Structured State-Space Regularization for Generation-Friendly Image Tokenization

Jinsung Lee[1]
jinsunglee@postech.ac.kr

Jaemin Oh[2]
jaemin_oh@brown.edu

Namhun Kim[1]
namhunkim@postech.ac.kr

Dongwon Kim[3]
kdwon@kaist.ac.kr

Byung-Jun Yoon[4,5]
bjyoon@tamu.edu

Suha Kwak[1]
suha.kwak@postech.ac.kr

https://jinsingsangsung.github.io/WHippo

Abstract

Image tokenizers play a central role in modern generative models, where the structure of the latent space critically determines the downstream generation performance. A key but underexplored property of effective latent representations is *spectral organization*, the ability to encode information across frequency components. In this work, we introduce *structured state-space regularization*, a principled approach to inducing spectral structure in latent spaces. We derive a regularization objective by revisiting state-space models (SSMs) as systems mimicking a basis function's behavior. This perspective reveals that hidden states of SSMs are induced to capture the frequency components, resulting in a novel regularizer that enforces the latent space to capture spectral structure of images. Experiments demonstrate that our regularizer improves the generative performance of image tokenizers while incurring only minimal loss in their reconstruction fidelity.

1 Introduction

Recent advances in computer vision increasingly rely on image tokenizers. Generative models [50, 67, 75, 76], world models [2, 6, 31], and representation models [4, 7] are all moving toward paradigms where modeling is performed in a learned latent space rather than in pixel space. This shift is driven by several practical advantages: latent representations are lower-dimensional, enabling substantially cheaper computation than high-dimensional pixel representation, and are able to capture semantic layout that raw pixel intensities do not explicitly encode. As a result, image tokenizers, which shape the geometry and content of the latent space, have become a critical component in vision models, and improving them is of significant value as it can directly boost the efficiency and quality of modern generative AI systems.

A growing body of recent work focus on constructing latent spaces that are *generation-friendly*; meaning that they can be modeled accurately and efficiently by downstream generative models. Such a property is crucial for applications that rely on probabilistic sampling, including planning [6, 41], conditional generation [35, 57], editing [40, 42], and manipulation [9, 83]. Despite its importance, the precise characteristics that make a latent space generation-friendly remain unclear and are actively being investigated. Prior studies suggest several contributing factors: Skorokhodov *et al*. [70] and Kouzelis *et al*. [43] highlight the role of scale and rotation equivariance between the pixel and latent spaces, while Chen *et al*. [15] insist the number of Gaussian mixture modes required to approximate

[1] POSTECH [2] Brown University [3] KAIST [4] Texas A&M University [5] Brookhaven National Laboratory

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

latent distribution as a key factor. More recently, incorporating features from large vision foundation models [46, 82, 89] has emerged as a highly effective approach for improving generation-friendliness.

In this work, we argue that an important yet largely overlooked property underlying generation-friendly latent spaces is *spectral organization*. Viewing image generation as an ill-posed inverse problem [16, 17], exploiting the frequency structure of natural images has long been recognized as a powerful inductive bias [59, 62, 77]. Natural images exhibit characteristic spectral statistics, such as power-law distributions over frequencies [12], which have been successfully utilized in classical and modern approaches to image restoration [23, 38, 39] and compression [3, 79]. Surprisingly, however, the role of such spectral structure has received limited attention in the design of latent spaces for generative modeling. A few prior attempts [2, 22, 56] incorporate frequency-domain representations directly into the tokenizer, but these approaches often suffer from significantly different training dynamics and achieve suboptimal performance due to the discrepancy between frequency and spatial domains.

To address this limitation, we introduce spectral structure into latent spaces in a principled yet non-disruptive manner by drawing inspiration from state-space models (SSMs). SSMs implicitly encode an input signal using a set of basis functions whose coefficients correspond to the spectral components of the signal [29, 45, 84]. In particular, the hidden state of SSMs is trained to react to a signal transformation in a way that the basis function's coefficients would behave, thereby implicitly equipping the capability to capture the spectral structure of the input signal. Building on this insight, we introduce **structured state-space regularization**, a lightweight objective that encourages latent features to achieve spectral organization without altering the underlying tokenizer architecture. Our framework not only enhances the image tokenizer's generation-friendliness with minimal loss in reconstruction fidelity, but also provides intriguing insights involving SSMs as a promising, yet largely unexplored, modeling tool for image tokenization.

In summary, our contribution is three-fold:

- We propose a novel, theoretically motivated regularizer to enhance image tokenizer's downstream generation performance.
- We present the first theoretical framework for incorporating principles of SSMs into image tokenization.
- Our method proves effective on widely used image tokenizers, improving generation performance with minimal sacrifice in reconstruction quality.

2 Preliminaries

As our method is formulated within the state-space model (SSM) framework, we first introduce the notation of SSMs, describe their operational principles, and explain their practical implementation.

2.1 SSMs

In the context of machine learning, SSMs are sequence-to-sequence models that map the input function $u(t)$ to the output $y(t)$, employing the state-space representation of a linear system defined as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \qquad (1) \qquad\qquad y(t) = \mathbf{C}\mathbf{x}(t), \qquad (2)$$

where $\mathbf{x}(t) \in \mathbb{R}^{N\times 1}$, $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$, and $\mathbf{C} \in \mathbb{R}^{1\times N}$. Eq. (1) specifies the direction in which the hidden state $\mathbf{x}(t)$ is updated to incorporate the input $u(t)$ at time t, and Eq. (2) produces the output $y(t)$ from the hidden state $\mathbf{x}(t)$ using linear projection, similar to the linear state-space system from control theory [86] or language models [64]. Here, the state transition matrix $\mathbf{A}$ is particularly important, as it determines the way the hidden state evolves to integrate new inputs over time. HiPPO matrix [25] is a family of the state transition matrix $\mathbf{A}$ that enhances the long-term dependency of the memory $\mathbf{x}$, which is derived from projecting the input $u(t)$ to predefined orthogonal polynomial bases $\{\phi_n(t)\}_{n=1}^N$ to obtain their coefficients $\{c_n(t)\}_{n=1}^N$ as follows:

$$c_n(t) = \langle u_{\le t}, \phi_n \rangle_\nu = \int_0^t u(x)\phi_n(x)\nu(x)\,\mathrm{d}x, \qquad (3)$$

where ν is a measure depending on the choice of bases. Here, the HiPPO matrix $\mathbf{A}$ is derived to describe the evolution of $c_n(t)$: as the streaming input $u_{\leq t}$ accumulates over time, its basis projection $c_n(t)$ changes continuously, and $\mathbf{A}$ can be deterministically obtained by differentiating $c_n(t)$ with respect to t, yielding a differential equation that governs its dynamics. Numerous studies have shown that the structure and properties of HiPPO matrices enhance the performance of SSMs: *e.g.*, the sparse structure of $\mathbf{A}$ enables efficient matrix-vector multiplication in Eq. (1) and initializing $\mathbf{A}$ with a HiPPO matrix yields superior performance compared to random initialization [26–29, 32].

2.2 Discretization

Since observations are inherently discrete samples (*e.g.*, images can be regarded as pixels sampled from an underlying continuous signal), the continuous-time dynamics in Eq. (1) must be approximated with a step size parameter $\Delta \in \mathbb{R}_+$, yielding the recursive form:

$$\mathbf{x}_t = \overline{\mathbf{A}}\mathbf{x}_{t-1} + \overline{\mathbf{B}}u_t, \tag{4}$$

$$y_t = \mathbf{C}\mathbf{x}_t, \tag{5}$$

where $\mathbf{x}_t := \mathbf{x}(t)$, $u_t := u(t)$, and $y_t := y(t)$ for $t \in \mathbb{N}$. The discretized form admits an interpretation in which $\mathbf{x}_t$ represents the compressed history accumulated up to the t-th timestep, and the new information u_t is integrated into the compressed state to update $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$. Common discretization methods include Euler, bilinear [25, 27, 28], and zero-order hold (ZOH) [30, 78]. For instance, applying the forward Euler discretization to Eq. (1) results in:

$$\begin{aligned}
\mathbf{x}(t_i) &\approx \mathbf{x}(t_{i-1}) + \Delta \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t)\big|_{t=t_{i-1}} \quad (\because \text{Forward Euler}) \\
&= \mathbf{x}(t_{i-1}) + \Delta\big(\mathbf{A}\mathbf{x}(t_{i-1}) + \mathbf{B}u(t_i)\big) \\
&:= \overline{\mathbf{A}}\mathbf{x}(t_{i-1}) + \overline{\mathbf{B}}u(t_i).
\end{aligned} \tag{6}$$

Through these methods, $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are expressed with respect to $\mathbf{A}$, $\mathbf{B}$, and Δ.

3 Proposed method

In this section, we examine the mechanisms that enable SSMs to achieve spectral organization, and introduce a novel regularization method that leverages these mechanisms to regularize tokenizers towards a generation-friendly direction.

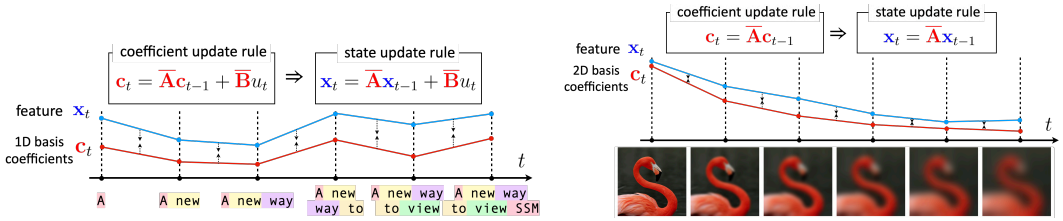
3.1 A generalized view of SSMs: basis projection and input transformation

What truly constructs SSMs? At their core, SSMs rely on a form of *basis projection* that summarizes the input, while allowing the update of the compressed representation to reflect the *input transformation*. We argue that such an update represents the core principle of SSMs that leads to a notion of *generalized view of state-space models*, which can be characterized by two key components:

1. a *basis projection* $\mathbf{c} : \bigcup_{L \in \mathbb{N}_{\geq N}} \mathbb{R}^L \to \mathbb{R}^N$,
2. an *input transformation* $\theta : \mathbf{I}_{t-1} \mapsto \mathbf{I}_t$, where $\mathbf{I}_t$ denotes the input at time t.

The choice of these two components directly determines the hidden state's update rule $\mathbf{x}_{t-1} \mapsto \mathbf{x}_t$, since the SSM formulation (Eq. (1)) adopts its update rule from the basis coefficient's dynamics computed on the predefined input transformation $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$. For instance, the update rule $\mathbf{x}_{t-1} \mapsto \mathbf{x}_t$ in Eq. (4) is adopted from the basis coefficients' update rule $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$, where $\mathbf{c}(\cdot)$ is defined as a projection onto orthogonal polynomial bases while $\theta : \mathbf{I}_{t-1} \mapsto \mathbf{I}_t$ represents the concatenation of the t-th token next to the 1D input at time $t-1$ [25]. If we let $\mathbf{c}(\cdot)$ be a projection onto 2D orthogonal polynomial bases and $\{\mathbf{I}_t\}$ be a corner-based sweeping construction of a 2D image, we obtain 2D SSMs proposed by Baron *et al*. [8] and Nguyen *et al*. [55].

This perspective on SSMs provides two key insights. First, it provides an intuitive explanation for SSM's frequency-capturing property [29, 45, 84]. Note that basis coefficients correspond to the magnitudes of spectral components in the input signal. Since the hidden state $\mathbf{x}_t$ is trained to resemble the behavior of the basis coefficients $\mathbf{c}(\cdot)$, its ability to capture frequency information can



(a) $\mathbf{c}(\cdot)$: Projection onto 1D orthogonal polynomial basis, $\theta(\cdot)$: Token concatenation

(b) $\mathbf{c}(\cdot)$: Projection onto 2D Fourier basis, $\theta(\cdot)$: Gaussian blurring

Figure 1: **Different choice of $\big(\mathbf{c}(\cdot), \theta(\cdot)\big)$ and the resulting update rules.** Existing SSMs update their hidden state based on Eq. (4), which is derived from the coefficient dynamics based on the choice **(a)**. One can choose a different combination of $\mathbf{c}(\cdot)$ and $\theta(\cdot)$ such as **(b)**, to derive new dynamics and formulate a new SSM framework.

be understood as arising from its resemblance to $\mathbf{c}(\cdot)$. This is also connected to the well-known ability of SSMs to produce compact representations, as basis representations are widely regarded as a compact way to encode signals [3, 79]. Second, this view naturally extends SSMs beyond the standard formulation in Eq. (4). By selecting different pairs of basis projections and input transformations $\big(\mathbf{c}(\cdot), \theta(\cdot)\big)$ that yield tractable update rule $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$, one can generalize SSMs to a broader range of modalities. We illustrate this generalization in Fig. 1.

3.2 Structured state-space regularization

Building on this philosophy, we propose **structured state-space regularization**, a novel regularizing formulation that emerges from the generalized view of the SSMs by extending the choice of the basis projection $\mathbf{c}(\cdot)$ and the input transformation $\theta(\cdot)$ that operates on 2D data (Fig. 1(b)).

Choice of the basis projection $\mathbf{c}(\cdot)$ A natural choice of $\mathbf{c}(\cdot)$ that allows update rules with favorable mathematical properties[1] is a projection using 2D basis functions like Fourier bases or 2D orthogonal polynomials [25, 29]. Formally, let the 2D orthogonal basis functions be denoted as $\{\phi_n\}_{n\in\mathbb{N}}$. We define the input image as a continuous surface function $\mathbf{I}(x, y) : [0, W] \times [0, H] \to \mathbb{R}$, where $\mathbf{I}(w, h)$ indicates the pixel intensity of the cell located at the (w, h)-th position. We assume a single-channel image for simplicity, as the extension to multi-channel can be achieved by independently applying the same logic to each channel [28]. Let c_n be the coefficient obtained by projecting the image $\mathbf{I}(x, y)$ onto the n-th orthonormal basis function ϕ_n:

$$c_n = \langle \mathbf{I}, \phi_n \rangle = \iint_{[0,W]\times[0,H]} \mathbf{I}(x, y)\phi_n(x, y)\, \mathrm{d}y\mathrm{d}x, \tag{7}$$

then, by the Parseval's identity, the surface function $\mathbf{I}(x, y)$ is expressed as a linear combination of the basis functions $\{\phi_n\}_{n\in\mathbb{N}}$ with the coefficients $\{c_n\}_{n\in\mathbb{N}}$:

$$\mathbf{I}(x, y) = \sum_{n=1}^{\infty} c_n(\mathbf{I})\phi_n(x, y). \tag{8}$$

If some terms in the expansion are omitted, the reconstruction becomes an approximation. Thus, we truncate the terms in Eq. (8) to define the basis projection as $\mathbf{c}(\mathbf{I}) := [c_1 \quad c_2 \quad \cdots \quad c_N]^\top$, where $N \leq HW$.

Choice of the input transformation $\theta(\cdot)$ Among various explorations for transforming an image such as autoregressive modeling [44, 65, 74], random masking/unmasking [13, 33, 49, 50], pooling [75], or denoising [34, 72], we adopt a *progressive blurring process based on Gaussian blur* [5] for two main reasons. First, Gaussian blur offers favorable mathematical properties, including linearity and differentiability, while its derivative being conveniently derived from the fundamental solution of the heat equation [61, 81]. Second, it aligns well with the inductive bias of images: Gaussian blur

[1] See Appendix A.2 for details.

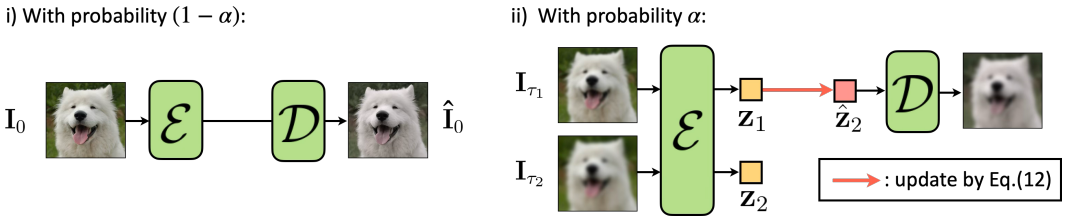


Figure 2: **State-space regularization applied to an image tokenizer.** With probability α, the update Eq. (12) is applied to the latent representation of the input images to produce $\hat{\mathbf{z}}_2$ and $\hat{\mathbf{I}}_{\tau_2}$. The network is trained to match $(\hat{\mathbf{z}}_2, \hat{\mathbf{I}}_{\tau_2})$ to $(\mathbf{z}_2, \mathbf{I}_{\tau_2})$.

is shift-invariant, preserves existing structures without introducing artifacts [52]. We examine both blurring and deblurring transformations and find that the blurring process emerges as the more natural choice during derivation. The details for its derivation and justification are provided in Appendix A.4. Accordingly, we present the derivation based on the blurring process. Formally, given an image $\mathbf{I} = \mathbf{I}_0$, we define the progressive Gaussian blurring process $\{\mathbf{I}_t\}_{t=0}^T$ as a series of uniformly sampled images from a continuous process $\{\mathbf{I}_\tau\}_{\tau\in[0,T\Delta]}$ by letting $\mathbf{I}_t := \mathbf{I}_\tau|_{\tau=t\Delta}$, where Δ is the step size between adjacent samples. For $\tau \in (0, T]$,

$$\mathbf{I}_\tau = G_\tau * \mathbf{I}, \tag{9}$$

where $G_{\sigma^2}(x, y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$ denotes the Gaussian blur kernel.

Update rule $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$ Owing to the well-studied analytical properties of Gaussian blur, we can formulate how $\mathbf{c}(\mathbf{I}_t)$ evolves as t increases, by deriving the derivatives of $\mathbf{c}(\mathbf{I}_\tau)$ with respect to τ. For brevity, we demonstrate the detailed derivation process in Appendix A.3, and present only the resulting formulation below.

Proposition 3.1. *Let $\mathbf{c}(\cdot)$ be a projection using 2D orthogonal basis functions $\{\phi_k\}_{k=1}^N$ and $\{\mathbf{I}_\tau\}$ be a continuous Gaussian blurring process. Then, the derivative $\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}(\mathbf{I}_\tau)$ and the update rule $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$ are derived as follows:*

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}(\mathbf{I}_\tau) = \mathbf{A}\mathbf{c}(\mathbf{I}_\tau), \qquad \textit{where } \mathbf{A} = \frac{1}{2}\begin{bmatrix} \langle\nabla^2\phi_1, \phi_1\rangle & \cdots & \langle\nabla^2\phi_N, \phi_1\rangle \\ \langle\nabla^2\phi_1, \phi_2\rangle & \cdots & \langle\nabla^2\phi_N, \phi_2\rangle \\ \vdots & \ddots & \vdots \\ \langle\nabla^2\phi_1, \phi_N\rangle & \cdots & \langle\nabla^2\phi_N, \phi_N\rangle \end{bmatrix}, \tag{10}$$

$$\mathbf{c}(\mathbf{I}_t) = \bar{\mathbf{A}}\mathbf{c}(\mathbf{I}_{t-1}), \qquad \textit{where } \bar{\mathbf{A}} = \delta(\Delta, \mathbf{A}) \textit{ for a discretization method } \delta. \tag{11}$$

Structured state-space regularization From the update rule in Eq. (11), one can naturally derive a regularizing term for a trainable encoder network $\mathcal{E} : \mathbf{I} \mapsto \mathcal{E}(\mathbf{I})$ that maps an image to a latent. Recall that the hidden state of SSMs obtains basis function properties by following the coefficient dynamics $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{c}(\mathbf{I}_t)$, or equivalently, the discretized update rule $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$ (Sec. 3.1)). Since we aim to endow basis-like inductive bias to the encoder output $\mathcal{E}(\mathbf{I})$, we let $\mathcal{E}$ follow the derived update rule over the predefined transformation $\{\mathbf{I}_t\}$:

$$\mathcal{E}(\mathbf{I}_t) = \bar{\mathbf{A}}\mathcal{E}(\mathbf{I}_{t-1}). \tag{12}$$

In general, the above equation does not hold for arbitrary encoder networks, and thus, we can regularize the network with the following regularization loss:

$$\mathcal{L} = d(\mathcal{E}(\mathbf{I}_t), \bar{\mathbf{A}}\mathcal{E}(\mathbf{I}_{t-1})), \tag{13}$$

for some distance measure $d(\cdot, \cdot)$.

3.3 Application to image tokenizers

In this section, we introduce how the regularization loss of Eq. (13) is implemented to train an image tokenizer. Let the tokenizer have a typical autoencoder structure: $f = \mathcal{D} \circ \mathcal{E}$. Since the

tokenizer should retain capability to compress and reconstruct the image, we apply the regularization with probability α, and let f be trained to reconstruct the image otherwise. An illustration of the overall framework is provided in Fig. 2. Given an input RGB image $\mathbf{I} = \mathbf{I}_0 \in \mathbb{R}^{3\times H_0\times W_0}$, we define a continuous blurring sequence $\{\mathbf{I}_\tau\}_{\tau\in[0,\tau_{\text{max}}]}$, where τ_{max} is a hyperparameter that decides the maximum level of blur. Then, we sample two images at different blur levels $0 \le \tau_1 < \tau_2 \le \tau_{\text{max}}$, where $|\tau_1 - \tau_2| = \Delta_{\mathbf{I}}$ for a step size $\Delta_{\mathbf{I}} \in \mathbb{R}_+$. Between the two sampled images, the less blurry image $\mathbf{I}_{\tau_1}$ is passed through the encoder $\mathcal{E}$ to produce $\mathbf{z}_1 := \mathcal{E}(\mathbf{I}_{\tau_1}) \in \mathbb{R}^{C\times H\times W}$ where C, H and W denote the channel, height, and width dimensions of the encoded feature map $\mathcal{E}(\mathbf{I}_{\tau_1})$, respectively. Next, we apply the update rule, *i.e.*, multiply the matrix $\bar{\mathbf{A}} \in \mathbb{R}^{C\times C}$, to the channel dimension of $\mathbf{z}_1$, which we denote as $\hat{\mathbf{z}}_2 := \bar{\mathbf{A}}\mathbf{z}_1$.[2] Since we want $\hat{\mathbf{z}}_2$ to match $\mathbf{z}_2 := \mathcal{E}(\mathbf{I}_{\tau_2})$, we encode $\mathbf{I}_{\tau_2}$ as well to produce its latent $\mathbf{z}_2$. Then, the regularization loss $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \lambda_{\mathbf{z}} d_{\mathbf{z}}(\mathbf{z}_2, \hat{\mathbf{z}}_2) + \lambda_{\mathbf{I}} d_{\mathbf{I}}\big(\mathbf{I}_{\tau_2}, \mathcal{D}(\hat{\mathbf{z}}_2)\big), \tag{14}$$

where $d_{\mathbf{z}}(\cdot,\cdot)$ and $d_{\mathbf{I}}(\cdot,\cdot)$ are distance measures used in latent and pixel spaces, and $\lambda_{\mathbf{z}}$ and $\lambda_{\mathbf{I}}$ are their respective weights. Note that we involve the decoder $\mathcal{D}$ in Eq. (14). Minimizing $d_{\mathbf{z}}(\mathbf{z}_2, \hat{\mathbf{z}}_2)$ alone can lead to learning a trivial encoder $\mathcal{E}$ (*e.g.*, $\mathcal{E} : \mathbf{I} \mapsto \mathbf{0}$) and involving $\mathcal{D}$ is empirically found to effectively mitigate such mode collapse, making it a favorable design in previous work [43, 70]. By minimizing $\mathcal{L}$, we enforce the state-space-like update between the latent representations of equi-distanced images sampled from $\{\mathbf{I}_\tau\}_{\tau\in[0,\tau_{\text{max}}]}$, thereby regularizing the latent features $\{\mathcal{E}(\mathbf{I}_\tau)\}_{\tau\in[0,\tau_{\text{max}}]}$ to follow the dynamics of $\{\mathbf{c}(\mathbf{I}_\tau)\}_{\tau\in[0,\tau_{\text{max}}]}$.

4 Experiments

We evaluate the effectiveness of the proposed regularization by training widely used tokenizers, Flux [10] and Cosmos [2]. Each tokenizer is initialized from their respective pretrained weights and fine-tuned with the reconstruction loss introduced in modern variational autoencoders (VAEs) [21, 67] together with the proposed regularization loss Eq. (14). We further train LightningDiT [60, 82], a latent diffusion model for image generation, on the resulting latent spaces to evaluate their effectiveness for generation tasks.

Datasets We train and evaluate both tokenizers and the diffusion model on ImageNet-1K [18], which contains 1.28M training images and 50K validation images of various animals, objects and scenes. Following the standard practice, images are resized to 256^2 resolution for both training and evaluation.

Implementation details The formulation of the proposed regularization varies by the choice of the basis function. Empirically, we find that using $\mathbf{A}$ derived from the Fourier basis yields the best result. Unless otherwise specified, all reported results correspond to models using Fourier-based $\mathbf{A}$. Also, following prior SSM literature, we set the non-zero values of the matrix $\mathbf{A}$ and discretization parameter Δ to be learnable, using a smaller learning rate than other parameters [26, 27, 29]. For tokenizer training, we adopt most hyperparameters from equivariance-regularized autoencoders [43, 70], including batch size, learning rate and KL regularization coefficients. For generation training, we adopt LightningDiT [82], and train all models for 80K steps, applying classifier-free guidance [35] at inference with a scale of 10.0. Further experimental details are provided in Appendix A.7. Due to spatial constraints, we provide relevant ablation studies in Appendix A.10.

Evaluation We evaluate both reconstruction and generation quality on the ImageNet validation set. Reconstruction quality is measured using Peak Signal-to-Noise Ratio (PSNR) and Fréchet Inception Distance (FID), where FID is computed between original images and their corresponding reconstructions (*i.e.*, rFID). For generation quality, we report generation FID (gFID) and Inception Score (IS), widely used to assess the perceptual realism of high-resolution generative models.

4.1 Generation performance comparison

We compare reconstruction and generation performance across different methods in Table 1, including EQVAE [43] and FT-SE [70], which similarly aim to improve tokenizer's generation-friendliness.

[2] For brevity, we abuse the matrix multiplication notation to refer to this operation. Technically correct notation should be mode-n product: $\mathbf{z}_2 = \mathbf{z}_1 \times_1 \bar{\mathbf{A}} \in \mathbb{R}^{C\times H\times W}$.

Table 1: **Reconstruction and generation performance on ImageNet-1K.** Our method improves generation quality across both Flux and Cosmos with minimal impact on reconstruction quality.

Tokenizer	Reconstruction		Generation (-B/2 / -L/2)	
	PSNR ↑	rFID ↓	gFID ↓	IS ↑
Flux	**31.70**	**0.87**	16.08 / 5.56	89.32 / 184.64
Flux + EQVAE [43]	31.53	0.94	13.95 / 5.37	97.81 / 195.64
Flux + FT-SE [70]	31.62	0.90	13.87 / **5.08**	99.79 / 201.12
Flux + Ours	31.54	0.90	**13.17** / 5.25	**105.99 / 207.14**
Cosmos	25.16	2.88	13.52 / 5.49	112.48 / 230.59
Cosmos + EQVAE [43]	25.20	2.85	11.66 / 5.45	123.54 / 231.70
Cosmos + FT-SE [70]	**25.37**	2.87	12.01 / 5.36	117.97 / 234.18
Cosmos + Ours	25.18	**2.82**	**11.20 / 5.23**	**128.91 / 237.69**



Figure 3: **Generation results comparison using the Flux tokenizer.** With only marginal loss in reconstruction quality, our method improves the generative performance of the image tokenizer.

For fair comparisons, we also fine-tune the Flux and Cosmos tokenizers on the ImageNet-1K training set and use these fine-tuned models to fill the rows for Flux and Cosmos, instead of their original pretrained counterparts.

Table 1 compares our method with the previous work in reconstruction and generation performance. On the Flux tokenizer, our method is on par with or slightly behind other regularizers in reconstruction performance, while yielding larger improvements in generation metrics. The results on the Cosmos tokenizer show a similar trend, except that all listed regularizers show improved reconstruction quality over the baseline. Notably, existing regularizers are designed to directly preserve reconstruction-oriented invariances such as scale or rotation consistency, whereas our regularization sacrifices the reconstruction training iterations by a factor of α. Despite not explicitly optimizing for reconstruction fidelity, our method achieves competitive reconstruction quality while consistently providing stronger downstream generative performance across most metrics, indicating a favorable tradeoff between reconstruction and generation. We provide the qualitative examples of generated images in Fig. 3.

One noteworthy observation is that our method provides complementary improvements over existing approaches that enhance generation, including ones involving vision foundation models [82, 85]. As shown in Table 2, our method consistently yields additional gains when combined with FT-SE [70], VF loss [82], and REPA [85]. For the latter two experiments, we use DINOv2 [58] features as the vision

Table 2: **Complementarity with prior generation improvements.** Our method complements (a) FT-SE, (b) VF loss, and (c) REPA, yielding additional improvements along orthogonal directions.

(a) FT-SE [70]

Tokenizer	PSNR ↑	rFID ↓	gFID ↓	IS ↑
Flux	**31.70**	**0.87**	16.08	89.32
+ FT-SE	31.62	0.90	13.87	99.79
+ Ours	31.54	0.90	13.17	105.99
+ FT-SE + Ours	31.50	0.94	**12.46**	**109.58**

(b) VF loss [82]

Tokenizer	PSNR ↑	rFID ↓	gFID ↓	IS ↑
LDM	**31.03**	0.17	36.83	N/A
+ VF loss	30.71	**0.15**	24.00	N/A
+ Ours	30.34	0.23	26.66	82.43
+ VF loss + Ours	30.37	0.24	**21.78**	**102.37**

(c) REPA [85]

Method	gFID ↓	IS ↑
Flux + LightningDiT	16.08	89.32
+ REPA	9.46	133.87
+ Ours	13.17	105.99
+ REPA + Ours	**8.71**	**142.26**

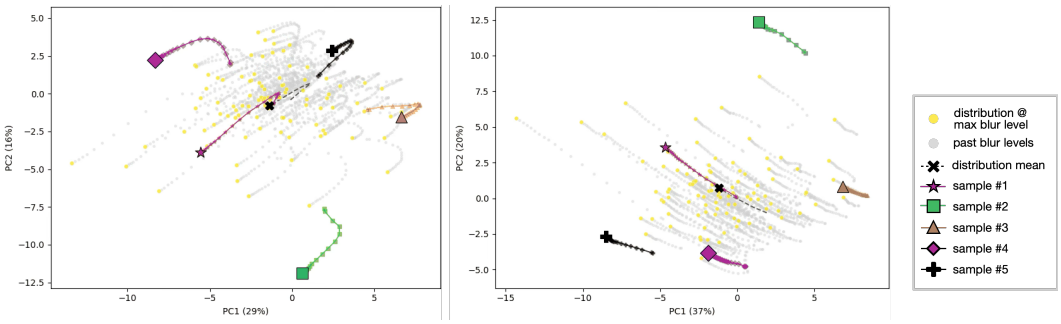


Figure 4: **Latent dynamics before (left) and after (right) the regularization.** Latent trajectories of 100 samples extracted from the ImageNet validation set, projected onto their respective PCA bases. Our regularization simplifies the latent dynamics induced by the blurring operation.

foundation model. Across all three settings, the combined models achieve better generation metrics than either component alone. This consistent behavior across different generation-improvement strategies suggests that our method captures a complementary source of improvement.

4.2 Emerging latent space structures from the regularization

Note that we regularize the latent space by enforcing a basis coefficient's dynamics corresponding to the Gaussian blurring process to the latent feature (Sec. 3.1). To better understand how the proposed regularization shapes the latent space, we analyze the structures that emerge across latent channels, both qualitatively and quantitatively. We organize our analysis around three observations.

(i) Blurring translates into simpler dynamics in the latent space. We randomly sample 100 images from the ImageNet validation set, track their latent representations throughout the image blurring process, and visualize the resulting trajectories using two PCA bases (Fig. 4). Before regularization, the latent trajectories exhibit inconsistency along different blur levels, whereas after applying our regularization, they become notably straighter and more consistent. This implies that our regularization transforms the latent space such that the progressive frequency decay in pixel space corresponds to simpler and more structured dynamics in the latent space.

(ii) Each latent channel encodes what basis coefficients would encode. Since each latent channel is enforced to capture a specific frequency band, we examine this behavior by masking selected channels and decoding remaining latent channels back to pixel space. As we progressively unmask channels from low to high frequencies, the reconstruction correspondingly accumulates coarse-to-fine details, effectively stacking low-to-high frequency components (Left, Fig. 5). We quantitatively examine this behavior by amplifying channels that are grouped by the frequency assigned to each under our derivation (see Appendix A.9 for the grouping details). Specifically, we divide channels into five groups, $\{g_i\}_{i=1}^{5}$, where a larger i denotes a group assigned to a higher frequency, amplify each group individually (*i.e.*, $\mathbf{z}[c] \mapsto 2(\mathbf{z}[c] - \mu(\mathbf{z}[c]) + \mu(\mathbf{z}[c]))$, and identify the frequency band with the largest increase in power (Right, Fig. 5). As shown in Table 3, our regularization progressively organizes the latent channels according to their intended frequency ordering: as the regularization strength α increases, the frequency bands captured by g_i become increasingly aligned with their prescribed ordering.

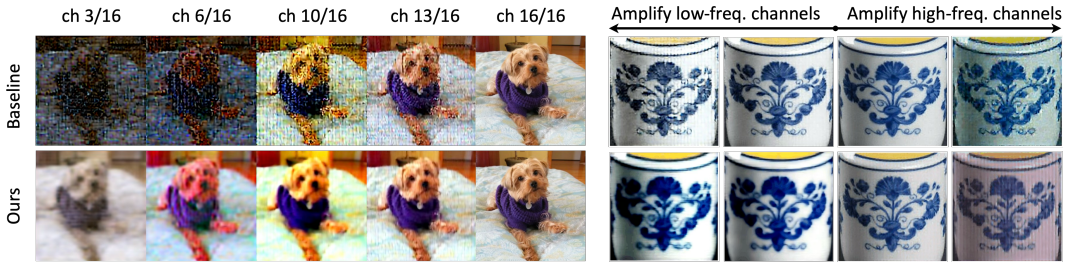


Figure 5: **Latent channels encoding different frequency bands.** (Left) The regularized tokenizer is able to reconstruct recognizable images solely from using three low-frequency channels. (Right) Amplifying each channel and decoding the latent back to pixel space reveals that each channel captures its assigned frequency component as intended.

Table 3: **Frequency bands captured by each channel group with respect to α.** Larger regularization strength α encourages channels to encode in an intended spectral order, leading to more spectrally organized representations.

α	channel groups				
	g_1	g_2	g_3	g_4	g_5
0.0	24.1	76.9	2.2	107.1	10.9
0.05	1.5	73.1	47.2	81.2	81.4
0.1	1.5	29.3	47.2	81.6	81.0
0.25	1.5	26.8	47.0	79.8	81.2
0.5	1.5	26.1	46.6	82.6	81.2

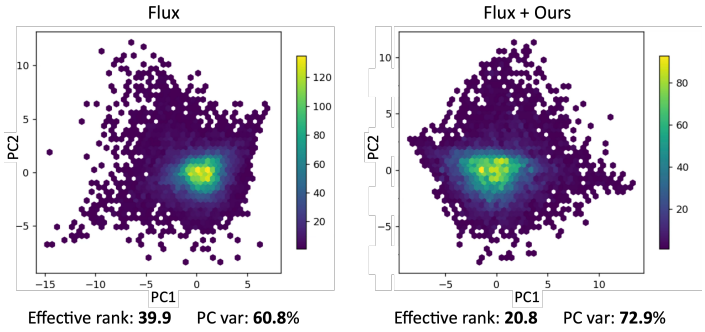


Figure 6: **Latent space dimensionality comparison.** Our method reduces the effective rank while increasing the variance captured by the leading principal components of the latent space.

(iii) Dimensionality of the latent space decreases significantly. Fig. 6 shows that the effective rank of the latent space is nearly halved after the regularization, while the leading principal components capture a larger portion of the latent variance. This suggests that the regularized latent space can be well approximated using fewer basis directions, indicating that the regularizer concentrates the latent information into a more compact set of dimensions.

We hypothesize that the enhanced generation-friendliness induced by our regularizer can be attributed to the three aforementioned properties, each supported by related prior studies. Specifically, (i) well-defined equivariance between the pixel and latent spaces has been shown to improve generation [43, 70]; (ii) incorporating frequency-aware structure into the latent space potentially benefits the generation process [19, 66]; and (iii) latent distributions that can be represented with fewer Gaussian modes [15] or concentrate their energy into a smaller number of frequency bands [90] have been associated with improved generation.

5 Conclusion

We introduced structured state-space regularization, a framework that extends state-space models (SSMs) to regularize image tokenizers toward generation-friendly representations. By reframing SSMs as a combination of basis projection and input transformation, we derive a regularization objective that transfers their frequency-aware structure to visual tokenizers. This formulation bridges state-space modeling and visual representation learning, allowing tokenizers to learn latent representations that effectively capture spectral structures of images. Experiments show that our method produces latent spaces that better support generative modeling while minimally sacrificing reconstruction quality. By offering a new perspective on leveraging SSMs for visual representation, this paper paves the way for integrating state-space modeling principles into future generative and representation learning frameworks.

Acknowledgment

This work was supported by the NRF grant (RS-2026-25618525), the IITP grants (RS-2022-II220290, RS-2022-II220926, RS-2024-00509258, RS-2024-00469482 RS-2024-00457882, RS-2026-25518317, RS-2019-II191906), the Advanced GPU Utilization Support Program (02-26-01-0372), and the AI Computing Support Project for R & D (RS-2026-25505492) funded by Ministry of Science and ICT, Korea.

References

[1] Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1692–1700. IEEE (2018)

[2] Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)

[3] Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE transactions on Computers **100**(1), 90–93 (2006)

[4] Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15619–15629 (2023)

[5] Ballard, D.H., Brown, C.M.: Computer vision. Prentice Hall Professional Technical Reference (1982)

[6] Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15791–15801 (2025)

[7] Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M., Ballas, N.: Revisiting feature prediction for learning visual representations from video. arXiv preprint arXiv:2404.08471 (2024)

[8] Baron, E., Zimerman, I., Wolf, L.: A 2-dimensional state space layer for spatial inductive bias. In: The Twelfth International Conference on Learning Representations (2024)

[9] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)

[10] Black Forest Labs: Flux. `https://github.com/black-forest-labs/flux` (2023)

[11] Boutell, T.: Png (portable network graphics) specification version 1.0. Tech. rep. (1997)

[12] Castleman, K.R.: Digital image processing. Prentice Hall Professional Technical Reference (1979)

[13] Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11315–11325 (2022)

[14] Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3557–3567. IEEE (2021)

[15] Chen, H., Han, Y., Chen, F., Li, X., Wang, Y., Wang, J., Wang, Z., Liu, Z., Zou, D., Raj, B.: Masked autoencoders are effective tokenizers for diffusion models. In: Forty-second International Conference on Machine Learning (2025)

[16] Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. arXiv preprint arXiv:2209.14687 (2022)

[17] Chung, H., Sim, B., Ryu, D., Ye, J.C.: Improving diffusion models for inverse problems using manifold constraints. Advances in Neural Information Processing Systems **35**, 25683–25696 (2022)

[18] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)

[19] Dieleman, S.: Diffusion is spectral autoregression (2024), https://sander.ai/2024/09/02/spectral-autoregression.html

[20] Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)

[21] Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)

[22] Esteves, C., Suhail, M., Makadia, A.: Spectral image tokenizer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17181–17190 (2025)

[23] Fu, M., Liu, H., Yu, Y., Chen, J., Wang, K.: Dw-gan: A discrete wavelet transform gan for nonhomogeneous dehazing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 203–212 (2021)

[24] Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: First conference on language modeling (2024)

[25] Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C.: Hippo: Recurrent memory with optimal polynomial projections. Advances in neural information processing systems **33**, 1474–1487 (2020)

[26] Gu, A., Goel, K., Gupta, A., Ré, C.: On the parameterization and initialization of diagonal state space models. Advances in Neural Information Processing Systems **35**, 35971–35983 (2022)

[27] Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)

[28] Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. Advances in neural information processing systems **34**, 572–585 (2021)

[29] Gu, A., Johnson, I., Timalsina, A., Rudra, A., Ré, C.: How to train your hippo: State space models with generalized orthogonal basis projections. arXiv preprint arXiv:2206.12037 (2022)

[30] Gupta, A., Gu, A., Berant, J.: Diagonal state spaces are as effective as structured state spaces. Advances in Neural Information Processing Systems **35**, 22982–22994 (2022)

[31] Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603 (2019)

[32] Hasani, R., Lechner, M., Wang, T.H., Chahine, M., Amini, A., Rus, D.: Liquid structural state-space models. In: The Eleventh International Conference on Learning Representations (2023)

[33] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)

[34] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

[35] Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)

[36] Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868 (2022)

[37] Hu, V.T., Baumann, S.A., Gui, M., Grebenkova, O., Ma, P., Fischer, J., Ommer, B.: Zigma: A dit-style zigzag mamba diffusion model. In: European conference on computer vision. pp. 148–166. Springer (2024)

[38] Hummel, R., Kimia, B., Zucker, S.: Gaussian blur and the heat equation: forward and inverse solutions. In: Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 668–671 (1985)

[39] Jiang, X., Zhang, X., Gao, N., Deng, Y.: When fast fourier transform meets transformer for image restoration. In: European conference on computer vision. pp. 381–402. Springer (2024)

[40] Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6007–6017 (2023)

[41] Kim, D., Seo, G., Lee, J., Cho, M., Kwak, S.: Planning in 8 tokens: A compact discrete tokenizer for latent world model. arXiv preprint arXiv:2603.05438 (2026)

[42] Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2426–2435 (2022)

[43] Kouzelis, T., Ioannis, K., Spyros, G., Nikos, K.: Eq-vae: Equivariance regularized latent space for improved generative image modeling. In: arxiv (2025)

[44] Lee, D., Kim, C., Kim, S., Cho, M., Han, W.S.: Autoregressive image generation using residual quantization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11523–11532 (2022)

[45] Lee, J., Kwak, S.: Exploring state-space models for data-specific neural representations. In: The Fourteenth International Conference on Learning Representations (2026)

[46] Leng, X., Singh, J., Hou, Y., Xing, Z., Xie, S., Zheng, L.: Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18262–18272 (2025)

[47] Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., Qiao, Y.: Videomamba: State space model for efficient video understanding. In: European Conference on Computer Vision. pp. 237–255. Springer (2025)

[48] Li, S., Singh, H., Grover, A.: Mamba-nd: Selective state space modeling for multi-dimensional data. In: European Conference on Computer Vision. pp. 75–92. Springer (2024)

[49] Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishnan, D.: Mage: Masked generative encoder to unify representation learning and image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2142–2152 (2023)

[50] Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. Advances in Neural Information Processing Systems **37**, 56424–56445 (2024)

[51] Liang, D., Zhou, X., Xu, W., Zhu, X., Zou, Z., Ye, X., Tan, X., Bai, X.: Pointmamba: A simple state space model for point cloud analysis. Advances in neural information processing systems **37**, 32653–32677 (2024)

[52] Lindeberg, T.: Scale-space theory: A basic tool for analyzing structures at different scales. Journal of applied statistics **21**(1-2), 225–270 (1994)

[53] Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y.: Vmamba: Visual state space model. Advances in neural information processing systems **37**, 103031–103063 (2024)

[54] Mehta, H., Gupta, A., Cutkosky, A., Neyshabur, B.: Long range language modeling via gated state spaces. In: The Eleventh International Conference on Learning Representations (2023)

[55] Nguyen, E., Goel, K., Gu, A., Downs, G., Shah, P., Dao, T., Baccus, S., Ré, C.: S4nd: Modeling images and videos as multidimensional signals with state spaces. Advances in neural information processing systems **35**, 2846–2861 (2022)

[56] Ning, M., Li, M., Su, J., Jia, H., Liu, L., Beneš, M., Chen, W., Salah, A.A., Ertugrul, I.O.: Dctdiff: Intriguing properties of image generative modeling in the dct space. arXiv preprint arXiv:2412.15032 (2024)

[57] Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Advances in neural information processing systems **29** (2016)

[58] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

[59] Pan, Q., Zhang, L., Dai, G., Zhang, H.: Two denoising methods by wavelet transform. IEEE transactions on signal processing **47**(12), 3401–3406 (1999)

[60] Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)

[61] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence **12**(7), 629–639 (2002)

[62] Phillips, A., Seror, T., Hutchinson, M., De Bortoli, V., Doucet, A., Mathieu, E.: Spectral diffusion processes. arXiv preprint arXiv:2209.14125 (2022)

[63] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)

[64] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)

[65] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International conference on machine learning. pp. 8821–8831. Pmlr (2021)

[66] Rissanen, S., Heinonen, M., Solin, A.: Generative modelling with inverse heat dissipation. arXiv preprint arXiv:2206.13397 (2022)

[67] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)

[68] Shen, J.: Efficient spectral-galerkin method i. direct solvers of second-and fourth-order equations using legendre polynomials. SIAM Journal on Scientific Computing **15**(6), 1489–1505 (1994)

[69] Shen, J.: Efficient spectral-galerkin method ii. direct solvers of second-and fourth-order equations using chebyshev polynomials. SIAM Journal on Scientific Computing **16**(1), 74–87 (1995)

[70] Skorokhodov, I., Girish, S., Hu, B., Menapace, W., Li, Y., Abdal, R., Tulyakov, S., Siarohin, A.: Improving the diffusability of autoencoders. In: Forty-second International Conference on Machine Learning (2025)

[71] Smith, J.T., Warrington, A., Linderman, S.: Simplified state space layers for sequence modeling. In: The Eleventh International Conference on Learning Representations (2023)

[72] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. pmlr (2015)

[73] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)

[74] Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., Yuan, Z.: Autoregressive model beats diffusion: Llama for scalable image generation. CoRR (2024)

[75] Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. Advances in neural information processing systems **37**, 84839–84865 (2024)

[76] Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. Advances in neural information processing systems **34**, 11287–11302 (2021)

[77] Vandewalle, P., Süsstrunk, S., Vetterli, M.: A frequency domain approach to registration of aliased images with application to super-resolution. EURASIP journal on advances in signal processing **2006**(1), 071459 (2006)

[78] Voelker, A., Kajić, I., Eliasmith, C.: Legendre memory units: Continuous-time representation in recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 15544–15553 (2019)

[79] Wallace, G.K.: The jpeg still picture compression standard. Communications of the ACM **34**(4), 30–44 (1991)

[80] Wang, X., Zhao, H., Lu, Y., Zhou, K., Ma, L., He, K.: Minit2i: A minimalist baseline for text-to-image generation (2026), `https://peppaking8.github.io/#/post/minit2i`

[81] Weickert, J., et al.: Anisotropic diffusion in image processing, vol. 1. Teubner Stuttgart (1998)

[82] Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15703–15712 (2025)

[83] Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y.L., Zhu, C., Xiang, J., et al.: World action models are zero-shot policies. arXiv preprint arXiv:2602.15922 (2026)

[84] Yu, A., Lyu, D., Lim, S.H., Mahoney, M.W., Erichson, N.B.: Tuning frequency bias of state space models. arXiv preprint arXiv:2410.02035 (2024)

[85] Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940 (2024)

[86] Zadeh, L., Desoer, C.: Linear system theory: the state space approach. Courier Dover Publications (2008)

[87] Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR (2022)

[88] Zhang, J., Nguyen, A.T., Han, X., Trinh, V.Q.H., Qin, H., Samaras, D., Hosseini, M.S.: 2dmamba: Efficient state space model for image representation with applications on giga-pixel whole slide image classification. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 3583–3592 (2025)

[89] Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690 (2025)

[90] Zhong, T., Tian, X., Wang, X., Tao, X., Wan, P.: Diffusing in the right space: A systematic study of latent diffusability. arXiv preprint arXiv:2606.03578 (2026)

[91] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. In: International Conference on Machine Learning. pp. 62429–62442. PMLR (2024)

Structured State-Space Regularization for Generation-Friendly Image Tokenization

A Appendix

We provide supporting details in the Appendix, omitted in the main manuscript due to space constraints.

A.1 Related work

A.1.1 Tokenizers for diffusion models

Modern generative models primarily employ the diffusion framework [34], which models the distribution transformation from Gaussian noise to sample distribution [73]. A common practice in diffusion models is to employ tokenizers [67] since performing the denoising process in the high-dimensional pixel space is prohibitively costly. Specifically, a tokenizer is trained to map input data into a latent space and reconstruct them back, offering a compact low-dimensional representation of each sample that is utilized as input for diffusion models instead of raw visual data. Examples of such a tokenizer include VQVAE [21], SDVAE [67], Flux [10], Cosmos [2], and CogVideo [36].

A.1.2 Generation-friendliness of tokenizers

Recent studies further explore desirable properties of autoencoders to enhance the diffusion process, such as enforcing rotation or scale equivariance between the pixel and latent spaces [43, 70], constraining the latent space with fewer Gaussian mixture modes [15], or incorporating features from vision foundation models [46, 82, 89]. Our work also introduces a novel form of equivariance between the pixel and latent spaces that is effective for diffusion modeling, aligning Gaussian blurring in the pixel space with the corresponding operation in the 2D orthogonal basis coefficient space.

A.1.3 State space models (SSMs)

1D SSMs SSMs have steadily evolved into a framework for sequence modeling, based on hidden state dynamics defined by a transition matrix. One of the pioneering studies on SSMs is HiPPO [25], which constructs a compact representation of 1D sequences through projection onto orthogonal polynomials and derives the *HiPPO matrix* $\mathbf{A}$, allowing the update of the compact representation that integrates new information in an online manner. LSSL [28] proposes a sequence-to-sequence layer based on the SSM formulation with the HiPPO matrix and establishes a grounding for its parameterization. Subsequently, several approaches [26, 27, 30] have introduced ways to effectively parameterize the state transition matrix $\mathbf{A}$, while other stream of work has proposed different implementations of the SSM framework [32, 54, 71]. The most well-known work is Mamba [24], which introduces input-dependent SSM parameters with a selective scan mechanism. It has given rise to numerous variants that apply SSM framework to various modalities [37, 47, 51, 53, 91].

Multi-dimensional SSMs Although SSMs are originally designed for 1D inputs, there have been several attempts to extend SSMs to multi-dimensional data. While numerous studies have introduced sophisticated scanning algorithms that flatten the multi-dimensional input into 1D sequence to enable the application of SSMs [48, 88, 91], S4ND [55] and 2D-SSM [8] take more theoretical approaches to derive the state-transition equation that works on N-dimensional inputs. Specifically, they both designate a starting point (*e.g.*, a corner of an image) and let the input tokens closer from the starting point are first integrated to the hidden state, building a multi-dimensional sequential process constructing a multi-dimensional input. Our method also introduces a new SSM formulation that works on 2D inputs, but provides a different view on the input transformation: rather than building a sequence within a single image, the initial hidden state encodes the original input image and evolves to represent progressively blurred versions. We then regularize the hidden state updates based on this blurring transformation, enforcing them to follow the dynamics of basis functions—thereby endowing the hidden states with compressive properties.

A.2 Combination of $\mathbf{c}(\cdot)$ and $\theta(\cdot)$ that yields tractable update rule

In Sec. 3.1, we mention that some choice of basis functions makes it easier to yield a tractable update rule. Here, the *tractable* update rule refers to having a property where we can directly update the compressed representation to reflect the transformation on the original input *without having to decompress the input*. Basis projections are favorable in terms of this tractability, since they are in general differentiable, and we can compute the dynamics of the resulting coefficients by differentiating them with respect to t. For example, one can come up with different compressed representations other than basis projection, such as PNG [11] or JPEG [79]. However, PNG compression is based on entropy-coding, which does not allow direct image-domain linear algebra that corresponds to operations on PNG bitstream. Thus, PNG necessitates decompression to update the compressed representation, making the update rule intractable. On the other hand, JPEG is essentially an operation of splitting an image into 8×8 blocks and applying discrete cosine transform [3] to each block. This indicates that JPEG representation holds frequency components of each block of the image, so the image transformation $\theta : \mathbf{I}_{t-1} \mapsto \mathbf{I}_t$ must be related to each block's frequency domain. Thus, a natural choice of $\{\mathbf{I}_t\}$ corresponding to JPEG is to apply transformation that involves frequency domain, *e.g.*, gradual blur, on each patch.

As we have seen from the last example, we can conclude that this tractability is determined by both basis projection $\mathbf{c}(\cdot)$ and input transformation $\theta(\cdot)$. In general, we find that choosing a discontinuous transformation such as downsampling or a stochastic transformation such as noising makes it hard to derive derivatives of $\mathbf{c}(\cdot)$. Exploring plausible combinations of $\big(\mathbf{c}(\cdot), \theta(\cdot)\big)$ that result in a nice form of state update rule would be another interesting direction.

A.3 Derivation of the update rule $\mathbf{c}(\mathbf{I}_{t-1} \mapsto \mathbf{I}_t)$

Let the k-th element of the vector $\mathbf{c}(\mathbf{I}_\tau) \in \mathbb{R}^N$ be $\mathbf{c}_k(\mathbf{I}_\tau)$, then we can derive its derivative as follows:

$$\mathbf{c}_k(\mathbf{I}_\tau) = \langle \mathbf{I}_\tau, \phi_k \rangle = \langle G_\tau * \mathbf{I}, \phi_k \rangle, \tag{15}$$

$$\Rightarrow \ \frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}_k(\mathbf{I}_\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\langle G_\tau * \mathbf{I}, \phi_k \rangle = \Big\langle \frac{\partial}{\partial \tau}(G_\tau * \mathbf{I}), \phi_k \Big\rangle. \tag{16}$$

Meanwhile, Gaussian blur kernel G_τ applied to an image $\mathbf{I}$ is known to be the solution to the heat equation [61], which yields

$$\Big\langle \frac{\partial}{\partial \tau}(G_\tau * \mathbf{I}), \phi_k \Big\rangle = \Big\langle \frac{1}{2}\nabla^2(G_\tau * \mathbf{I}), \phi_k \Big\rangle, \tag{17}$$

where ∇^2 is the Laplacian operator. Since we can express $G_\tau * \mathbf{I}$ by the weighted sum using bases $\{\phi_n\}_{n=1}^N$ and coefficients $\big\{\mathbf{c}_n(G_\tau * \mathbf{I})\big\}_{n=1}^N = \big\{\mathbf{c}_n(\mathbf{I}_\tau)\big\}_{n=1}^N$,

$$\Big\langle \frac{1}{2}\nabla^2(G_\tau * \mathbf{I}), \phi_k \Big\rangle = \Big\langle \frac{1}{2}\sum_{n=1}^N \mathbf{c}_n(\mathbf{I}_\tau)\nabla^2\phi_n, \phi_k \Big\rangle \tag{18}$$

$$= \frac{1}{2}\sum_{n=1}^N \mathbf{c}_n(\mathbf{I}_\tau)\big\langle \nabla^2\phi_n, \phi_k \big\rangle. \tag{19}$$

Thus, gathering all together yields:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}_k(\mathbf{I}_\tau) = \frac{1}{2}\sum_{n=1}^N \mathbf{c}_n(\mathbf{I}_\tau)\Big\langle \nabla^2\phi_n, \phi_k \Big\rangle. \tag{20}$$

Aggregating $\mathbf{c}_k$ for all $k \in \{1, 2, \ldots, N\}$ gives:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}(\mathbf{I}_\tau) = \frac{1}{2}\begin{bmatrix} \langle \nabla^2\phi_1, \phi_1 \rangle & \cdots & \langle \nabla^2\phi_N, \phi_1 \rangle \\ \langle \nabla^2\phi_1, \phi_2 \rangle & \cdots & \langle \nabla^2\phi_N, \phi_2 \rangle \\ \vdots & \ddots & \vdots \\ \langle \nabla^2\phi_1, \phi_N \rangle & \cdots & \langle \nabla^2\phi_N, \phi_N \rangle \end{bmatrix}\mathbf{c}(\mathbf{I}_\tau) \tag{21}$$

$$:= \mathbf{A}\mathbf{c}(\mathbf{I}_\tau). \tag{22}$$

Based on the derivative of $\mathbf{c}(\mathbf{I}_\tau)$, we can apply discretization to define the update rule $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$. We adopt either Euler or ZOH discretization method depending on the choice of the basis functions $\{\phi_n\}_{n=1}^N$, following the common practice in modern SSMs [24, 26, 30, 71]. For a discretization step Δ, we get

$$\text{(Euler)} \quad \mathbf{c}(\mathbf{I}_t) = (I + \mathbf{A}\Delta)\mathbf{c}(\mathbf{I}_{t-1}), \tag{23}$$

$$\text{(ZOH)} \quad \mathbf{c}(\mathbf{I}_t) = e^{\mathbf{A}\Delta}\mathbf{c}(\mathbf{I}_{t-1}), \tag{24}$$

which describe the update rules that corresponds to the choice of 2D basis projection $\mathbf{c}(\cdot)$ and Gaussian blurring process $\theta(\cdot)$. Following the convention of Eq. (6), we denote $\bar{\mathbf{A}} := \delta(\Delta, \mathbf{A})$ for a discretization method δ, and write the update rule after discretization as:

$$\mathbf{c}(\mathbf{I}_t) = \bar{\mathbf{A}}\mathbf{c}(\mathbf{I}_{t-1}). \tag{25}$$

We derive the matrix $\mathbf{A}$ for Fourier bases as well as for several families of orthogonal polynomials including Chebyshev, Legendre, and Hermite. The detailed derivation and results for each basis are provided in Appendix A.6.

A.4 Derivation based on Gaussian deblurring process

One can derive the update rule based on Gaussian deblurring process by letting the input image $\mathbf{I} := \mathbf{I}_T$. For $\tau \in (0, T]$, we adjust $\mathbf{I}_\tau$ of Eq. (9) as follows:

$$\mathbf{I}_\tau = G_{T-\tau} * \mathbf{I}. \tag{26}$$

Then, the derivative of $\mathbf{c}_k(\mathbf{I}_\tau)$ becomes:

$$\mathbf{c}_k(\mathbf{I}_\tau) = \langle \mathbf{I}_\tau, \phi_k \rangle = \langle G_{T-\tau} * \mathbf{I}, \phi_k \rangle \tag{27}$$

$$\Rightarrow \ \frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}_k(\mathbf{I}_\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\langle G_{T-\tau} * \mathbf{I}, \phi_k \rangle = \Big\langle \frac{\partial}{\partial \tau}(G_{T-\tau} * \mathbf{I}), \phi_k \Big\rangle = -\Big\langle \frac{\partial}{\partial s}(G_s * \mathbf{I}), \phi_k \Big\rangle. \tag{28}$$

Applying the same logic from Sec 3.2 yields

$$\Big\langle \frac{\partial}{\partial s}(G_s * \mathbf{I}), \phi_k \Big\rangle = \Big\langle \frac{1}{2}\nabla^2(G_s * \mathbf{I}), \phi_k \Big\rangle, \tag{29}$$

where ∇^2 is the Laplacian operator. Substituting $G_s * \mathbf{I}$ with the linear combination of bases $\{\phi_n\}_{n=1}^N$ using coefficients $\mathbf{c}_n(G_s * \mathbf{I}) = \mathbf{c}_n(G_{T-\tau} * \mathbf{I}) = \mathbf{c}_n(\mathbf{I}_\tau)$ gives:

$$\Big\langle \frac{1}{2}\nabla^2(G_s * \mathbf{I}), \phi_k \Big\rangle = \Big\langle \frac{1}{2}\sum_{n=1}^N \mathbf{c}_n(\mathbf{I}_\tau)\nabla^2\phi_n, \phi_k \Big\rangle = \frac{1}{2}\sum_{n=1}^N \mathbf{c}_n(\mathbf{I}_\tau)\big\langle \nabla^2\phi_n, \phi_k \big\rangle. \tag{30}$$

Thus, gathering all together yields:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}_k(\mathbf{I}_\tau) = -\frac{1}{2}\sum_{n=1}^N \mathbf{c}_n(\mathbf{I}_\tau)\Big\langle \nabla^2\phi_n, \phi_k \Big\rangle \tag{31}$$

$$\Rightarrow \frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{c}(\mathbf{I}_\tau) = -\frac{1}{2}\begin{bmatrix} \langle \nabla^2\phi_1, \phi_1 \rangle & \langle \nabla^2\phi_2, \phi_1 \rangle & \cdots & \langle \nabla^2\phi_N, \phi_1 \rangle \\ \langle \nabla^2\phi_1, \phi_2 \rangle & \langle \nabla^2\phi_2, \phi_2 \rangle & \cdots & \langle \nabla^2\phi_N, \phi_2 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \nabla^2\phi_1, \phi_N \rangle & \langle \nabla^2\phi_2, \phi_N \rangle & \cdots & \langle \nabla^2\phi_N, \phi_N \rangle \end{bmatrix}\mathbf{c}(\mathbf{I}_\tau) \tag{32}$$

$$:= -\mathbf{A}\mathbf{c}(\mathbf{I}_\tau), \tag{33}$$

which is a sign-reversed version of the original dynamics introduced in Eq. (10). For a discretization step Δ, we get

$$\text{(Euler)} \quad \mathbf{c}(\mathbf{I}_t) = (I - \mathbf{A}\Delta)\mathbf{c}(\mathbf{I}_{t-1}), \tag{34}$$

$$\text{(ZOH)} \quad \mathbf{c}(\mathbf{I}_t) = e^{-\mathbf{A}\Delta}\mathbf{c}(\mathbf{I}_{t-1}), \tag{35}$$

Adopting the discretization function $\delta(\cdot, \cdot)$ from Sec. 3.1, we obtain the update of the deblurring process:

$$\mathbf{c}(\mathbf{I}_t) = \delta(-\Delta, \mathbf{A})\mathbf{c}(\mathbf{I}_{t-1}) \tag{36}$$

$$:= \overline{\mathbf{A}}_{\text{deblur}}\mathbf{c}(\mathbf{I}_{t-1}). \tag{37}$$

A.4.1 What makes it inadequate to apply the deblurring process

Since the original dynamics based on the Gaussian blurring process (Eq. (11)) does not differ much from the update introduced in Eq. (37), the deblurring process may seem straightforwardly applicable to the current framework. However, two critical issues arise when employing the Gaussian deblurring process instead of the blurring process. First, Gaussian deblurring is fundamentally an *ill-posed* problem: although we obtain the derivative of coefficients $\mathbf{c}(\mathbf{I}_\tau)$ with respect to τ, predicting $\mathbf{c}(\mathbf{I}_{\tau_0+\Delta})$ from $\mathbf{c}(\mathbf{I}_{\tau_0})$ for a timestep τ_0 and a step size Δ is not a well-defined problem unless the input image has a special constraint, such as being exactly identical to a 2D polynomial of degree N [38]. On the other hand, the blurring process is a *deterministic* process, where having the coefficients of the current timestep τ_0 already suffices to provide a fixed trajectory of $\mathbf{c}(\mathbf{I}_\tau)|_{\tau>\tau_0}$. This difference ensures the update of the blurring process Eq. (11) can accurately predict the next coefficients given a sufficiently small step size Δ, whereas Eq. (37) cannot. One may introduce an additional information gap $u_t = \mathbf{I}_t - \mathbf{I}_{t-1}$ to modify the deblurring process into a deterministic process, making the formulation more coherent with the original SSM equation Eq. (4). Yet, under the same assumption of occupying a 2D orthogonal basis projection $\mathbf{c}(\cdot)$, the update becomes very trivial due to the linearity of basis functions:

$$\mathbf{c}(\mathbf{I}_t) = \mathbf{c}(\mathbf{I}_{t-1} + u_t) = \mathbf{c}(\mathbf{I}_{t-1}) + \mathbf{c}(u_t), \tag{38}$$

which turns into a feature dynamics expressed as $f(\mathbf{I}_t) = f(\mathbf{I}_{t-1}) + f(u_t)$ for a feature extractor f. Here, f inherits one of the fundamental properties of the basis projection $\mathbf{c}(\cdot)$, linearity, but it is not a unique property of basis projection and f hardly achieves our goal of obtaining frequency awareness and compactness via mimicking the behavior of $\mathbf{c}(\cdot)$. Choosing a different pair of $\big(\mathbf{c}(\cdot), \phi(\cdot)\big)$ may lead to a different conclusion and yield a formulation that is more coherent to the original 1D SSMs, but for now we leave it for future work. Second, when applied to a tokenizer's encoder $\mathcal{E}$, Eq. (37) hinders encoder's ablity to capture details. Note that the amount of information in the blurred image's feature $f(\mathbf{I}_{\text{blurred}})$ cannot be amplified by simply multiplying with $\bar{\mathbf{A}}$ to match that of $f(\mathbf{I}_{\text{sharp}})$. Since the regularization loss based on Eq. (37) forces $f(\mathbf{I}_{\text{sharp}})$ to match $\bar{\mathbf{A}} f(\mathbf{I}_{\text{blurred}})$ during training, f ends up encoding only as much information as the blurred input, even when given a sharp image.

A.5 Relation to the heat equation

The heat equation is a partial differential equation that illustrates thermodynamics. Formally, let $T(x, y, t) : [0, W] \times [0, H] \times [0, \infty) \to \mathbb{R}$ be a surface function that indicates a temperature at the point (x, y) at time t. Then, the heat equation is defined as follows:

$$\frac{\partial T}{\partial t} = \alpha\Big(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\Big) := \alpha \nabla^2 T, \tag{39}$$

where α is a positive constant that decides the speed of heat diffusion. Let $T(x, y, 0)$ be the image $\mathbf{I}(x, y)$ and $\frac{\partial T}{\partial x}\big|_{x\in\{0,W\}} = \frac{\partial T}{\partial y}\Big|_{y\in\{0,H\}} = 0$, then its known solution $u(x, y, t)$ is:

$$u(x, y, t) = G_t * \mathbf{I}, \tag{40}$$

where $G_{\sigma^2}(x, y) = \frac{1}{4\alpha\pi\sigma^2} \exp(-\frac{x^2+y^2}{4\alpha\sigma^2})$. Intuitively, if the pixel intensity at coordinate (x, y) is viewed as temperature, then the diffusion of heat over time corresponds precisely to the Gaussian-blurred image. Hence, the rightmost term $G_s * \mathbf{I}$ of Eq. (16) satisfies $\frac{\partial T}{\partial t} = \frac{1}{2}\Big(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\Big) := \frac{1}{2}\nabla^2 T$, which explains the equality in Eq. (17).

A.6 Derivation of A matrices

Let us restate the state transition matrix $\mathbf{A}$ for reference:

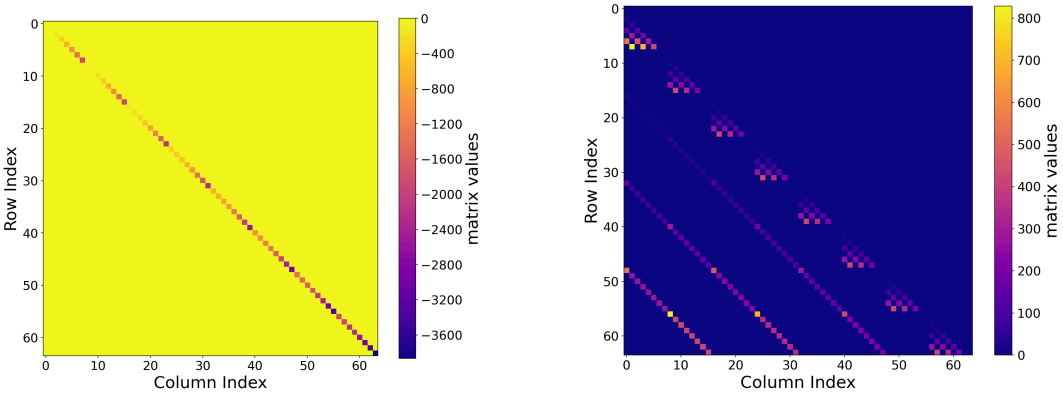
$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} \langle \nabla^2\phi_1, \phi_1\rangle & \langle \nabla^2\phi_2, \phi_1\rangle & \cdots & \langle \nabla^2\phi_N, \phi_1\rangle \\ \langle \nabla^2\phi_1, \phi_2\rangle & \langle \nabla^2\phi_2, \phi_2\rangle & \cdots & \langle \nabla^2\phi_N, \phi_2\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \nabla^2\phi_1, \phi_N\rangle & \langle \nabla^2\phi_2, \phi_N\rangle & \cdots & \langle \nabla^2\phi_N, \phi_N\rangle \end{bmatrix}. \tag{41}$$

Here, $\{\phi_n\}_{n=1}^N = \{\phi_{w,h}\}_{(w,h)\in\{0,1,\ldots,W-1\}\times\{0,1,\ldots,H-1\}}$ is a set of 2D basis functions. For each index n in $\{1, 2, \ldots N\}$, we specifically define the mapping function ι between two index sets

$\{1, 2, \ldots, N\}$ and $\{(0, 0), \ldots, (W-1, H-1)\}$ as follows:

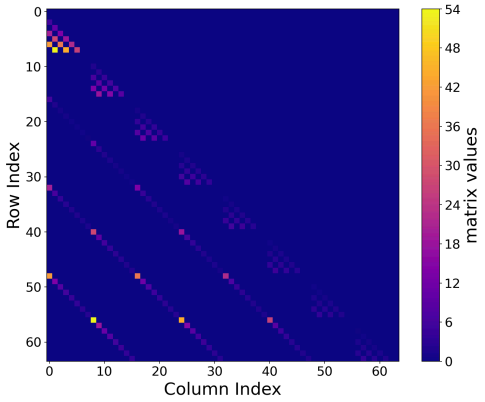
$$\iota : n \mapsto \big((n-1) \mod W, \lfloor \frac{n-1}{W} \rfloor\big). \tag{42}$$

As we obtain a different form of $\mathbf{A}$ depending on the choice of the basis function ϕ, we introduce derivations for different basis choices: Fourier, Chebyshev polynomial, Legendre polynomial, and Hermite polynomial. Since deriving $\mathbf{A}$ is equivalent to computing $\langle \phi_m, \nabla^2 \phi_n \rangle$ (or equivalently $\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle$), we focus on deriving the inner product between a basis function and the second derivative of another basis function. We additionally display each matrix in Fig. 7.

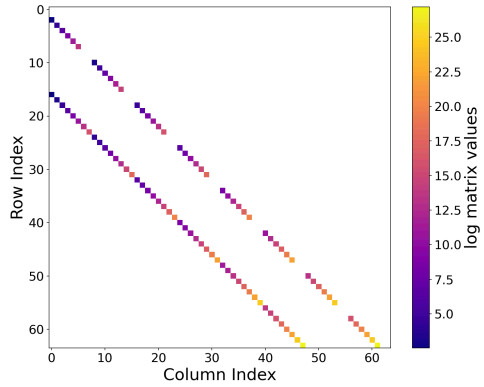


(a) Visualization of Fourier $\mathbf{A}$.

(b) Visualization of Chebyshev $\mathbf{A}$.



(c) Visualization of Legendre $\mathbf{A}$.



(d) Visualization of Hermite $\mathbf{A}$. Due to the exponential scale of matrix values, log values are displayed. Zero values marked white.

Figure 7: **Visualization of derived A matrices.** We set $H = W = 8$. Every matrix we derived is very sparse, which enables efficient matrix-vector multiplication. To maintain training stability, we normalize the matrix to have maximum absolute value of 1.

A.6.1 Fourier A

Note that a 2D Fourier basis defined on $[0, W] \times [0, H]$ is defined as follows:

$$\phi_{w,h}(x, y) = \exp\Big(2\pi i \big(\frac{wx}{W} + \frac{hy}{H}\big)\Big). \tag{43}$$

We obtain its derivative and second derivative with respect to x as:

$$\frac{\partial \phi_{w,h}(x,y)}{\partial x} = \frac{2\pi i w}{W} \exp\Big(2\pi i\big(\frac{wx}{W} + \frac{hy}{H}\big)\Big), \tag{44}$$

$$\begin{aligned}\frac{\partial^2 \phi_{w,h}(x,y)}{\partial x^2} &= -\frac{4\pi^2 w^2}{W^2} \exp\Big(2\pi i\big(\frac{wx}{W} + \frac{hy}{H}\big)\Big) \\ &= -\frac{4\pi^2 w^2}{W^2} \phi_{w,h}(x,y).\end{aligned} \tag{45}$$

Therefore, $\nabla^2 \phi_{w,h}(x,y)$ becomes:

$$\begin{aligned}\nabla^2 \phi_{w,h}(x,y) &= \frac{\partial^2 \phi_{w,y}(x,y)}{\partial x^2} + \frac{\partial^2 \phi_{w,y}(x,y)}{\partial y^2} \\ &= -4\pi^2\big(\frac{w^2}{W^2} + \frac{h^2}{H^2}\big)\phi_{w,h}(x,y).\end{aligned} \tag{46}$$

Note that Fourier bases are orthogonal bases; *i.e.*, $\langle \phi_{w_1,h_1}, \phi_{w_2,h_2} \rangle = 0$ if $(w_1,h_1) \neq (w_2,h_2)$. Hence, the matrix element $\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle$ becomes:

$$\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle = \begin{cases} 0 & (w_1,h_1) \neq (w_2,h_2), \\ -4WH\pi^2\big(\frac{w_1^2}{W^2} + \frac{h_1^2}{H^2}\big) & \text{otherwise.} \end{cases} \tag{47}$$

which makes the matrix $\mathbf{A}$ diagonal.

A.6.2 Chebyshev A

A 2D Chebyshev polynomial basis defined on $[0,W] \times [0,H]$ is defined as follows:

$$\phi_{w,h}(x,y) = \cos\Big(w \cos^{-1}\big(\frac{2x}{W} - 1\big)\Big) \cdot \cos\Big(h \cos^{-1}\big(\frac{2y}{H} - 1\big)\Big). \tag{48}$$

Let us denote $\phi_w(u)$ and $\phi_h(v)$ as follows:

$$\phi_w(u) = \cos\big(w \cos^{-1}(u)\big) \quad \forall u \in [-1,1], \tag{49}$$

$$\phi_h(v) = \cos\big(h \cos^{-1}(v)\big) \quad \forall v \in [-1,1], \tag{50}$$

then $\phi_{w,h}(x,y) = \phi_w(\frac{2x}{W} - 1) \cdot \phi_h(\frac{2y}{H} - 1)$. We obtain its derivative and second derivative with respect to x as:

$$\frac{\partial \phi_{w,h}(x,y)}{\partial x} = \frac{2}{W}\phi'_w\big(\frac{2x}{W} - 1\big) \cdot \phi_h\big(\frac{2y}{H} - 1\big), \tag{51}$$

$$\frac{\partial^2 \phi_{w,h}(x,y)}{\partial x^2} = \frac{4}{W^2}\phi''_w\big(\frac{2x}{W} - 1\big) \cdot \phi_h\big(\frac{2y}{H} - 1\big). \tag{52}$$

Therefore, $\nabla^2 \phi_{w,h}(x,y)$ becomes:

$$\nabla^2 \phi_{w,h}(x,y) = \frac{4}{W^2}\phi''_w\left(\frac{2x}{W} - 1\right) \cdot \phi_h\left(\frac{2y}{H} - 1\right) + \frac{4}{H^2}\phi_w\left(\frac{2x}{W} - 1\right) \cdot \phi''_h\left(\frac{2y}{H} - 1\right). \tag{53}$$

Reparameterize $(u, v) = \left(\frac{2x}{W} - 1, \frac{2y}{H} - 1\right)$, and let the weight function of the Chebyshev polynomial $\omega(u, v) = \frac{1}{\sqrt{1-u^2}} \cdot \frac{1}{\sqrt{1-v^2}} := \omega(u) \cdot \omega(v)$. Then, $\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle_\omega$ becomes:

$$
\begin{aligned}
&\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle_\omega \\
&= \frac{W}{2} \cdot \frac{H}{2} \iint_{[-1,1]^2} \phi_{w_1}(u)\phi_{h_1}(v) \left(\frac{4}{W^2} \phi''_{w_2}(u)\phi_{h_2}(v) + \frac{4}{H^2} \phi''_{h_2}(v)\phi_{w_2}(u) \right) \omega(u, v) \, \mathrm{d}v\mathrm{d}u \qquad (54) \\
&= \frac{H}{W} \iint_{[-1,1]^2} \phi_{w_1}(u)\phi''_{w_2}(u)\phi_{h_1}(v)\phi_{h_2}(v)\omega(u, v) \, \mathrm{d}v\mathrm{d}u \\
&\qquad\qquad + \frac{W}{H} \iint_{[-1,1]^2} \phi_{w_1}(u)\phi_{w_2}(u)\phi_{h_1}(v)\phi''_{h_2}(v)\omega(u, v) \, \mathrm{d}v\mathrm{d}u \\
&= \frac{H}{W} \int_{-1}^{1} \phi_{w_1}(u)\phi''_{w_2}(u)\omega(u) \, \mathrm{d}u \cdot \int_{-1}^{1} \phi_{h_1}(v)\phi_{h_2}(v)\omega(v) \, \mathrm{d}v \\
&\qquad\qquad + \frac{W}{H} \int_{-1}^{1} \phi_{w_1}(u)\phi_{w_2}(u)\omega(u) \, \mathrm{d}u \cdot \int_{-1}^{1} \phi_{h_1}(v)\phi''_{h_2}(v)\omega(v) \, \mathrm{d}v \qquad (55) \\
&= \frac{H}{W} \left\langle \phi_{w_1}, \phi''_{w_2} \right\rangle_\omega \cdot \langle \phi_{h_1}, \phi_{h_2} \rangle_\omega + \frac{W}{H} \langle \phi_{w_1}, \phi_{w_2} \rangle_\omega \cdot \left\langle \phi_{h_1}, \phi''_{h_2} \right\rangle_\omega . \qquad (56)
\end{aligned}
$$

From Shen [69, Eq. (2.9)], we have

$$
\phi''_k(x) = \sum_{\substack{0 \le n \le k-2 \\ n+k \text{ even}}} \frac{k(k^2 - n^2)}{c_n} \phi_n(x), \quad c_n = \begin{cases} 2 & n = 0, \\ 1 & n > 0. \end{cases} \qquad (57)
$$

Hence,

$$
\langle \phi_j, \phi''_k \rangle_\omega = \left\langle \phi_j, \sum_{\substack{0 \le n \le k-2 \\ n+k \text{ even}}} \frac{k(k^2 - n^2)}{c_n} \phi_n \right\rangle_\omega . \qquad (58)
$$

Also we know by the property of the Chebyshev polynomial that:

$$
\langle \phi_j, \phi_k \rangle_\omega = \begin{cases} \pi & j = k = 0, \\ \pi/2 & j = k \neq 0, \\ 0 & j \neq k. \end{cases} \qquad (59)
$$

Thus, we obtain

$$
\langle \phi_j, \phi''_k \rangle_\omega = \begin{cases} 0 & j > k - 2, \\ 0 & j \not\equiv k \pmod 2, \\ \dfrac{\pi}{2} k(k^2 - j^2) & j \equiv k \pmod 2,\ j \le k - 2. \end{cases} \qquad (60)
$$

By substituting Eq. (59) and Eq. (60) into Eq. (54), we obtain $\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle_\omega$, which allows us to construct the matrix $\mathbf{A}$ associated with the Chebyshev polynomials.

A.6.3 Legendre A

Analogous to the derivation of the Chebyshev $\mathbf{A}$, we can also express Legendre basis $\phi_{w,h}$ defined on $[0, W] \times [0, H]$ as $\phi_{w,h}(x, y) = \phi_w(\frac{2x}{W} - 1) \cdot \phi_h(\frac{2y}{H} - 1)$. Then, we can apply the similar logic of Eq. (54):

$$
\begin{aligned}
&\langle \phi_{w_1,h_1}, \nabla^2 \phi_{w_2,h_2} \rangle_\omega = \\
&\qquad \frac{H}{W} \Big(\langle \phi_{w_1}, \phi''_{w_2} \rangle_\omega \cdot \langle \phi_{h_1}, \phi_{h_2} \rangle_\omega \Big) + \frac{W}{H} \Big(\langle \phi_{w_1}, \phi_{w_2} \rangle_\omega \cdot \langle \phi_{h_1}, \phi''_{h_2} \rangle_\omega \Big), \qquad (61)
\end{aligned}
$$

where $\phi_w(u)$ and $\phi_h(v)$ are

$$\phi_w(u) = \frac{1}{2^w w!}\frac{\mathrm{d}^w}{\mathrm{d}u^w}(u^2-1)^w \ \forall u \in [-1,1], \tag{62}$$

$$\phi_h(v) = \frac{1}{2^h h!}\frac{\mathrm{d}^h}{\mathrm{d}v^h}(v^2-1)^h \ \forall v \in [-1,1], \tag{63}$$

respectively. Note that the weight function of the Legendre polynomial $\omega(u,v)=1$. Thus, calculating $\langle\phi_j,\phi_k\rangle$ and $\langle\phi_j,\phi_k''\rangle$ suffices to derive Legendre $\mathbf{A}$. From Shen [68, Eq. (2.6)], we have

$$\phi_k''(x) = \sum_{\substack{0\le n\le k-2\\ n+k \text{ even}}} (n+\tfrac{1}{2})\big[k(k+1)-n(n+1)\big]\,\phi_n(x). \tag{64}$$

Thus,

$$\langle\phi_j,\phi_k''\rangle = \left\langle \phi_j, \sum_{\substack{0\le n\le k-2\\ n+k \text{ even}}} (n+\tfrac{1}{2})\big[k(k+1)-n(n+1)\big]\phi_n \right\rangle. \tag{65}$$

By the orthogonality of the Legendre polynomials,

$$\langle\phi_j,\phi_k\rangle = \begin{cases} \dfrac{2}{2k+1} & j=k,\\ 0 & j\neq k. \end{cases} \tag{66}$$

Therefore,

$$\langle\phi_j,\phi_k''\rangle = \begin{cases} 0 & j > k-2,\\ 0 & j \not\equiv k \pmod 2,\\ \big[k(k+1)-j(j+1)\big] & j\equiv k \pmod 2,\ j\le k-2. \end{cases} \tag{67}$$

A.6.4 Hermite A

We adopt the physicist's Hermite polynomial defined on $\mathbb{R}$ as follows:

$$\phi_m^{\mathbb{R}}(x) = (-1)^m e^{x^2}\frac{\mathrm{d}^m}{\mathrm{d}x^w}. \tag{68}$$

We define the basis whose domain is restricted to $[0,W]$ as:

$$\phi_m(x) := \phi_m^{[0,W]}(x) = \phi_m^{\mathbb{R}}\Big(\frac{2x-W}{\alpha W}\Big), \tag{69}$$

where α is a constant that controls how tightly the Hermite basis is concentrated inside the image. We set $\alpha=\frac{1}{2}$. Thus, the 2D Hermite basis defined on $[0,W]\times[0,H]$ is:

$$\phi_{w,h}(x,y) = \phi_w^{\mathbb{R}}\Big(\frac{4x}{W}-2\Big)\cdot\phi_h^{\mathbb{R}}\Big(\frac{4y}{H}-2\Big) \tag{70}$$

Reparameterize $(u,v)=\big(\frac{4x}{W}-2,\frac{4y}{H}-2\big)$, and let the weight function of the Hermite polynomial $\omega(u,v)=\frac{e^{-(u^2+v^2)}}{\sqrt{\pi}}$. Then, by the similar logic of Eq. (54), $\langle\phi_{w_1,h_1},\nabla^2\phi_{w_2,h_2}\rangle_\omega$ becomes:

$$\langle\phi_{w_1,h_1},\nabla^2\phi_{w_2,h_2}\rangle_\omega \tag{71}$$

$$= \frac{H}{W}\Big(\langle\phi_{w_1}^{\mathbb{R}},\phi_{w_2}^{\mathbb{R}}{}''\rangle_\omega\cdot\langle\phi_{h_1}^{\mathbb{R}},\phi_{h_2}^{\mathbb{R}}\rangle_\omega\Big) + \frac{W}{H}\Big(\langle\phi_{w_1}^{\mathbb{R}},\phi_{w_2}^{\mathbb{R}}\rangle_\omega\cdot\langle\phi_{h_1}^{\mathbb{R}},\phi_{h_2}^{\mathbb{R}}{}''\rangle_\omega\Big). \tag{72}$$

Note the recurrence in Hermite polynomial:

$$\phi_m^{\mathbb{R}}{}'(x) = 2m\phi_{m-1}^{\mathbb{R}}(x), \tag{73}$$

$$\phi_m^{\mathbb{R}}{}''(x) = 4m(m-1)\phi_{m-2}^{\mathbb{R}}(x). \tag{74}$$

Also, it is known that

$$\langle\phi_j^{\mathbb{R}},\phi_k^{\mathbb{R}}\rangle_\omega = \begin{cases} \sqrt{\pi}2^k k! & j=k,\\ 0 & j\neq k. \end{cases} \tag{75}$$

Hence, $\langle \phi_j^{\mathbb{R}}, \phi_k^{\mathbb{R}\prime\prime} \rangle_\omega$ becomes

$$\langle \phi_j^{\mathbb{R}}, \phi_k^{\mathbb{R}\prime\prime} \rangle_\omega = \begin{cases} 0 & k < 2, \\ 0 & j \neq k-2, \\ 4k(k-1)\sqrt{\pi}2^{k-2}(k-2)! & j = k-2. \end{cases} \tag{76}$$

Plugging Eq. (75) and Eq. (76) to Eq. (71) completes the construction of matrix $\mathbf{A}$ associated with the Hermite polynomial.

Table 4: Implementation details of the reconstruction experiments from Sec. 4.

Tokenizer	Num params	batch size	KL weight	Warmup init. learning rate	Base learning rate	Min learning rate	$\mathbf{A}, \Delta$ learning rate	Gradient clipping norm
Flux	83.8M	32	10^{-6}	1.0×10^{-7}	1.0×10^{-5}	1.0×10^{-6}	1.0×10^{-6}	50.0
Cosmos	81.5M	32	-	1.0×10^{-7}	1.0×10^{-4}	1.0×10^{-5}	1.0×10^{-5}	50.0

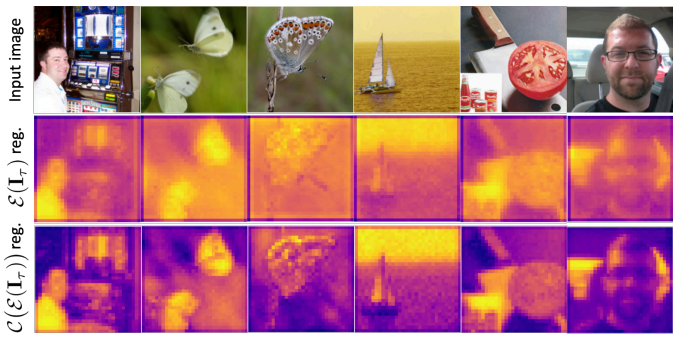
Table 5: Hyperparameters of the regularization framework

Tokenizer	τ_{max}	$\Delta_{\mathbf{I}}$	Δ init.	Max $\mathbf{A}$ init.	$d_{\mathbf{z}}(\cdot,\cdot)$	$d_{\mathbf{I}}(\cdot,\cdot)$	$\lambda_{\mathbf{z}}$	$\lambda_{\mathbf{I}}$	λ_{MAE}	λ_{LPIPS}
Flux	8.0	4.0	0.1	16.0	MSE	MAE & LPIPS	0.0	1.0	1.0	0.1
Cosmos	8.0	4.0	0.1	1.0	MSE	MAE & LPIPS	0.05	1.0	1.0	0.1

A.7 Experiment details

We provide details of the tokenizer training experiment in Table 4. The batch size for both tokenizers and gradient clipping norm are adopted from FT-SE [70], while KL weight is set to 10^{-6} following EQVAE [43]. We use a cosine annealing learning rate scheduler combined with linear warmup and a constant learning rate scheduler, which linearly raises the learning rate from initialized warmup learning rate to base learning rate, maintain the base learning rate, and then plateau to the minimum learning rate at the end. Specifically, the warmup period is set to one-tenth of the total training iterations, while the annealing period spans half of the total iterations. Following the practice in modern SSMs [27, 71], we apply smaller learning rates to SSM parameters $\mathbf{A}$ and Δ, specifically, one-tenth of the base learning rate. Borrowing the perspective discussed in Sec. 3.1, occupying smaller learning rates for these parameters indicates regularizing a network to more strictly follow the coefficient dynamics $\mathbf{c}(\mathbf{I}_{t-1}) \mapsto \mathbf{c}(\mathbf{I}_t)$, since the regularization loss $\mathcal{L}$ (Eq. (13)) would be minimized merely by optimizing the network parameters rather than dramatically updating the SSM parameters. Furthermore, we freeze the first two encoder blocks and the last two decoder blocks of tokenizers, similar to Skorokhodov *et al*. [70]. This strategy preserves the general capabilities of the pretrained weights and mitigates overfitting to the target dataset during fine-tuning. Moreover, it has been empirically shown to effectively replace the need for discriminator loss [21], thereby accelerating training [70].

Table 5 shows the hyperparameter setting we use for the regularization. The maximum level of blur τ_{max} on image is set 10.0, while the gaussian blurring difference between images $\Delta_{\mathbf{I}}$ is fixed to 4.0. To illustrate how the magnitude of τ influences the image, we show the corresponding blurring sequence in Fig. 9. When we sample an image pair of distance $\Delta_{\mathbf{I}}$ from the continuous interval $[0, \tau_{\text{max}}]$, we assume a linearly decaying probability density: $\tau_1 \sim p(x) = \frac{-2x}{(\tau_{\text{max}}-\Delta_{\mathbf{I}})^2} + \frac{2}{\tau_{\text{max}}-\Delta_{\mathbf{I}}} \; \forall x \in [0, \tau_{\text{max}} - \Delta_{\mathbf{I}}]$. Since the perceptual difference between image pairs becomes more subtle as τ_1 increases, sampling smaller τ_1 values provides stronger learning signals for the regularization. The SSM parameter Δ, which is used for the discretization introduced in Eq. (23) and Eq. (24), is initialized to 0.1. As mentioned in Fig. 7, the matrix $\mathbf{A}$ is normalized to have the maximum absolute value of 1 and is used as initialization values. To maintain the structure of $\mathbf{A}$, only non-zero values of the matrix are updated during the training. For the distance measure in latent space $d_{\mathbf{z}}$, we use mean squared error (MSE), while those in image space $d_{\mathbf{I}}$ occupies mean absolute error (MAE) combined with learned perceptual image patch similarity (LPIPS). We find that the optimal weight of the latent space measure $\Delta_{\mathbf{z}}$ varies by the type of the tokenizer. Since the loss between $\mathbf{z}_1$ and $\mathbf{z}_2$ is computed by encoding two different images, we take additional methods to prevent mode collapse, *i.e*. $\mathcal{E} : \mathbf{I} \mapsto c$ for a constant c. Specifically, to avoid $\mathbf{z}_1$ collapsing toward the representation of $\mathbf{z}_2$, we stop gradients when encoding $\mathbf{I}_{\tau_1}$ to obtain $\mathbf{z}_1$. Also, the encoder used to produce $\mathbf{z}_1$ is updated



(a) Latent visualization, where channel dimension is reduced with ℓ_2 norm. Brighter regions are higher values.

	$\mathcal{E}(\mathbf{I}_\tau)$	$\mathcal{C}(\mathcal{E}(\mathbf{I}_\tau))$
PSNR ↑	31.53	31.54
LPIPS ↓	0.0373	0.0370
SSIM ↑	0.9083	0.9101
rFID ↓	0.849	0.901
KL ↓	10.45M	203.2K
gFID ↓	17.23	13.17
sFID ↓	10.02	6.71
IS ↑	88.14	105.99
precision ↑	0.74	0.71
recall ↑	0.31	0.48

(b) Reconstruction and generation metric comparison

Figure 8: **Effect of spatial mean-centering function** $\mathcal{C}$. The one trained without $\mathcal{C}$ shows latents with higher latent norm, and exhibits tremendous degree of KL loss.

using an exponential moving average (EMA) of the main encoder weights with a decay rate of 0.999, thereby letting $\mathbf{z}_1$ not directly affected by dynamically updated weights that is optimized to minimize $d_{\mathbf{z}}(\mathbf{z}_1, \mathbf{z}_2)$.

A.8 Modification for practical implementation

We find that $\|\bar{\mathbf{A}}\| < 1$ for all choices of the basis function, which leads to the latent of the blurrier image having much smaller norm than that of the sharper image.[3] Although this indeed reflects the decaying behavior of the basis coefficients under blurring operation, in practice, this can cause latent representations of sharp images to exhibit undesirably high norm, which in turn destabilitize training. To mitigate this issue, we regularize $\mathcal{C}\big(\mathcal{E}(\mathbf{I}_\tau)\big)$ rather than $\mathcal{E}(\mathbf{I}_\tau)$, where $\mathcal{C}$ denotes a spatial mean-centering function defined as follows:

$$\mathcal{C} : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H\times W}, \qquad \mathcal{C}(\mathbf{I})_{c,h,w} = \mathbf{I}_{c,h,w} - \frac{1}{HW}\sum_{\forall(h',w')} \mathbf{I}_{c,h',w'}. \tag{77}$$

Intuitively, it drives latent values to approach to their respective spatial mean instead of zero as the input image blurs, and thus, the latent norm can avoid decaying towards zero.

Although the centering may appear to alter the target dynamics, the centered features in fact follow the same update rule with the same $\overline{\mathbf{A}}$.

Proposition A.1. *If* $\mathcal{E}(\mathbf{I}_t) = \overline{\mathbf{A}}\mathcal{E}(\mathbf{I}_{t-1})$, *then* $\mathcal{C}\big(\mathcal{E}(\mathbf{I}_t)\big) = \overline{\mathbf{A}}\mathcal{C}(\mathcal{E}(\mathbf{I}_{t-1}))$.

Proof. Let the latent $\mathcal{E}(\mathbf{I})$ be denoted as $\mathbf{z} \in \mathbb{R}^{C\times HW}$. While the update acts on the left, *i.e.*, $\mathbf{z} \mapsto \overline{\mathbf{A}}\mathbf{z}$, the spatial mean-centering $\mathcal{C}$ acts on the right:

$$\mathcal{C} : \mathbf{z} \mapsto \mathbf{z}\mathbf{P}, \qquad \text{where} \quad \mathbf{P} = \mathbf{I} - \frac{1}{HW}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{HW\times HW}. \tag{78}$$

By associativity, $\mathcal{C}(\overline{\mathbf{A}}\mathbf{z}) = \overline{\mathbf{A}}\mathbf{z}\mathbf{P} = \overline{\mathbf{A}}\mathcal{C}(\mathbf{z})$. Hence, if $\mathcal{E}(\mathbf{I}_t) = \overline{\mathbf{A}}\mathcal{E}(\mathbf{I}_{t-1})$, applying $\mathcal{C}$ to both sides yields:

$$\mathcal{C}\big(\mathcal{E}(\mathbf{I}_t)\big) = \mathcal{C}\big(\overline{\mathbf{A}}\mathcal{E}(\mathbf{I}_{t-1})\big) = \overline{\mathbf{A}}\mathcal{C}(\mathcal{E}(\mathbf{I}_{t-1})), \tag{79}$$

which implies that the centered dynamics is directly obtained if the latent follows our target dynamics. □

To justify our choice of regularizing $\mathcal{C}\big(\mathcal{E}(\mathbf{I}_\tau)\big)$ to follow the coefficient dynamics, we train the model with and without the mean-centering function $\mathcal{C}$. Without $\mathcal{C}$, the learned representations consistently show larger latent norms in Fig. 8(a). Since blurred image should always have smaller norms to satisfy Eq. (12), it results in mapping clear images to the latents with relatively larger norm magnitudes,

[3] Note that the latent of a blurry image $\mathbf{I}_N$ is represented as $\mathcal{E}(\mathbf{I}_N) = \bar{\mathbf{A}}^N\mathcal{E}(\mathbf{I}_0)$, by unrolling Eq. (12).

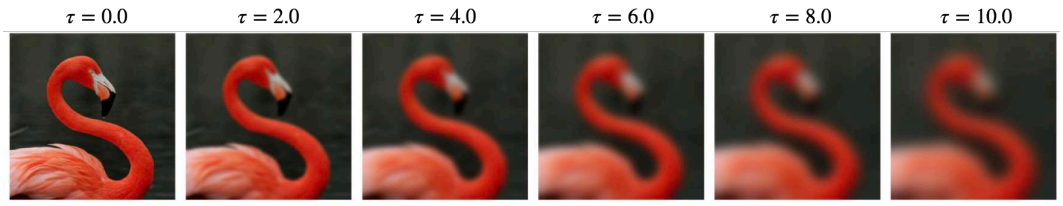


Figure 9: Image displayed with respective τ values.

inflating the value scales of the latent distribution. Apparently, such an inflation pushes the model away from a generation-friendly latent distribution; Tab. 8(b) shows that it results in a sharp decline in generation performance, as hinted from its dramatically high KL loss.

A.9 Which channels correspond to low- and high-frequency components?

In Sec. 4.2 and Fig. 5, we reveal latent channels in a low-to-high frequency order in order to validate what information each channel encodes. In this section, we describe how this ordering is determined.

Note that our regularization enforces the latent channels to mimic the dynamics of basis coefficients. Since each basis coefficient captures a different frequency component and there is a bijective mapping between a latent channel and a basis, we can sort latent channels by the frequency order. For example, let's say we designate a set of 2D Fourier basis functions: $\{\phi_n\}_{n=1}^{16} = \{\phi_{w,h}\}_{(w,h)\in\{0,1,2,3\}\times\{0,1,2,3\}}$. Remark that each basis function is defined as follows:

$$\phi_{w,h}(x,y) = \exp\Big(2\pi i\big(\frac{wx}{W} + \frac{hy}{H}\big)\Big). \tag{80}$$

Here, a basis with smaller w and h corresponds to the basis that captures low frequency components. Using this, we can group basis functions based on the frequency component they capture (low-to-high order)[4]:

$$\begin{gathered}\{\phi_{0,0}\}, \{\phi_{1,0}, \phi_{0,1}\}, \{\phi_{2,0}, \phi_{1,1}, \phi_{0,2}\}, \\ \{\phi_{3,0}, \phi_{2,1}, \phi_{1,2}, \phi_{0,3}\}, \{\phi_{3,1}, \phi_{2,2}, \phi_{1,3}\}, \{\phi_{3,2}, \phi_{2,3}\}, \{\phi_{3,3}\}.\end{gathered} \tag{81}$$

Since we map n-th channel to resemble the n-th basis ϕ_n, by the index mapping function ι (Eq. (42)), we can map each channel component to an element in Eq. (81):

$$\begin{gathered}\{\phi_1\}, \{\phi_2, \phi_5\}, \{\phi_3, \phi_6, \phi_9\}, \\ \{\phi_4, \phi_7, \phi_{10}, \phi_{13}\}, \{\phi_8, \phi_{11}, \phi_{14}\}, \{\phi_{12}, \phi_{15}\}, \{\phi_{16}\}.\end{gathered} \tag{82}$$

Hence, we complete sorting channels in low-to-high frequency order. We provide more intuitive illustrations to understand this channel-to-coefficient mapping in Fig. 10. The grouping used for experiments in Sec. 4.2 merges the first two groups and the last two groups to construct five groups:

$$\begin{gathered}g_1 = \{\phi_1, \phi_2, \phi_5\}, \quad g_2 = \{\phi_3, \phi_6, \phi_9\}, \\ g_3 = \{\phi_4, \phi_7, \phi_{10}, \phi_{13}\}, \quad g_4 = \{\phi_8, \phi_{11}, \phi_{14}\}, \quad g_5 = \{\phi_{12}, \phi_{15}, \phi_{16}\}.\end{gathered} \tag{83}$$

A.10 Ablation studies

We provide ablation studies across the type of state transition matrix $\mathbf{A}$ (Table 6(a)), the regularization strength α (Table 6(b)), the maximum blur of an image $\tau_{\max}$ (Table 6(c)), and the step size initialization value Δ (Table 6(d)). Every ablation is conducted on the Flux tokenizer, using LightningDiT-B/2 as a generator. Among all basis functions we implemented, using Fourier and Legendre bases leads to the best performance. We stick to the Fourier $\mathbf{A}$ despite its worse generation performance compared to the Legendre $\mathbf{A}$, as we consider the Fourier's case provides better trade-off between reconstruction and generation, indicated by the lower rFID value. Interestingly, random initialization of $\mathbf{A}$ leads to unstable convergence in learning, meaning that finding a properly

[4]Technically, a magnitude of frequency $\phi_{w,h}$ captures is proportional to $w^2 + h^2$, but we group them by $w + h$ for simplicity.

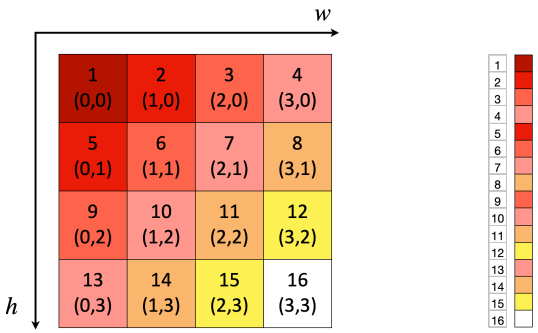


Figure 10: (Left) Illustration of the channel-to-coefficient index mapping. (Right) Channels illustrated with colors, where darker color indicates that a channel encodes lower frequency.

Table 6: **Ablation studies.** We ablate across **(a)** the type of matrix $\mathbf{A}$, **(b)** the regularization strength α, **(c)** the maximum blur $\tau_{\max}$, and **(d)** the step size Δ. The rows containing the primarily chosen settings are highlighted.

(a) Type of $\mathbf{A}$

$\mathbf{A}$ type	PSNR ↑	rFID ↓	gFID ↓	IS ↑
Fourier	31.54	0.90	13.17	105.99
Legendre	31.57	0.97	12.61	108.84
Chebyshev	31.56	0.90	13.21	105.18
Hermite	31.60	0.92	16.27	88.93
random	N/A (does not converge)			

(b) Regularization strength α

α	PSNR ↑	rFID ↓	gFID ↓	IS ↑
0	31.70	0.87	16.08	89.32
0.25	31.54	0.90	13.17	105.99
0.5	30.95	1.01	13.73	111.65
0.75	29.83	1.28	15.48	112.02

(c) Maximum blur $\tau_{\max}$

$\tau_{\max}$	PSNR ↑	rFID ↓	gFID ↓	IS ↑
6	31.50	0.94	13.20	107.77
8	31.54	0.90	13.17	105.99
10	31.54	0.93	13.02	106.88
12	31.53	0.92	13.96	101.48

(d) Step size Δ

Δ	PSNR ↑	rFID ↓	gFID ↓	IS ↑
0.1	31.60	0.91	14.06	100.54
0.5	31.50	0.93	13.04	101.34
1.0	31.54	0.90	13.17	105.99
2.0	31.50	0.96	13.00	106.86

Table 7: **Training complexity of the proposed regularizer.** Measured with 4 RTX 6000 Ada GPUs, with regularization strength $\alpha = 0.25$.

Tokenizer	speed (ms/it)	throughput (img/s)	training time (h/epoch)	memory (GB)
Flux	338.7	94.3	3.77	13.09
+ Ours	353.5 +4.4%	90.3 −4.2%	3.94 +4.4%	13.41 +2.4%
Cosmos	86.4	368.0	0.97	3.83
+ Ours	91.9 +6.4%	345.5 −6.1%	1.03 +6.4%	4.12 +7.6%

structured matrix is important to implement SSM formulation to the regularization framework. We also ablated the regularization strength α. Note that we use $\alpha = 0.25$ for all the experiments from the main manuscript (Tables 1, 8(b)). As the regularization strength α increases, we observe both decrease in reconstruction and generation capability. We hypothesize that increasing α over 0.25 overly deteriorates the tokenizer's reconstruction ability and leads to the tokenizer being insufficient to properly decode the latent back to the pixel space. For the maximum blur $\tau_{\max}$, the performance remains stable for $\tau_{\max} \in [8, 10]$, while values outside of that range degrade generation. Varying Δ over $[0.5, 2.0]$ changes performance only marginally, with a slight trade-off between reconstruction and generation.

A.11 Complexity analysis

We demonstrate the complexity of our regularizer in Table 7. Although our regularizer requires two feed-forward pass of the encoder, it barely affects the training efficiency of the tokenizer.

A.12 Application to different tasks

We explore the broader applicability of structured state-space regularization, by broadening our experiments to following tasks:

- **Text-conditioned generation** on CC12M [14], using the Flux tokenizer and a generator built with MMDiT [20] blocks.
- **Image denoising** on real-world denoising dataset SIDD [1], based on the image restoration model Restormer [87].

Table 8: Text-conditioned generation results on CC12M [14], using CLIP [63] as the text encoder

Tokenizer	COCO FID ↓	CLIP score ↑
Flux	28.57	0.290
Flux + Ours	**24.90**	**0.300**

Table 9: Image denoising results on SIDD [1], based on Restormer [87]

Model	PSNR ↑	SSIM ↑
Restormer	39.5803	0.9111
Restormer + Ours	**39.8939**	**0.9128**

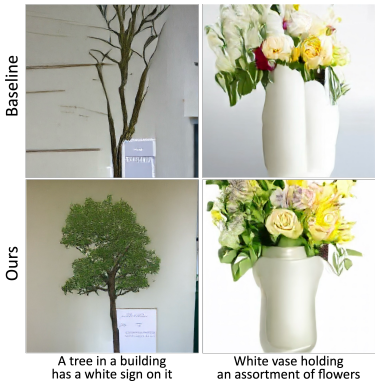


Figure 11: Qualitative examples from text-conditioned generation

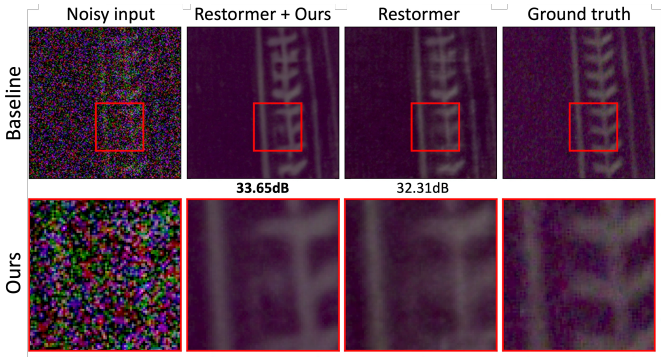


Figure 12: Qualitative examples from image denoising

For text-conditioned generation, we use Flux tokenizers fine-tuned on ImageNet with and without our regularizer, and train the corresponding models for 200K iterations, following the training recipe of MiniT2I [80]. For image denoising, we regularize the output of the refinement stage, the last Transformer block group before the final output convolution, with $\alpha = 0.1$.

We report the results in Tables 8, 9 and Figures 11, 12: applying the regularizer consistently improves performance on both tasks, highlighting the usefulness of frequency awareness in different computer vision tasks.

A.13 Additional qualitative examples

We provide additional qualitative examples of the channel revealing experiment (Fig. 13) and the generation experiment from the Cosmos tokenizer (Fig. 14). We provide more examples from the Flux tokenizer in Fig. 15 as well.

A.14 Societal impacts

Our work contributes to improving image tokenizers for generative modeling, which may positively impact applications requiring efficient and high-quality visual generation, including content creation, simulation, robotics, and scientific visualization. By improving latent-space organization without substantially increasing computational overhead, the proposed method may also help reduce training and inference costs for large-scale generative systems. However, as with other generative modeling technologies, improved image generation capabilities may also increase the risk of misuse, including the creation of deceptive or synthetic media.

A.15 Limitations

One limitation of our work is that the proposed regularization primarily targets latent-space organization for generative modeling, and therefore may not consistently improve pixel-level reconstruction fidelity compared to objectives explicitly designed for reconstruction preservation. In addition, while our method demonstrates effectiveness across multiple tokenizer architectures, we currently evaluate it mainly on image generation settings, leaving its impact on broader downstream tasks such as representation learning, video generation, or multimodal modeling as future work.



Figure 13: Additional results from the channel revealing experiment



Figure 14: Generation results from the Cosmos tokenizer

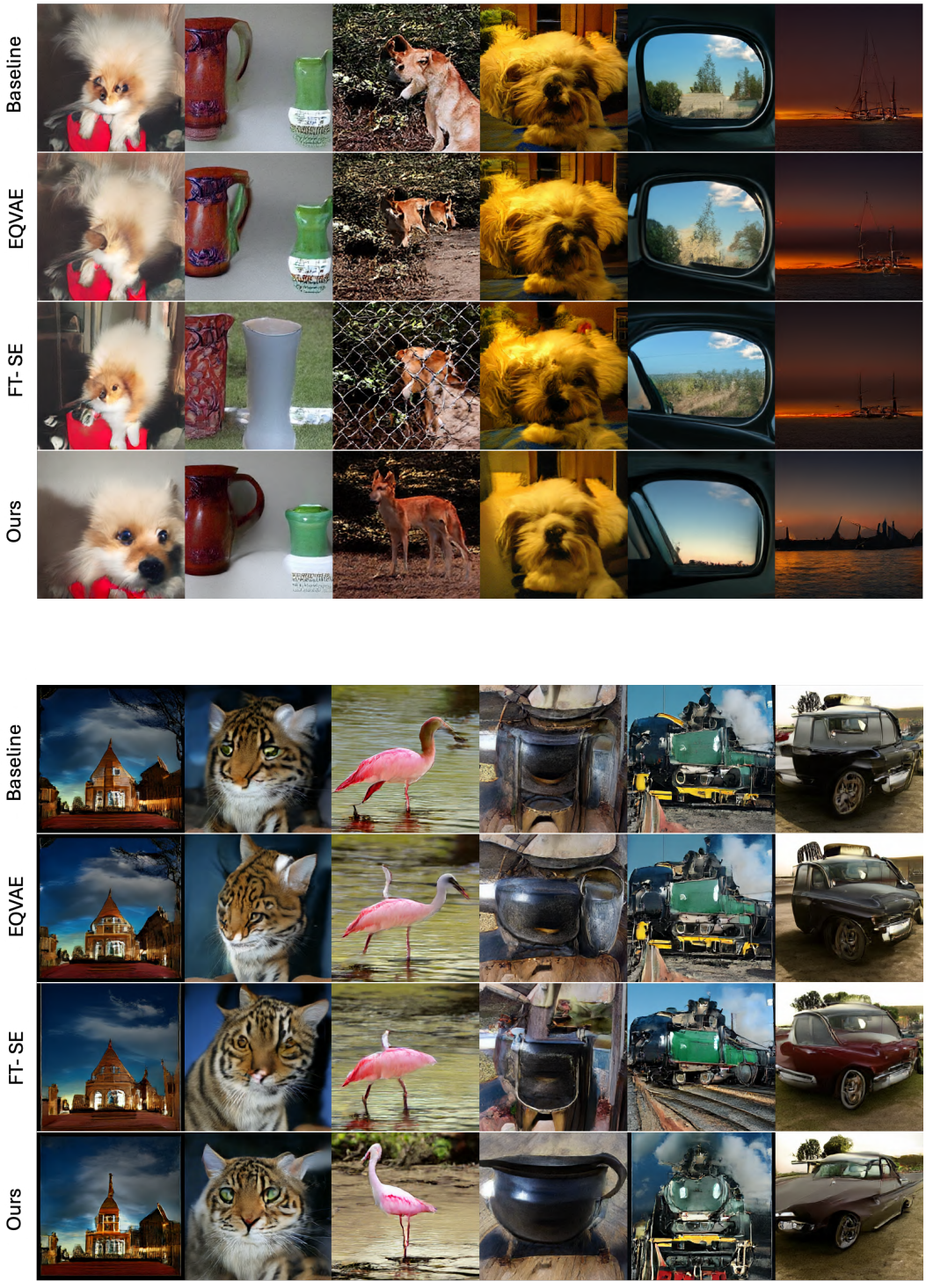


Figure 15: Additional generation results from the Flux tokenizer