tab. 4.

**Visual Results.** In this section, we provide additional qualitative results on CIFAR-10 (unconditional), FFHQ (unconditional), and CIFAR-10 (conditional) to further demonstrate the visual quality and effectiveness of our proposed method. The results are presented in Fig. 7, Fig. 8, Fig. 9, respectively.

Table 4: Class-conditional sample quality on CIFAR-10 dataset.

| Methods       | NFE( $\downarrow$ ) | FID( $\downarrow$ ) |
|---------------|---------------------|---------------------|
| Score SDE     | 2000                | 2.20                |
| EDM           | 35                  | 1.79                |
| DMD (w/o REG) | 1                   | 5.58                |
| DMD (w/o KL)  | 1                   | 3.82                |
| DMD           | 1                   | 2.66                |
| HKD           | 1                   | 2.77                |

**Training Stability Results.** We include the results of training HKD five times independently on the CIFAR-10 dataset in Tab. 5. The FID scores and their standard deviation are reported in the table below. We summarize the FID results over every 10 epochs during training across the five runs, demonstrating stable convergence behavior.

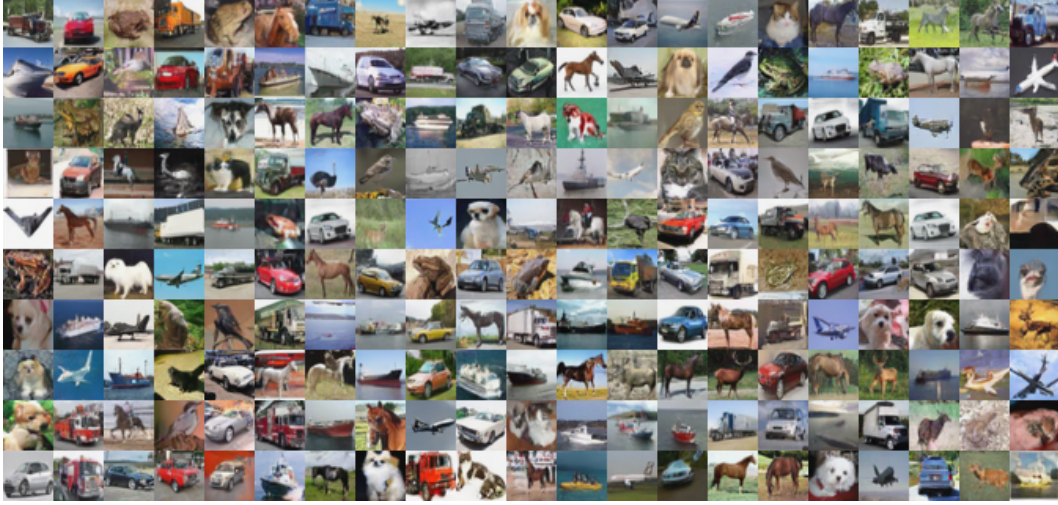


Figure 7: The additional visualization of image generations from HKD trained by the CIFAR10 dataset.



Figure 8: The additional visualization of image generations from HKD trained by the FFHQ dataset.

**The Wall-clock Time Per Image for Generation Methods.** We provide the wall-clock time per image for different generation methods in Tab. 6, measured on an NVIDIA V100 GPU. These methods were trained on the CIFAR-10 dataset.

## D.2 Additional Experimental Results for Sec. 4.2

We provide additional visualization results of image models on the CIFAR-10 dataset in Fig. 10, and on the FFHQ dataset in Fig. 11. As shown in Fig. 11, the reconstructed images using low-frequency components primarily capture the semantic structure of the face. The mid-frequency components contribute to the overall contour and shape details of the face, while the high-frequency components are responsible for fine-grained details such as facial hair.



Figure 9: The additional visualization of conditional image generations from HKD trained by the CIFAR10 dataset.

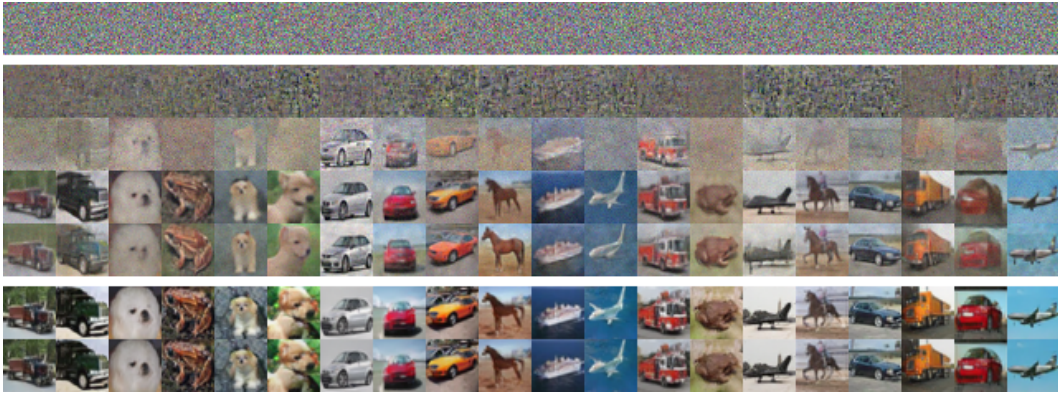


Figure 10: The additional visualization of image modes for the CIFAR10 dataset. The rows correspond to those in the manuscript.

### D.3 Additional Experimental Results for Sec. 4.3

**Additional Results on One-step Image Editing via Frequency-aware Interventions.** We provide the additional results on the one-step image editing experiment via frequency-aware interventions along the diffusion trajectory. The results are provided in Fig. 12, 13.

### D.4 More Implementation Details about Experiments for Sec. 4.1

**Network Architecture.** In our experiment, both the encoder and decoder are designed to follow the architecture of the EDM [8] encoder and decoder, leveraging the proven effectiveness of existing diffusion model designs, similar to consistency models [32]. Inspired by ECM [4], we utilize EDM’s pre-trained weights for our encoder and decoder modules to provide structured latent-to-output mappings and prevent mode collapse, retaining rich hierarchical features acquired during large-scale training. For each linear operator  $A^{(l)}(i, j)$ , which adopts a block-diagonal structure, we perform vectorization by concatenating the real and imaginary parts—placing the real components in the first half of the vector and the imaginary components in the second half. The optimization is then conducted over the elements of this vectorized form. We train our model end-to-end.

Table 5: FID scores (mean  $\pm$  standard deviation) during training across epochs.

|                                        | Epoch 0          | Epoch 10        | Epoch 20        |
|----------------------------------------|------------------|-----------------|-----------------|
| <b>FID (Mean <math>\pm</math> Std)</b> | 11.82 $\pm$ 0.29 | 5.70 $\pm$ 0.21 | 4.71 $\pm$ 0.17 |
|                                        | Epoch 30         | Epoch 40        | Epoch 50        |
| <b>FID (Mean <math>\pm</math> Std)</b> | 4.09 $\pm$ 0.18  | 3.52 $\pm$ 0.11 | 3.31 $\pm$ 0.03 |
|                                        | Epoch 52         | Epoch 53        | Before Training |
| <b>FID (Mean <math>\pm</math> Std)</b> | 3.31 $\pm$ 0.02  | 3.30 $\pm$ 0.02 | 447.13          |

Table 6: NFE and per-image latency across different generation methods.

|                | DDPM       | Score SDE | DDIM       | DDIM  |
|----------------|------------|-----------|------------|-------|
| <b>NFE</b>     | 1000       | 2000      | 10         | 100   |
| <b>Latency</b> | 15.32s     | 45.89s    | 0.51s      | 2.35s |
|                | EDM        | EDM       | KD         | PD    |
| <b>NFE</b>     | 35         | 15        | 1          | 1     |
| <b>Latency</b> | 0.69s      | 0.30s     | 0.26s      | 0.28s |
|                | CD (LPIPS) | DMD       | RF-1       | RF-2  |
| <b>NFE</b>     | 1          | 1         | 1          | 1     |
| <b>Latency</b> | 0.02s      | 0.03s     | 0.03s      | 0.03s |
|                | RF-3       | RF-2++    | CT (LPIPS) | iCT   |
| <b>NFE</b>     | 1          | 1         | 1          | 1     |
| <b>Latency</b> | 0.03s      | 0.03s     | 0.02s      | 0.02s |
|                | iCT-deep   | ECM       | HKD        | —     |
| <b>NFE</b>     | 1          | 1         | 1          | —     |
| <b>Latency</b> | 0.02s      | 0.02s     | 0.03s      | —     |

Compared to conventional diffusion models, our framework introduces only one additional parameter component, the linear operator  $\mathbf{A}$ . However, due to the structured block-diagonal assumption imposed on  $\mathbf{A}$ , the overall increase in parameter count is minimal.

In addition, the EDM U-Net architecture supports `augment_label` as an input, which we set to `None`. Additionally, as in EDM, the U-Net receives the noise scale  $\sigma$  as an input. Although the observable functions in our model are inherently time-independent, we follow the EDM convention and feed the time-dependent  $\sigma(t)$  into the encoder to maintain architectural compatibility and enhance representational capacity. In contrast, our decoder only operates on the final timestep; therefore, it consistently receives  $\sigma(t=1)$  as input. The EDM architecture also allows for `class_labels` as input. For unconditional generation tasks on CIFAR-10 and FFHQ, we set `class_labels = None`. For conditional generation on CIFAR-10, we adopt the same label configuration as in EDM.

**Hyperparameter Setting.** We conduct experiments using 8 NVIDIA V100 GPUs, with a batch size of 256 for the CIFAR-10 dataset and 64 for the FFHQ dataset. The loss function weights are set as  $\lambda_2 = 1$  and  $\lambda_1 = 10^{-3(\text{current\_epoch}/\text{overall\_epoch})}$ , where  $\lambda_1$  decays exponentially over the course of training. For the implementation of the trajectory consistency loss, we employ a Monte Carlo sampling strategy: in each training iteration, four intermediate timesteps are randomly sampled uniformly, and the loss is computed as the average over these sampled time points.

## D.5 More Details about Comparison Methods

To provide a comprehensive comparison, we categorize the baseline methods into three groups: multi-step diffusion models, distillation-based one-step models, and consistency-based models. Below, we summarize the key characteristics and mechanisms of each method.

**Multi-step Diffusion Baselines.** 1. DDPM (Denoising Diffusion Probabilistic Models) [7] A foundational diffusion model that learns to reverse a fixed Markovian noising process via iterative

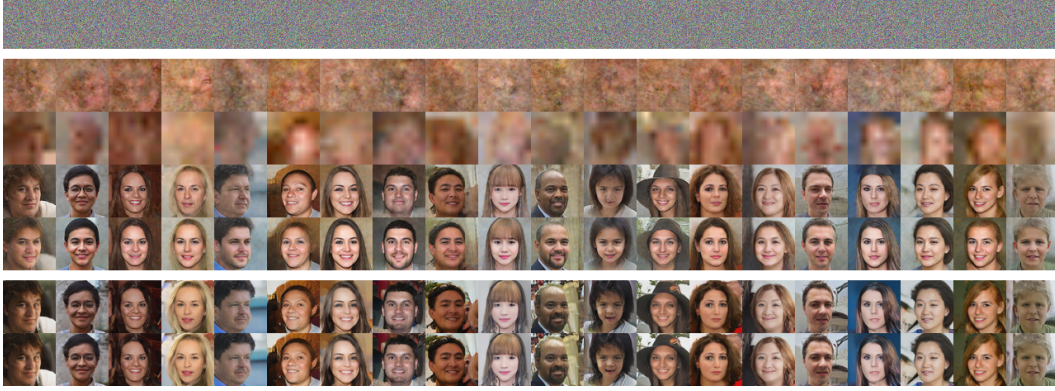


Figure 11: The visualization of image modes for the FFHQ dataset. The rows correspond to those in the manuscript. The third row, indicating the low-frequency semantics, underwent a downsampling to avoid the uncanny valley effect.



Figure 12: The visualization of frequency-aware interventions of images from CIFAR10. The rows correspond to the columns in the manuscript, meaning the original image for row 1, frequency-aware editing at factors of 10%, 20%, 50%, 80%, and 90%, frequency-agnostic editing at a factor of 90%, and the reference image, respectively.

denoising. The model is trained using a mean squared error (MSE) loss between predicted and actual noise. Sampling typically requires hundreds to thousands of steps.

2. Score SDE (Score-based Generative Modeling via Stochastic Differential Equations) [33] A Generalization towards continuous time via stochastic differential equations (SDEs). The model is trained via score matching and enables sampling using reverse-time SDE or ODE solvers, improving sample quality and flexibility.

3. DDIM (Denoising Diffusion Implicit Models) [30] A non-Markovian and deterministic variant of DDPM that introduces a sampling ODE, enabling fast sampling with fewer steps while preserving high visual fidelity. It is compatible with pre-trained DDPM models without requiring retraining.



Figure 13: The visualization of frequency-aware interventions of images from FFHQ. The rows correspond to the columns in the manuscript.

4. EDM (Elucidating the Design Space of Diffusion Models) [8] A state-of-the-art diffusion framework with improved noise scheduling (log-normal distribution) and score network preconditioning. It achieves superior performance across datasets and serves as the backbone in our proposed framework.

**Distillation-based One-step Models.** 1. KD (Knowledge Distillation) [17] A straightforward distillation approach where a student model learns to replicate the output of a pre-trained diffusion teacher via MSE loss. While efficient, performance can degrade with large step-size reductions.

2. PD (Progressive Distillation) [27] Gradually distills a high-step teacher into a low-step student via intermediate models, reducing the number of steps incrementally. This approach improves stability and fidelity over direct distillation.

3. CD (LPIPS) (Consistency Distillation with Perceptual Loss) [32] Extends distillation by employing perceptual similarity metrics (LPIPS) as the training objective. It improves visual quality by preserving perceptual features across time steps.

4. DMD (Distribution Matching Distillation) [39] Aims to directly match the student’s output distribution to that of the teacher using divergence-based objectives (e.g., KL divergence). This results in more accurate distribution alignment and improved sample diversity.

5. ReFlow Series (Rectified Flow and Distilled Variants) [14, 13] Utilizes a rectified flow framework to optimize ODE trajectories for efficient sampling. Variants such as 1-, 2-, and 3-Rectified Flow, and 2-Rectified Flow++ offer different trade-offs between speed and quality. Distilled versions further reduce sampling cost to 1 step with minimal performance loss.

**Consistency-based Models.** 1. CT / iCT / iCT-deep (Consistency Training) [32, 31] CT models are trained to produce consistent outputs when given different noise levels. Improved variants (iCT and iCT-deep) enforce deeper consistency across time steps, using perceptual and cosine similarity losses to enhance training stability and sample quality.

2. ECM (Easy Consistency Models) [4] Simplifies consistency training by initializing model parameters with those from a pretrained score-based model. This reduces training complexity and improves convergence without requiring complex schedule tuning.

## **E Broader Impact**

The proposed Hierarchical Koopman Diffusion (HKD) framework introduces a novel approach to balancing sampling efficiency and interpretability in generative modeling, which may have a significant and multifaceted societal impact. By enabling one-step image synthesis with transparent generative dynamics, HKD has the potential to make image generation systems more accessible, controllable, and explainable — attributes that are essential for responsible deployment in sensitive domains such as design, and scientific simulation.

The explicit modeling of generative trajectories and access to intermediate states could enhance human-AI collaboration, enabling users to guide or edit synthetic content with semantic intent, thus reducing risks of undesired generation and improving user trust. Furthermore, the spectral interpretability introduced by our framework can offer insights into the internal mechanics of generative models, fostering research in model debugging, safety validation, and fairness auditing.

In the broader landscape, our work contributes to the growing field of explainable generative AI, reinforcing the possibility of building models that are not only performant but also transparent and controllable. We hope this inspires further research toward interpretable, reliable, and socially responsible generative systems.