Mixture-of-Hierarchical Experts: Optimized Mamba Architecture for Vision Diffusion

Yejun Jung[1] **Dongyun Kim**[2] **Jinsun Park**[1,3*]
[1] School of Computer Science and Engineering
[2] Department of Information Convergence Engineering
[3] Digital-X AIoT Research Center
Pusan National University, Busan, Republic of Korea
{yejun2, dongyoon0201, jspark}@pusan.ac.kr

Abstract

Existing Mamba-based diffusion backbones primarily rely on sequential propagation, providing limited mechanisms for explicitly modeling spatial and multi-scale representations. We address this limitation by proposing **Mixture-of-Hierarchical Experts (MoH)**, a Mamba-based diffusion backbone that introduces hierarchical Mamba scanning across multiple spatial resolutions. This hierarchy allows Mamba to capture complementary representations ranging from coarse global structure to fine-grained spatial details. However, the representations and computations required for denoising vary substantially across timesteps, making a fixed hierarchical computation suboptimal. Rather than using a fixed hierarchy, MoH organizes choices in representation scale, feature transformation, and Mamba propagation depth into a structured expert space. These choices are dynamically coordinated according to the denoising state, allowing both spatial representation and sequential processing to evolve throughout denoising. At comparable model scales, MoH substantially improves generation quality in both unconditional and text-conditional settings, reducing FID from 14.27 to 7.99 on CelebA-HQ 256×256 and from 41.80 to 20.00 on MS-COCO, while also reducing GFLOPs and inference latency. Code will be available at https://github.com/yejun2/Mixture-of-Hierarchical-Experts.

1 Introduction

Diffusion models have emerged as the dominant paradigm for image generation, achieving state-of-the-art performance across a wide range of vision tasks [20, 11, 38, 41, 45, 51]. Recent progress has been driven by advances in backbone architectures, with U-Net-based models [39, 38, 40] providing strong multi-scale inductive biases [9], and Transformer-based models offering powerful global context modeling [34, 2, 4, 31, 18, 6, 27, 32]. However, these approaches suffer from fundamental computational inefficiency [8, 10]. In particular, Transformer-based diffusion models rely on global self-attention with $\mathcal{O}(N^2)$ complexity, and in diffusion models, backbone computation is repeatedly applied across denoising steps, amplifying the overall computational cost.

To address this challenge, recent work has explored linear-time architectures such as State Space Models (SSMs) [48], notably Mamba [16], which enables sequence modeling with $\mathcal{O}(N)$ complexity and offers a promising direction for scalable diffusion backbones. However, directly applying Mamba to vision tasks remains challenging, as visual data is inherently two-dimensional, while Mamba operates through one-dimensional sequential propagation [52, 29, 22, 17, 35, 50]. This mismatch limits its ability to effectively capture spatial topology and multi-scale features.

*Corresponding author.

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



Figure 1: Samples generated by MoH-B on CelebA-HQ at 256 × 256 resolution.

Existing approaches, such as ZigMa [21], attempt to alleviate this issue through multi-directional or stochastic scanning strategies that improve spatial coverage. However, these methods remain fundamentally sequential and primarily modify the traversal order, rather than explicitly modeling how spatial information should be structured and aggregated. As a result, spatial features are still processed in a largely uniform manner, without a mechanism to adaptively integrate information across resolutions or spatial regions.

At the same time, diffusion models exhibit inherently timestep-dependent behavior. At high-noise timesteps, effective denoising relies on capturing coarse global structures, whereas at low-noise timesteps, the model must refine fine-grained details [7, 1]. Despite this, existing Mamba-based approaches apply identical computations across timesteps, lacking an explicit mechanism to adapt representation, computation depth, and feature mixing to the evolving requirements of the denoising process. Consequently, they often rely on uniformly deep architectures to handle all cases, leading to inefficient computation and suboptimal feature utilization.

In this work, we propose **Mixture-of-Hierarchical Experts (MoH)** to address these limitations. Unlike existing Mamba backbones that process visual tokens through sequential scanning at a fixed resolution, MoH organizes Mamba scanning hierarchically across multiple spatial resolutions, providing an explicit multi-scale spatial inductive bias. Rather than using this hierarchy as a fixed architecture, MoH formulates the architectural choices arising from hierarchical scanning as a structured expert space that can adapt to the current denoising state. Specifically, MoH adapts the active representation scales, the degree of spatial mixing before compression, and Mamba propagation depth. The latter provides a complementary control over representation, as deeper Mamba layers tend to form higher-level representations while shallower layers retain more localized information [23]. An integrated routing controller coordinates these architectural decisions to construct a computation path that evolves throughout denoising.

Contributions.

- We introduce a **hierarchical Mamba architecture** that performs Mamba scanning across multiple spatial resolutions, providing explicit multi-scale spatial structure for visual generation beyond flat sequential processing.
- Rather than using hierarchy as a fixed architecture, we formulate the **architectural choices arising from the hierarchy**—including representation scale, feature transformation, and Mamba propagation depth—as a structured expert space that can be adaptively selected throughout denoising.
- We introduce **coordinated routing** conditioned on denoising states, where SSR, CMR, and EDR jointly adapt active representation scales, spatial mixing and compression, and Mamba propagation depth to construct an appropriate computation path for each denoising state.
- MoH substantially improves generation quality on both CelebA-HQ and MS-COCO while reducing computational cost and inference latency. As shown in Fig. 1, MoH also generates visually coherent samples with fine-grained details.

2 Related Work

Diffusion Backbones. Early diffusion models [20] adopted U-Net [39] as the standard denoising backbone, and it has remained the dominant choice in much of the subsequent literature. U-Net follows an encoder–decoder architecture that progressively compresses the input into hierarchical representations and reconstructs the output through a coarse-to-fine decoding process, with skip connections. This design endows the network with a strong multi-scale inductive bias, while the skip connections directly propagate high-resolution information from the encoder to the decoder. Such properties align naturally with the requirements of diffusion models, where the denoising network must repeatedly recover both global structure and local detail from noisy inputs at every timestep, making U-Net a particularly effective backbone for this task.

More recently, Transformer-based architectures [34, 2] have emerged as a competitive alternative and are now widely adopted in large-scale diffusion models [13]. Unlike U-Net backbones, these models discard the convolutional encoder–decoder hierarchy. Instead, the noisy latent is partitioned into non-overlapping patches, projected into tokens, and processed by a stack of homogeneous Transformer blocks. Timestep and conditioning information are injected via adaptive layer normalization. This design enables global receptive fields from the first layer through self-attention and removes the architectural asymmetry between encoder and decoder. It also exhibits favorable scaling behavior, as performance improves predictably with model size, data, and compute. These properties are particularly beneficial for diffusion models in high-dimensional latent spaces, where capturing long-range dependencies and scaling to large model sizes are critical. This explains the strong empirical performance of Transformer backbones in modern text-to-image systems.

More recently, State Space Models (SSMs), particularly Mamba [16], have been adapted to 2D vision tasks [52, 29] and have been explored as efficient alternatives to attention-based backbones for diffusion [47, 49, 21]. Mamba treats the input as a one-dimensional sequence and propagates information through a selective state-space update, where transition and projection parameters are conditioned on the input. The full sequence is processed in linear time. This design provides linear complexity in sequence length and introduces input-adaptive global memory, making it appealing for high-resolution latents whose token counts grow rapidly with image size. However, existing Mamba-based diffusion models still fall short of Transformer-based counterparts in generation quality. The underlying 1D causal scan imposes an arbitrary ordering on inherently 2D latents, leading to a range of hand-designed scan strategies, including bidirectional [52], cross-directional [29], and zigzag [21] scans. While these approaches improve spatial coverage, they introduce biases toward specific traversal patterns, with no superior strategy. The compressed recurrent state limits dense long-range interactions that self-attention captures more directly. As a result, designing competitive and scan-agnostic SSM-based diffusion backbones remains an open problem.

Adaptive Computation and Routing. A complementary line of research investigates adaptive computation [43, 15, 37, 14]. This direction is motivated by the observation that not all inputs require the same amount of compute, and that conditionally activated subnetworks can improve the capacity–efficiency trade-off over uniformly dense models. These methods introduce learned routing mechanisms that selectively activate a subset of parameters per input. For example, mixture-of-experts layers dispatch each token to a small number of expert sub-networks via a gating function, while dynamic-depth and early-exit models adjust the number of executed layers based on input difficulty. Such conditional execution allows total model capacity to scale with the number of experts or layers without a proportional increase in per-input compute. It also enables different inputs to specialize in different parameter subsets, leading to improved efficiency and performance across language and vision tasks. However, existing methods primarily perform routing at the token or layer level within a fixed architecture. As a result, higher-level design choices—such as the allocation of computation across feature scales, network depth, and feature compression—remain uniform across inputs. In contrast, our approach performs routing at the architectural level, enabling timestep-conditioned adaptation along these dimensions.

3 Preliminaries

Diffusion Models. We consider a standard forward noising process at timestep t:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

and train a network f_θ to predict a target (e.g., velocity v) [25]:

$$\mathcal{L} = \mathbb{E}_{t,x_0,\epsilon}\left[\|f_\theta(x_t, t, c_t) - v\|^2\right], \tag{2}$$

Here, c_t denotes the conditioning signal, including timestep information and optional inputs such as text embeddings. During inference, samples are generated by iteratively applying the learned denoising function.

Mamba. Mamba [16] is a selective state space model that updates hidden states sequentially [17]:

$$h_i = A_i h_{i-1} + B_i x_i, \quad y_i = C_i h_i, \tag{3}$$

where (A_i, B_i, C_i) are input-dependent parameters. In practice, Mamba blocks are used with residual connections, preserving input information while enabling progressive feature transformation. Stacking multiple blocks increases the effective receptive field, allowing deeper configurations to capture stronger global dependencies, whereas shallower configurations preserve more local structure.

4 Mixture-of-Hierarchical Experts

4.1 Overview

We propose *Mixture-of-Hierarchical Experts (MoH)*, a Mamba-based diffusion backbone that organizes Mamba computation into a spatial hierarchy. As illustrated in Fig. 2, the encoder performs Mamba scanning across multiple resolutions, producing hierarchical representations with progressively different spatial scales.

MoH interprets the architectural choices created by this hierarchy as a structured expert space. Given the current denoising state, SSR selects the active hierarchical stages, while CMR and EDR adapt feature transformation and Mamba propagation depth within the selected stages. These routing decisions are coordinated by the integrated routing controller (IRC), enabling complementary adaptation across routing spaces for the current denoising state.

The lowest active stage serves as a dynamic bottleneck. From this bottleneck, the decoder progressively upsamples and fuses the routed encoder representations to reconstruct the output in a coarse-to-fine manner. The following sections describe the hierarchical Mamba architecture, routing mechanism, routing modules, and prediction head in detail.

4.2 Hierarchical Mamba Architecture

MoH employs an S-stage hierarchical encoder that progressively forwards features across multiple spatial resolutions. Given the input feature z_s at stage s, EDR first determines the stage-wise Mamba computation and produces the processed representation h_s. CMR then transforms and spatially compresses h_s to construct the input z_{s+1} for the next stage:

$$z_{s+1} = \mathrm{CMR}_s\left(h_s, c_t, \alpha_{t,s}^{\mathrm{CMR}}\right), \quad h_s = \mathrm{EDR}_s\left(z_s, c_t, \alpha_{t,s}^{\mathrm{EDR}}\right), \qquad s = 1, \ldots, S-1. \tag{4}$$

Here, c_t denotes the conditioning signal at timestep t, incorporating the timestep embedding and optional conditions such as text embeddings, while $\alpha_{t,s}^{\mathrm{EDR}}$ and $\alpha_{t,s}^{\mathrm{CMR}}$ denote the corresponding routing decisions. Importantly, the stages do not operate as independent multi-scale branches. Instead, they form a sequential hierarchical forwarding path: each stage processes the representation received from the previous scale, and its routed output is subsequently transformed and forwarded to the next, lower-resolution stage. Consequently, the representation constructed at each stage also determines the information propagated to deeper stages.

As features are forwarded through the hierarchy, their spatial resolution progressively decreases. By scanning at multiple resolutions, the hierarchy provides explicit multi-scale spatial structure and allows each stage to use relatively shallow Mamba stacks, since lower-resolution stages can capture broader spatial context without relying solely on additional Mamba depth. This progressive forwarding enables the model to capture complementary representations from global semantics to fine details.

For text-conditioned generation, textual information is incorporated throughout this hierarchical forwarding process. In the encoder, text embeddings are injected through cross-attention during

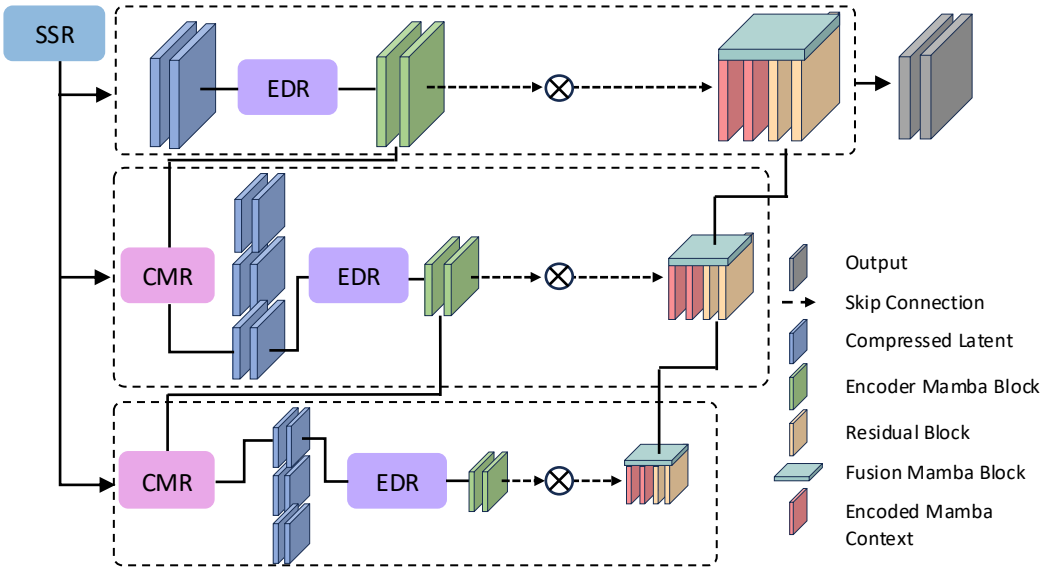

Figure 2: Overview of Mixture-of-Hierarchical Experts (MoH). MoH treats a hierarchical backbone as a structured expert space and dynamically constructs a timestep-dependent computation path via routing over architectural decisions.

stage-wise feature processing, allowing the intermediate representations at different resolutions to be conditioned on the input text. In the decoder, stage-wise projections of the text embedding are used during coarse-to-fine reconstruction, so that textual conditioning is preserved as hierarchical features are progressively integrated across resolutions.

4.3 Routing Mechanism: Integrated Routing Controller (IRC)

As shown in Fig. 3, we employ a unified routing framework that adapts hierarchical computation to the current denoising state. Given timestep t and conditioning signal y, we construct a shared conditioning representation c_t and extract stage-wise statistics ϕ_t from the input feature F_t. The IRC additionally predicts a global scale prior p_t, which provides shared routing context across the three routing modules. Without IRC, each router is optimized independently without this shared routing prior, whereas IRC couples their decisions through p_t to encourage coordinated optimization across routing spaces. Based on these signals, SSR, CMR, and EDR make specialized decisions for stage selection, compression and mixing, and Mamba depth, respectively.

4.4 Routing Modules

Stage Selection. SSR selects a subset of hierarchical stages via adaptive sparse routing. Given routing logits b_t^{SSR}, we compute activation probabilities and select stages using a threshold τ:

$$\rho_{t,r} = \mathrm{sigmoid}(b_{t,r}^{\mathrm{SSR}}), \quad r \in \mathcal{S}, \qquad m_{t,r}^{(0)} = \mathbf{1}[\rho_{t,r} > \tau], \quad k_t^{(0)} = \sum_r m_{t,r}^{(0)}. \tag{5}$$

where $\mathcal{S}$ denotes the set of candidate stage resolutions. To prevent degenerate sparsity, we enforce a minimum number $k_{\min}$ of active stages:

$$m_{t,r} = \begin{cases} m_{t,r}^{(0)}, & k_t^{(0)} \geq k_{\min}, \\ \mathbf{1}[r \in \text{Top-K}_{k_{\min}}(b_t^{\mathrm{SSR}})], & k_t^{(0)} < k_{\min}. \end{cases} \tag{6}$$

We normalize the hard selection and activation probabilities and apply a straight-through estimator [3] to obtain the final stage-selection weights:

$$\tilde{m}_{t,r} = \frac{m_{t,r}}{\sum_j m_{t,j}}, \qquad \tilde{\rho}_{t,r} = \frac{\rho_{t,r}}{\sum_j \rho_{t,j}}, \qquad \alpha_{t,r}^{\mathrm{SSR}} = \tilde{m}_{t,r} + \tilde{\rho}_{t,r} - \mathrm{sg}(\tilde{\rho}_{t,r}). \tag{7}$$

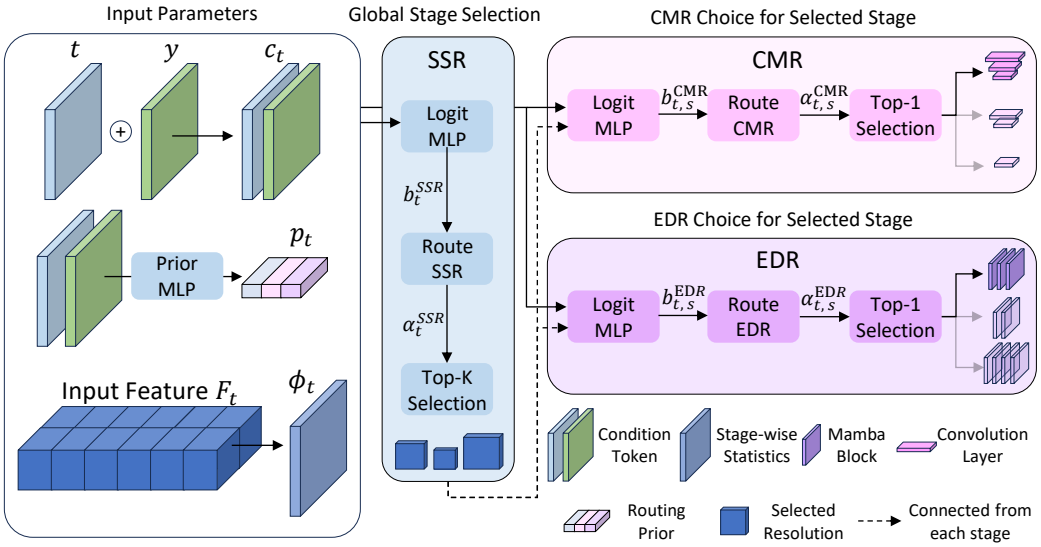

Figure 3: Integrated routing mechanism of MoH. A shared conditioning signal and routing prior coordinate SSR, CMR, and EDR to adapt stage selection, feature mixing, and Mamba propagation depth.

The selected stage set $\mathcal{S}_t = \{r \in \mathcal{S} \mid \alpha_{t,r}^{\mathrm{SSR}} > 0\}$ defines the active representation levels, and the deepest stage determines the bottleneck resolution. This dynamic bottleneck enables adaptive control over the level of abstraction, using deeper stages for global structure modeling and shallower stages for fine-grained denoising.

Feature Mixing and Compression. CMR selects a transformation branch via Top-1 routing:

$$\rho_{t,s}^{\mathrm{CMR}} = \mathrm{softmax}\big(b_{t,s}^{\mathrm{CMR}}\big)\,, \qquad \alpha_{t,s}^{\mathrm{CMR}} = \text{Top-1}\big(\rho_{t,s}^{\mathrm{CMR}}\big)\,. \tag{8}$$

where $b_{t,s}^{\mathrm{CMR}}$ denotes the routing logits. CMR is trained using the same straight-through formulation as Eq. 7. We define three transformation presets, *weak*, *base*, and *strong*, corresponding to convolutional depths of 0, 1, and 2, respectively, before a shared stride-2 downsampling operation. These presets control the amount of local spatial processing applied before Mamba operates on the compressed representation. Consequently, CMR provides a CNN-like spatial inductive bias whose strength can be adapted to the current denoising state, rather than relying solely on a fixed sequential scan for spatial interaction. CMR is applied when constructing the 16×16 and 8×8 hierarchical representations.

Mamba Scanning Depth. EDR selects the number of Mamba blocks via Top-1 routing:

$$\rho_{t,s}^{\mathrm{EDR}} = \mathrm{softmax}\big(b_{t,s}^{\mathrm{EDR}}\big)\,, \qquad \alpha_{t,s}^{\mathrm{EDR}} = \text{Top-1}\big(\rho_{t,s}^{\mathrm{EDR}}\big)\,. \tag{9}$$

where $b_{t,s}^{\mathrm{EDR}}$ denotes the routing logits. EDR is trained using the same straight-through formulation as Eq. 7. We define *shallow*, *base*, and *deep* configurations using shared-prefix Mamba blocks. Deeper routes extend the same shared prefix with additional Mamba blocks according to the selected preset; detailed stage-wise configurations are provided in Table 8.

Each additional Mamba block applies another input-dependent selective state transition, so the selected depth directly controls how information is repeatedly propagated and aggregated along the scan path. Previous observations in Vision Mamba [23] further suggest that deeper layers tend to form higher-level representations, whereas shallower layers retain more localized information. EDR therefore treats Mamba propagation depth as a denoising-state-dependent control axis, rather than merely varying generic network capacity.

Discussion. This asymmetric routing design reflects the distinct functional roles of each module, enabling complementary adaptation across representation, feature transformation, and computational depth. Together, these coordinated routing mechanisms allow the model to jointly adapt representation and computation across timesteps, resulting in a more efficient and expressive denoising process.

4.5 Prediction Head

The decoder reconstructs the output by progressively integrating hierarchical representations in a coarse-to-fine manner. Starting from the bottleneck representation, it iteratively upsamples and refines features at each stage. At each stage, the upsampled decoder feature is fused with the corresponding routed encoder feature via gated summation:

$$\hat{u}_{s-1} = \mathrm{Up}(u_s) + G_{s-1} \odot \tilde{z}_{s-1}, \qquad u_{s-1} = M_{s-1}(\hat{u}_{s-1}, c_t), \tag{10}$$

where stage s denotes a lower-resolution level and $s-1$ a higher-resolution level. Here, u_s denotes the decoder feature at stage s, and $\hat{u}_{s-1}$ is the fused feature obtained by combining the upsampled decoder feature with the routed encoder feature at stage $s-1$. $\tilde{z}_{s-1}$ is the routed encoder feature, G_{s-1} is a learnable gating function, and M_{s-1} is a Mamba block for feature refinement.

The fusion combines low-resolution semantic context from the decoder with higher-resolution gated encoder features, while the Mamba block refines the fused representation by integrating global context and local details. Importantly, the encoder features $\tilde{z}_{s-1}$ are not raw activations but outputs of hierarchical processing via EDR, which already incorporate Mamba-based transformations. Thus, the decoder operates on multi-level representations encoding both semantic structure and fine-grained details. Under conditioning, both encoder and decoder features are modulated by the shared signal c_t, and the refinement step further integrates this information. This enables the reconstruction process to adapt both feature integration and refinement based on the input condition.

5 Experiments

5.1 Datasets and Training Details

Datasets. We conduct experiments on CelebA-HQ 256×256 [24] and MS-COCO 256×256 [28]. For CelebA-HQ, we use precomputed latent representations, where images are encoded into a $32 \times 32 \times 4$ latent space using the Stable Diffusion v1.5 VAE [26, 38]. For MS-COCO, we use the 2014 split with text conditioning. Each sample consists of a latent image representation ($32 \times 32 \times 4$) and a CLIP text embedding [36] of dimension 768.

Experiment Settings. We adopt velocity prediction [25] with timesteps sampled from $t \sim \mathcal{U}[0, 1]$. Models are trained with AdamW [30] using mixed precision and EMA (decay $= 0.9999$). We use the adaptive Dormand–Prince (DOPRI5) ODE solver [44, 5] with 50 uniformly spaced output time points for validation during training and 250 for final Fréchet Inception Distance (FID) [19] evaluation. All ablations are conducted consistently with AMP-FP16, while the final models use AMP-BF16 for improved numerical stability. We explicitly report these settings since prior Mamba baselines specify only mixed-precision training. Batch sizes are 64 for CelebA-HQ and 256 for MS-COCO. Additional details are provided in Appendix A.1.

Evaluation. We report Fréchet Inception Distance (FID) as the primary evaluation metric [19]. We additionally report IS [42], sFID [33], and FDD [46] for comprehensive evaluation. All FID results are computed using the EMA model.

5.2 Results on Image Generation

CelebA-HQ 256×256**.** As shown in Table 1, all models other than ZigMa are trained for 100K steps under identical settings, while the ZigMa result is taken from the original paper. MoH-B achieves an FID of 7.99, substantially improving over the 14.27 FID of ZigMa [21], while using 129M parameters and 80.88G FLOPs, compared with 133M and 97.52G for ZigMa. MoH also consistently outperforms both SSM-based GSPN [48] and Transformer-based SiT [32] across the compared model scales. These results demonstrate that hierarchical scanning Mamba architecture with timestep-conditioned routing substantially improves generation quality without relying on increased model scale or computation.

Table 1: Generation quality and model complexity of diffusion backbones on CelebA-HQ 256×256. Except for ZigMa, all models are trained for 100K steps under identical training settings. The ZigMa FID is taken from the original paper.

Method	Backbone	FID^{5k} ↓	FLOPs (G) ↓	Params (M) ↓
GSPN-B/2 [48]	SSM	21.00	36.17	141
GSPN-L/2	SSM	20.71	118.14	461
SiT-B/2 [32]	Transformer	17.30	**23.02**	129
SiT-L/2	Transformer	15.65	80.74	456
ZigMa [21]	Mamba	14.27	97.52	133
MoH-S (ours)	Mamba	9.16	35.79	**57**
MoH-B (ours)	Mamba	**7.99**	80.88	129

Table 2: Comparison of Mamba-based diffusion backbones on text-conditional image generation on MS-COCO 256×256. MoH-B follows the training protocol of ZigMa [21].

Method	Venue	FID^{5k} ↓
Vision Mamba [52]	ICML'24	60.20
ZigMa [21]	ECCV'24	41.80
U-Shape Mamba [12]	CVPRW'25	39.10
MoH-B (ours)	–	**20.00**

Table 3: Inference efficiency comparison across models. **Thr.** denotes denoiser evaluations per second, **Lat.** denotes mean latency per denoiser evaluation (milliseconds), and **Mem.** denotes peak GPU memory usage (MiB).

Model	Thr. ↑	Lat. ↓	Mem. ↓
ZigMa [21]	51.170	19.543	573.32
MoH-B	**66.244**	**15.096**	**545.12**

MS-COCO 256×256**.** We evaluate MoH on the more complex MS-COCO dataset. As shown in Table 2, MoH-B achieves an FID of 20.00, substantially outperforming prior Mamba-based models, with 60.20 for Vision Mamba [52], 41.80 for ZigMa [21], and 39.10 for U-Shape Mamba [12]. Unlike MoH, U-Shape Mamba [12] employs a U-shaped encoder–decoder hierarchy over Mamba features, rather than treating multiple spatial resolutions as Mamba scanning spaces. The baseline results are adopted from the comparisons reported in the corresponding prior work. For a controlled comparison, MoH-B follows the comparable model scale as ZigMa; a more specific comparison of each model is provided in Appendix A.2. These results demonstrate the effectiveness of timestep-conditioned hierarchical routing for complex, multi-object generation, which requires capturing both global semantics and fine-grained details.

Efficiency. We further evaluate the inference efficiency of MoH-B against ZigMa, where ZigMa refers to the ZigMa configuration reported in Table 1. All measurements are conducted with the CelebA-HQ generation checkpoints at batch size 1. As shown in Table 3, MoH-B improves throughput from 51.170 to 66.244 denoiser evaluations per second while reducing the mean inference latency from 19.543 ms to 15.096 ms per denoiser evaluation. This corresponds to a 29.5% increase in throughput and a 22.8% reduction in inference latency. MoH-B also reduces peak GPU memory usage from 573.32 MiB to 545.12 MiB. Together with the improvements in generation quality and model complexity reported in Table 1, these results demonstrate that MoH achieves more efficient inference without relying on increased model scale or computation.

5.3 Ablation Study

Hierarchical Architecture. We evaluate the effect of hierarchical modeling and stage configuration. As shown in Table 5, introducing a two-stage hierarchy $(32, 16)$ without routing reduces FID from 14.27 (ZigMa) to 13.17. Adding the 8×8 stage further reduces FID to 11.93, demonstrating the benefit of hierarchical modeling even without adaptive routing. However, extending the hierarchy to $(32, 16, 8, 4)$ worsens FID to 12.64, indicating that increasing hierarchy depth is not uniformly beneficial. We therefore adopt the three-stage configuration $(32, 16, 8)$, which achieves the lowest FID among the evaluated configurations, and exclude the 4×4 stage from the routing candidate set.

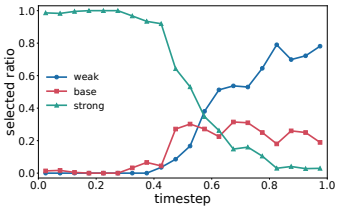

Figure 4: CMR behavior at stage 2 (16×16 resolution).

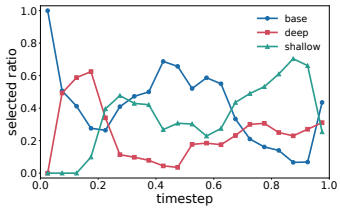

Figure 5: EDR behavior at stage 2 (16×16 resolution).

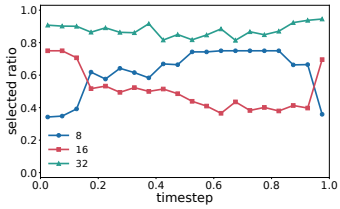

Figure 6: SSR behavior across timesteps.

Table 4: Scan strategy ablation on CelebA-HQ 256×256, varying only the scan in a fixed $(32, 16, 8)$ hierarchy.

Model	Scan type	FID^{5k} ↓
Hierarchy-only	zigzagN8 [21]	12.37
Hierarchy-only	v2 [52]	**11.93**

Table 5: Ablation of hierarchical stage configurations on CelebA-HQ 256×256.

Model	Stages	FID^{5k} ↓
ZigMa [21]	–	14.27
Hierarchy-only	$(32, 16)$	13.17
	$(\mathbf{32}, \mathbf{16}, \mathbf{8})$	**11.93**
	$(32, 16, 8, 4)$	12.64

Scan Strategy. We evaluate the sensitivity to scanning patterns in the hierarchical architecture. As shown in Table 4, different scanning strategies yield similar performance (12.37 vs. 11.93), indicating that the proposed formulation substantially reduces scan-order bias. This contrasts with prior scanning strategy-based approaches, where performance is highly sensitive to traversal design [52, 29, 22, 21].

Routing Components. As shown in Table 6, CMR and EDR independently improve the fixed hierarchical baseline, reducing FID from 11.93 to 10.60 and 10.81, respectively. However, simply combining the two routers without IRC does not preserve these gains, yielding a worse FID of 11.00. This indicates that increasing routing capacity alone is insufficient, since spatial mixing and Mamba propagation depth jointly determine the resulting computation path and therefore require coordinated decisions. Introducing IRC substantially improves the joint configuration from 11.00 to 9.51. Further enabling adaptive-k SSR reduces FID to 9.28, showing an additional benefit from dynamically selecting the active hierarchical stages. Note that the 9.28 FID in Table 6 uses the same MoH-B architecture reported in Table 1; the ablation result is obtained with AMP-FP16, while the final 7.99 result uses AMP-BF16.

Coordination of Routing Components. We define coordination as complementary routing between CMR and EDR, rather than independent optimization of their individual decisions. This dependency naturally follows the hierarchical computation order: CMR first determines how features are spatially mixed and compressed, and EDR subsequently determines the Mamba propagation depth for the resulting representation. Therefore, an effective EDR decision should account for the spatial configuration produced by CMR.

This distinction is reflected in the learned routing behavior shown in Table 7. Without IRC, EDR exhibits routing patterns similar to the independently trained EDR-only policy and remains largely insensitive to the preceding CMR choice. With IRC, in contrast, the EDR distribution changes according to the selected CMR configuration, indicating that propagation depth is adapted complementarily to the spatial processing already performed by CMR. This behavioral difference is consistent with the ablation in Table 6: jointly enabling CMR and EDR without coordination does not preserve their individual gains, whereas introducing IRC improves FID from 11.00 to 9.51.

Routing Behaviors. The learned policies also exhibit a structured evolution over the denoising trajectory. As shown in Figs. 4–6, CMR progressively shifts toward weaker spatial mixing, EDR adapts its propagation depth, and SSR utilizes multiple hierarchical representations without collapsing to a single scale. Together, these behaviors are consistent with the coarse-to-fine nature of diffusion, with

Table 6: Ablation study of routing components on CelebA-HQ 256×256. All routing ablations use the $(32, 16, 8)$ hierarchy, which achieves the best FID^{5k} among the evaluated stage configurations. Each module provides complementary gains, and the integrated controller achieves the best FID^{5k} $\downarrow$ performance.

Model	SSR	CMR	EDR	IRC	FID^{5k} $\downarrow$	sFID^{5k} $\downarrow$	FDD^{5k} $\downarrow$	IS^{5k} $\uparrow$
MoH-B	top-3	–	–	–	11.93	10.56	3.45	3.04
MoH-B	top-3	✓	–	–	10.60	9.16	2.70	2.95
MoH-B	top-3	–	✓	–	10.81	8.93	2.93	2.80
MoH-B	top-3	✓	✓	–	11.00	9.22	3.14	2.86
MoH-B	top-3	✓	✓	✓	9.51	8.39	2.46	2.91
MoH-B (full)	adaptive-k	✓	✓	✓	**9.28**	8.50	2.74	2.95

Table 7: Conditional EDR routing under CMR decisions during early denoising (timestep $t \in [0, 0.33]$).

Model	CMR	EDR decision (%)		
		shallow	base	deep
w/o IRC	weak	0.0	0.0	**100.0**
	base	0.0	0.0	**100.0**
	strong	0.0	0.0	**100.0**
with IRC	weak	24.8	**74.5**	0.7
	base	12.4	33.3	**54.3**
	strong	38.7	14.0	**47.3**

the architectural computation path evolving as denoising progresses. Additional routing behaviors across timesteps and hierarchical stages are analyzed in Appendix A.4.

6 Conclusion

We present **Mixture-of-Hierarchical Experts (MoH)**, a Mamba-based diffusion backbone that improves both generation quality and efficiency through timestep-conditioned hierarchical routing. By treating hierarchical representations as a structured expert space, MoH dynamically selects feature levels, transformation strength, and Mamba depth to match the evolving requirements of the denoising process. MoH achieves consistent improvements over both Transformer- and Mamba-based baselines [34, 21], with particularly large gains on complex datasets such as MS-COCO [28]. These results demonstrate that coordinated adaptation across representation and computation is crucial for modeling complex generative processes. More broadly, our findings suggest that the limitations of Mamba in vision stem from the lack of adaptive mechanisms rather than the architecture itself. By introducing architectural-level routing, MoH provides a flexible and scalable framework, highlighting *adaptive architectural routing* as a promising direction for generative modeling.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00358935, 50%) and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ITRC (Information Technology Research Center) grant funded by the Korea government (MSIT) (IITP-2026-RS-2023-00260098, 50%).

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22669–22679, 2023.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[4] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[5] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

[6] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Diffusion transformers for image and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6441–6451, 2024.

[7] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

[8] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[9] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3965–3977. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/20568692db622456cc42a2e853ca21f8-Paper.pdf`.

[10] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[12] Alex Ergasti, Filippo Botti, Tomaso Fontanini, Claudio Ferrari, Massimo Bertozzi, and Andrea Prati. U-shape mamba: State space model for faster diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3242–3249. IEEE, 2025.

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

[14] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022.

[15] Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

[16] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[17] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[18] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pages 393–411. Springer, 2024.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[21] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

[22] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan, 2024. URL `https://arxiv.org/abs/2403.09338`.

[23] Zizheng Huang, Haoxing Chen, Jiaqi Li, Jun Lan, Huijia Zhu, Weiqiang Wang, and Limin Wang. Stochastic layer-wise shuffle for improving vision mamba training. In *International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=6vP20U55bA`.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hk99zCeAb`.

[25] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[27] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *Advances in Neural Information Processing Systems*, 35:26295–26308, 2022.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[29] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in Neural Information Processing Systems*, 37:103031–103063, 2024.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

[32] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[33] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.

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

[35] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6443–6451, 2025.

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

[37] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[40] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.

[41] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29, 2016.

[43] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[45] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[46] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023.

[47] Yao Teng, Yue Wu, Han Shi, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Dim: Diffusion mamba for efficient high-resolution image synthesis. *arXiv preprint arXiv:2405.14224*, 2024.

[48] Hongjun Wang, Wonmin Byeon, Jiarui Xu, Jinwei Gu, Ka Chun Cheung, Xiaolong Wang, Kai Han, Jan Kautz, and Sifei Liu. Parallel sequence modeling via generalized spatial propagation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4473–4483. IEEE, 2025.

[49] Jing Nathan Yan, Jiatao Gu, and Alexander M Rush. Diffusion models without attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8239–8249, 2024.

[50] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*, 2024.

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

[52] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning*, pages 62429–62442. PMLR, 2024.

A Technical appendices and supplementary material

A.1 Implementation Details

We provide specific training details of MoH in Table 8.

Detailed Training Setup. We employ mixed-precision training with both AMP-FP16 and AMP-BF16. All ablation studies are conducted using AMP-FP16 for fair comparison, while final models are trained with AMP-BF16 for improved stability and performance. We verify that relative improvements are consistent across both settings. We use an exponential moving average (EMA) of parameters with decay 0.9999, and report results using the EMA model. For CelebA-HQ, we use a batch size of 16 per GPU (4 A6000 GPUs, total batch size 64). For MS-COCO, we use a batch size of 64 per GPU (4 A100 GPUs, total batch size 256).

Table 8: Implementation details of MoH-B.

Category	Configuration
Input	Latent size 32×32, channels 4, image resolution 256
Patch	patch size 1
Learning Rate	0.0001
Hierarchy	Encoder stages $\{32, 16, 8\}$
Stage Depth	High-res 4, hierarchy 3, bottleneck 4
Routing (SSR)	Adaptive-k stage routing over $\{32, 16, 8\}$; threshold 0.5, min stages 2, max stages 3; batch balancing enabled
Dynamic Bottleneck	Enabled; candidates $\{32, 16, 8\}$; decoder starts from the lowest selected resolution
Routing (CMR)	Top-1 compression routing at output stages $\{16, 8\}$; choices are premix depths $\{weak : 0, base : 1, strong : 2\}$ before stride-2 downsampling
Routing (EDR)	Top-1 encoder-depth routing at stages $\{32, 16, 8\}$; choices are shallow/base/deep depths: $32 : \{3, 4, 5\}$, $16 : \{2, 3, 4\}$, $8 : \{3, 4, 5\}$
Training Batch	Batch 64 (16×4) for CelebA-HQ 256×256 and Batch 256 (64×4) for MS-COCO 256×256
Mamba	Scan type v2; EDR uses shared-prefix depth routing

A.2 Model size specification

Table 9 summarizes the model scales used for comparison. On CelebA-HQ, MoH-S and MoH-B contain 57M and 129M parameters, respectively, while ZigMa contains 133M parameters. For text-conditioned MS-COCO generation, both ZigMa and MoH additionally employ cross-attention modules for text conditioning, resulting in larger parameter counts even under the same backbone scale. Accordingly, MoH-S and MoH-B contain 162M and 237M parameters on MS-COCO, compared with 214M for ZigMa.

Table 9: Model size comparison on CelebA-HQ and MS-COCO 256×256.

Dataset	Model	Params (M)	FID^{5k} $\downarrow$
CelebA-HQ 256×256	ZigMa	133	14.27
	MoH-S	57	9.16
	MoH-B	129	7.99
MS-COCO 256×256	ZigMa	214	41.80
	MoH-S	162	22.32
	MoH-B	237	20.00

A.3 Sampling and Evaluation Details

We use the adaptive Dormand–Prince (Dopri5) solver for sampling with $\mathrm{atol} = 10^{-6}$ and $\mathrm{rtol} = 10^{-3}$. The initial latent is sampled from a standard Gaussian distribution with shape $4 \times 32 \times 32$,

and evaluation is performed using EMA model parameters. For MS-COCO, classifier-free guidance is not used ($\mathrm{CFG} = 1.0$).

During online FID validation, we request 50 output time points from the ODE solver, while 250 output time points are used for offline sampling. Since Dopri5 uses adaptive integration, these values specify the sampled output trajectory rather than a fixed number of solver steps.

A.4 Routing Coordination Analysis and Routing Behavior

Table 10: Timestep-wise CMR routing distributions (%).

Timestep	CMR-only			CMR+EDR w/o IRC			CMR+EDR with IRC		
	weak	base	strong	weak	base	strong	weak	base	strong
0.05	50.0	0.0	50.0	50.0	50.0	0.0	0.0	51.6	48.4
0.15	0.0	50.0	50.0	61.0	38.6	0.4	5.3	39.8	54.9
0.25	0.0	50.0	50.0	39.7	50.5	9.8	53.0	7.0	39.9
0.35	41.9	50.0	8.1	1.3	48.7	50.0	89.2	1.4	9.5
0.45	50.0	47.0	3.0	0.0	50.0	50.0	96.1	0.2	3.7
0.55	58.4	0.0	41.6	0.0	50.0	50.0	98.5	0.0	1.5
0.65	77.1	0.0	22.9	47.9	2.1	50.0	99.3	0.0	0.7
0.75	96.3	0.0	3.7	49.8	0.2	50.0	99.4	0.0	0.6
0.85	100.0	0.0	0.0	50.0	0.0	50.0	100.0	0.0	0.0
0.95	100.0	0.0	0.0	50.0	0.0	50.0	100.0	0.0	0.0

Table 11: Timestep-wise EDR routing distributions (%).

Timestep	EDR-only			CMR+EDR w/o IRC			CMR+EDR with IRC		
	shallow	base	deep	shallow	base	deep	shallow	base	deep
0.05	0.0	33.3	66.7	0.0	0.0	100.0	33.3	1.4	65.4
0.15	0.0	33.3	66.7	0.0	0.0	100.0	34.8	51.1	14.1
0.25	33.3	66.7	0.0	0.0	0.0	100.0	36.3	62.0	1.7
0.35	33.3	66.7	0.0	0.0	66.7	33.3	34.0	65.7	0.3
0.45	66.7	33.3	0.0	0.0	66.7	33.3	27.9	71.9	0.2
0.55	100.0	0.0	0.0	0.0	100.0	0.0	22.9	77.0	0.1
0.65	100.0	0.0	0.0	33.3	33.3	33.3	11.3	88.4	0.3
0.75	100.0	0.0	0.0	33.3	33.3	33.3	0.6	95.2	4.2
0.85	100.0	0.0	0.0	100.0	0.0	0.0	0.0	99.2	0.8
0.95	100.0	0.0	0.0	100.0	0.0	0.0	0.7	99.2	0.2

Table 12: Conditional EDR routing distributions under CMR decisions, comparing models with and without IRC.

Timestep range	Model	CMR decision	EDR decision (shallow / base / deep)
Early (0.00–0.33)	w/o IRC	weak ($n = 1545$, 50.3%)	0.0% / 0.0% / 100.0%
		base ($n = 1424$, 46.4%)	0.0% / 0.0% / 100.0%
		strong ($n = 103$, 3.4%)	0.0% / 0.0% / 100.0%
	with IRC	weak ($n = 597$, 19.4%)	24.8% / 74.5% / 0.7%
		base ($n = 1008$, 32.8%)	12.4% / 33.3% / 54.3%
		strong ($n = 1467$, 47.8%)	38.7% / 14.0% / 47.3%
Mid (0.34–0.66)	w/o IRC	weak ($n = 504$, 12.3%)	0.0% / 0.0% / 100.0%
		base ($n = 1544$, 37.7%)	0.0% / 33.2% / 66.8%
		strong ($n = 2048$, 50.0%)	0.0% / 100.0% / 0.0%
	with IRC	weak ($n = 3923$, 95.8%)	33.2% / 66.5% / 0.3%
		base ($n = 16$, 0.4%)	12.5% / 87.5% / 0.0%
		strong ($n = 157$, 3.8%)	99.4% / 0.6% / 0.0%
Late (0.67–1.00)	w/o IRC	weak ($n = 1534$, 49.9%)	66.8% / 0.0% / 33.2%
		base ($n = 2$, 0.1%)	0.0% / 0.0% / 100.0%
		strong ($n = 1536$, 50.0%)	66.7% / 33.3% / 0.0%
	with IRC	weak ($n = 3066$, 99.8%)	0.6% / 97.0% / 2.4%
		base ($n = 0$, 0.0%)	– / – / –
		strong ($n = 6$, 0.2%)	16.7% / 16.7% / 66.7%

Analysis of Router Coordination. Tables 10–12 further illustrate how IRC coordinates CMR and EDR throughout the denoising trajectory. CMR and EDR control distinct but interdependent aspects of the same hierarchical computation path. CMR determines the spatial configuration presented to the subsequent Mamba blocks by controlling feature mixing and compression, whereas EDR determines how deeply information is propagated under that spatial configuration. Consequently, the appropriate Mamba depth depends not only on the denoising timestep, but also on the spatial representation produced by CMR.

Without IRC, jointly training CMR and EDR does not naturally establish this dependency. The coarse-to-fine CMR policy becomes less structured, while EDR frequently exhibits saturated depth decisions that are insensitive to the selected CMR route. For example, during early denoising, EDR selects the deep route in 100% of cases regardless of whether CMR selects weak, base, or strong mixing. With IRC, the conditional EDR distribution instead changes according to the spatial configuration selected by CMR. During the same early interval, weak CMR is predominantly followed by base EDR (74.5%), whereas base CMR is more frequently paired with deep EDR (54.3%). This conditional relationship also evolves across the denoising trajectory: during the mid interval, strong CMR is paired with shallow EDR in 99.4% of cases, while during the late interval, weak CMR is paired with base EDR in 97.0% of cases.

The marginal routing distributions provide a complementary view of this behavior. With IRC, CMR progressively shifts toward weaker spatial processing as denoising proceeds, selecting the weak route almost exclusively at later timesteps, while EDR simultaneously adapts its propagation depth to the resulting spatial context. Thus, IRC does not simply encourage CMR and EDR to make similar routing decisions; rather, it coordinates their distinct roles so that Mamba propagation depth is selected in relation to both the spatial configuration and the current denoising state. This behavior is consistent with the coarse-to-fine motivation of MoH: stronger spatial transformation and context propagation are available when constructing representations from highly noisy inputs, while later stages favor lighter spatial processing as fine-grained information emerges. Consistent with this coordinated behavior, IRC improves FID from 11.00 to 9.51 under the same fixed top-3 SSR setting.

Analysis of Routing Behavior across Timesteps and Stages. Figures 7–9 further visualize how the routing decisions evolve across denoising timesteps and hierarchical stages. As shown in Fig. 7,

CMR exhibits a clear transition in spatial processing: stronger mixing is preferred during earlier timesteps, whereas the weak route becomes dominant as denoising progresses. This suggests that the model allocates more explicit spatial transformation when constructing representations from noisy inputs and progressively reduces such processing as finer structures emerge.

EDR exhibits a complementary, stage-dependent behavior. At the highest-resolution stage (Fig. 9), deep Mamba processing dominates the early timesteps and is gradually replaced by lighter propagation as denoising proceeds. At the lower-resolution stage (Fig. 8), the routing distribution is more diverse across shallow, base, and deep configurations, indicating that the required propagation depth also depends on the spatial scale at which Mamba operates. Thus, EDR does not follow a single depth schedule shared across the hierarchy, but adapts Mamba propagation jointly to the denoising state and representation scale.

Finally, Fig. 6 shows that SSR also changes the active representation levels throughout denoising. The 32×32 stage remains a primary pathway, while the relative use of the 16×16 and 8×8 stages varies substantially with timestep, demonstrating that the model dynamically adjusts its multi-scale representation rather than relying on a fixed hierarchy. Taken together, Figs. 7–9 show that spatial mixing, Mamba propagation depth, and active representation scale all evolve along the denoising trajectory. These learned routing patterns are consistent with the coarse-to-fine nature of diffusion: stronger aggregation and propagation are emphasized when forming structure from noisy representations, while the computation shifts toward finer and lighter processing as denoising progresses.

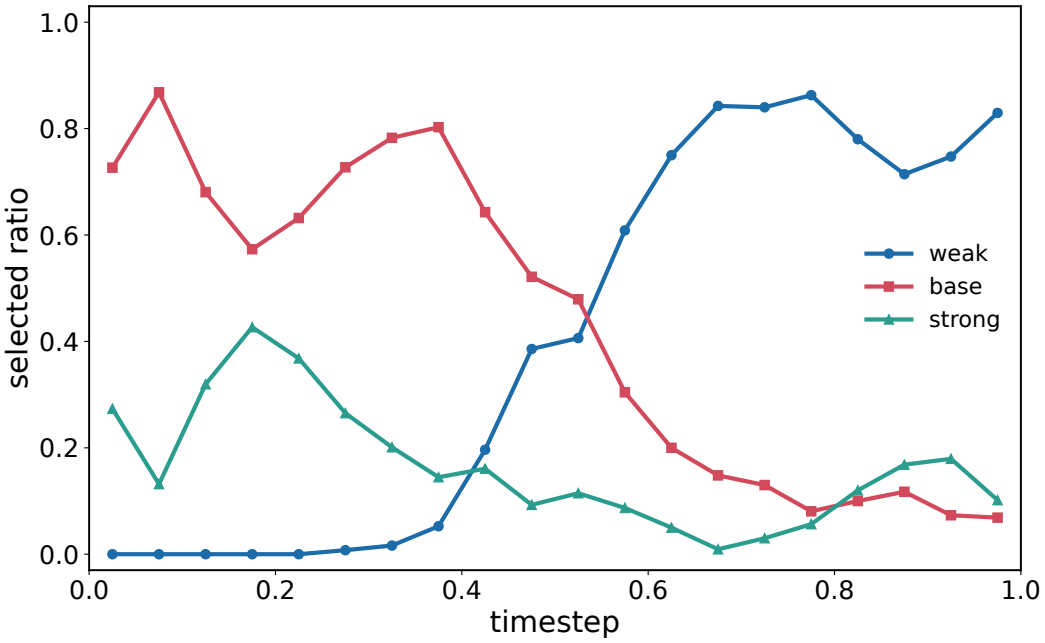

Figure 7: CMR behavior at third stage (8×8 resolution).

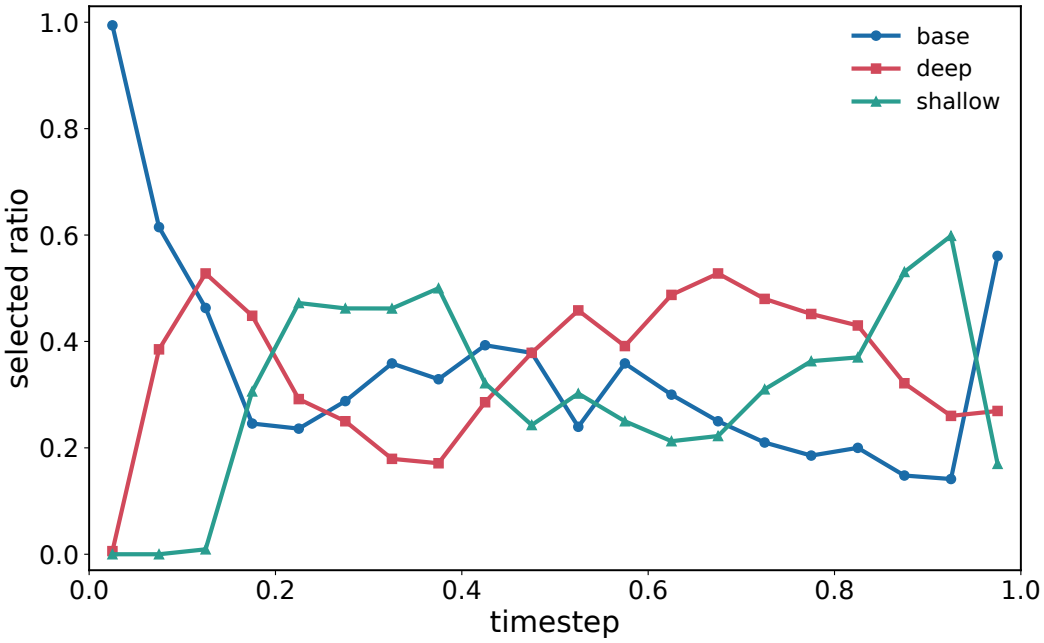

Figure 8: EDR behavior at third stage (8×8 resolution).

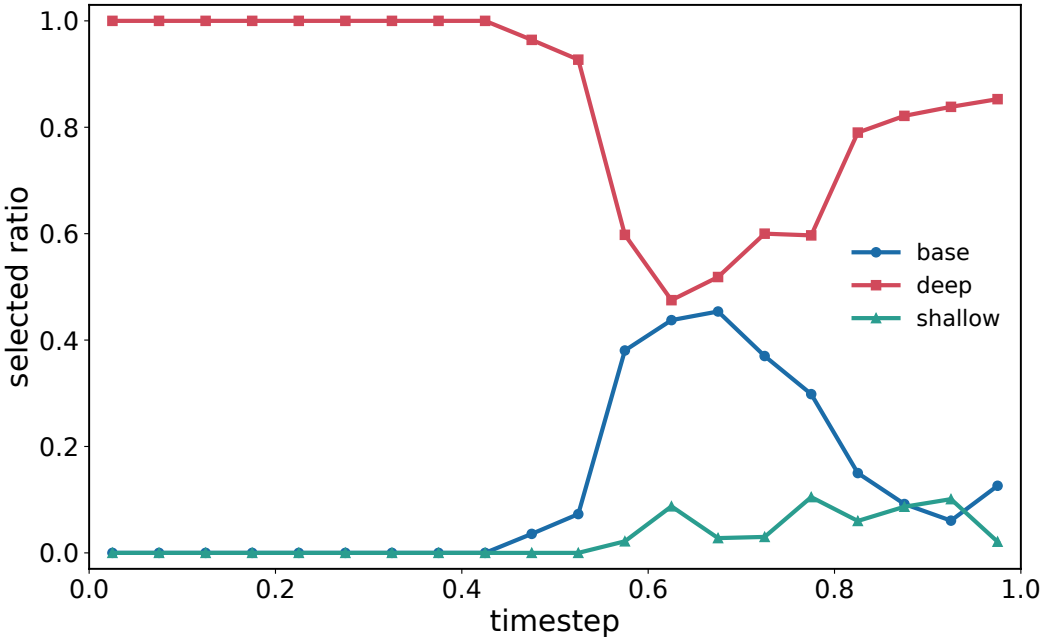

Figure 9: EDR behavior at first stage (32×32 resolution).

A.5 Training Stability

As shown in Fig. 10 and Fig. 11, both training loss and FID improve smoothly throughout training, with no evidence of optimization collapse or late-stage instability. MoH reaches an FID of 7.99 at 100K steps. These results indicate that dynamic routing does not hinder convergence.

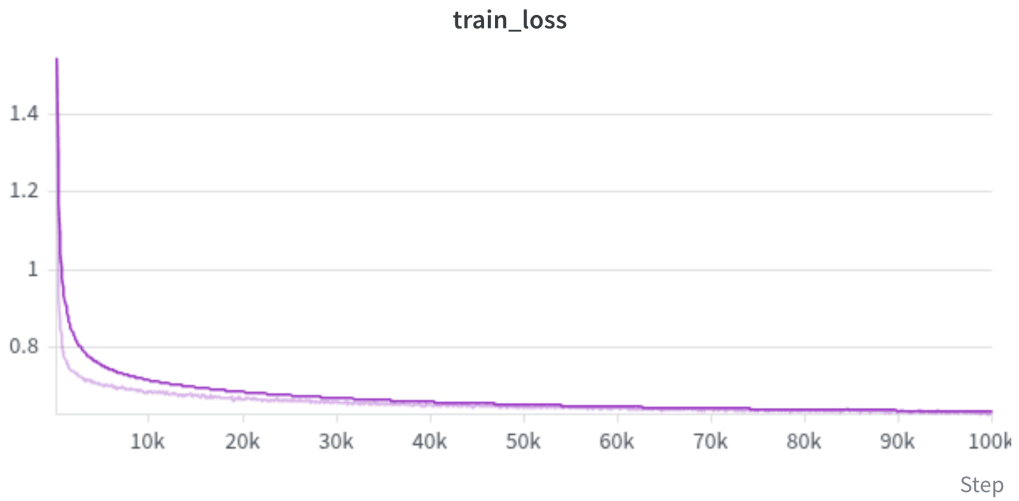

Figure 10: Training loss curves of MoH-B on CelebA-HQ 256×256.

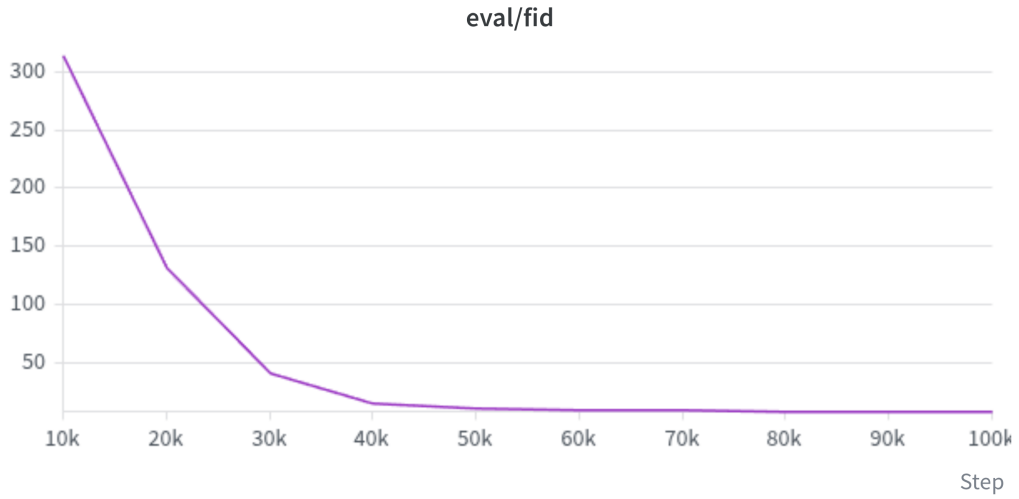

Figure 11: FID^{5k} progression of MoH-B on CelebA-HQ 256×256.

Router Utilization. We further examine whether the learned routers collapse to a small subset of routing choices. As shown in Table 13, all routing options remain actively utilized. SSR selects all three stages with non-trivial frequencies, while CMR and EDR also distribute their decisions across all available compression and depth choices at both hierarchical stages. These results indicate that MoH learns structured routing policies without collapsing to a fixed computation path.

Table 13: Average router utilization of MoH-B on CelebA-HQ 256×256.

Router / Stage	Selection ratio		
SSR	32: 0.776	16: 0.633	8: 0.591
CMR (16×16)	weak: 0.201	base: 0.216	strong: 0.583
CMR (8×8)	weak: 0.261	base: 0.286	strong: 0.453
EDR (32×32)	shallow: 0.027	base: 0.125	deep: 0.848
EDR (16×16)	shallow: 0.195	base: 0.362	deep: 0.443
EDR (8×8)	shallow: 0.324	base: 0.381	deep: 0.295

A.6 Qualitative results

Fig. 12 and Fig. 13 show generated samples from **MoH-B** on CelebA-HQ 256×256 and MS-COCO 256×256. On CelebA-HQ, MoH-B generates visually coherent facial structures with fine-grained details. On MS-COCO, the model produces diverse scenes and object compositions, demonstrating its applicability to more complex image distributions.



Figure 12: Visualization of generated samples on CelebA-HQ 256×256.

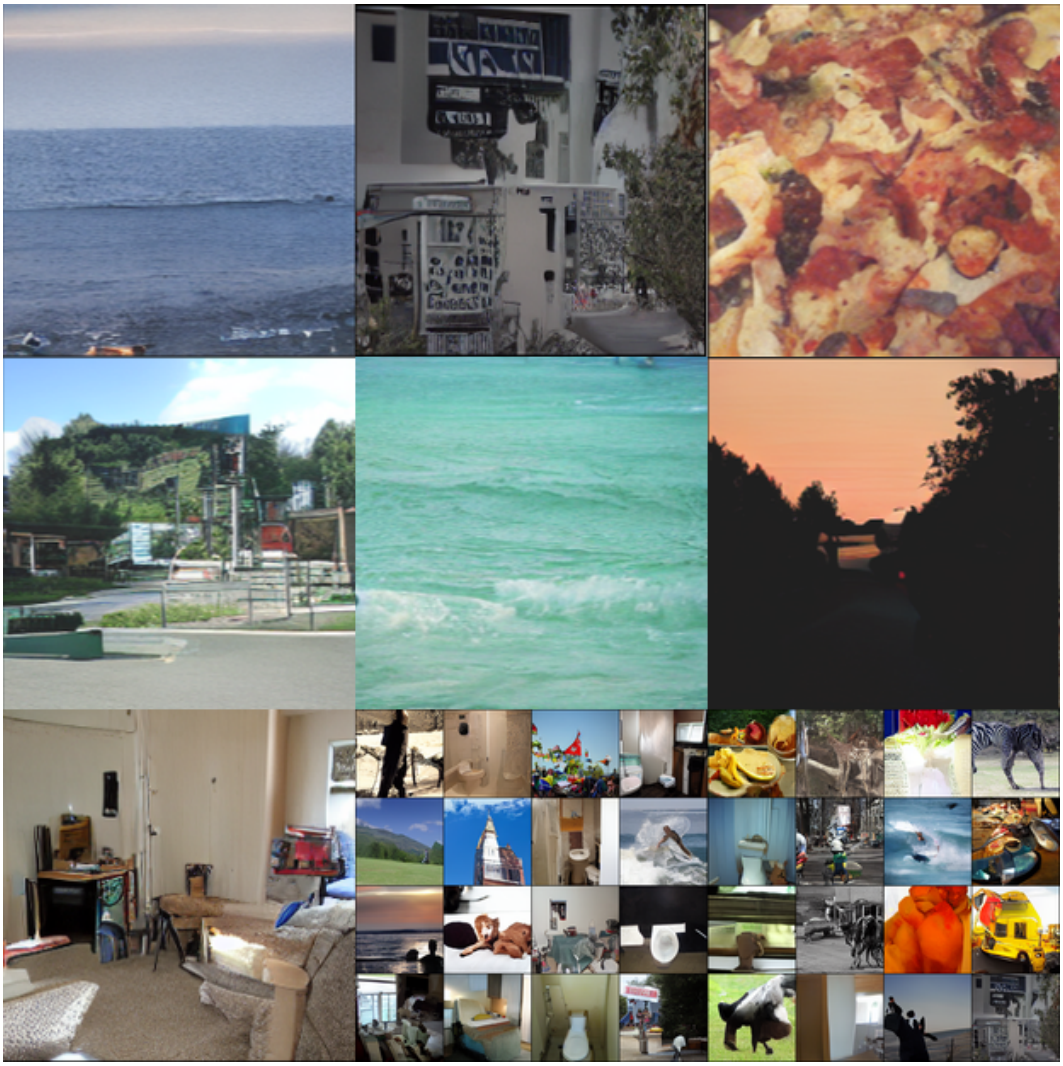

Figure 13: Visualization of generated samples on MS-COCO 256×256.

B Limitations

Due to limited computational resources, our experiments are primarily conducted at 256×256 resolution and do not cover substantially larger model scales or higher-resolution generation. We plan to extend the evaluation to larger-scale and higher-resolution generation in future work.

C License note

CelebA-HQ is used for non-commercial research purposes under the applicable dataset usage terms. MS-COCO annotations are distributed under CC BY 4.0, while the images retain their respective Flickr licenses. We use both datasets for research purposes and follow the applicable usage terms.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our main claims are supported by the experimental comparisons in Tables 1, 2, and 3, and the ablation studies in Tables 5 and 6. Additional model-scale comparisons are provided in Appendix A.2.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work shown in Appendix B.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: Our work is empirical in nature and does not present any formal theorems.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper provides the main architectural components, routing formulation, datasets, training objective, optimizer, learning rate, mixed-precision settings, EMA usage, batch sizes, GPU types, sampling settings, and evaluation metrics in Sec 4 and 5. These details provide a reasonable basis for reproducing the main experimental results, although releasing code would further improve reproducibility.

Guidelines:

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

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

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

   Answer: [No]

   Justification: Code, pretrained checkpoints, and reproduction instructions will be publicly released following the camera-ready submission.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5.1 and Appendix A.1 specify the datasets, latent representations, text conditioning setup, velocity prediction objective, timestep sampling, optimizer, learning rate, EMA, mixed precision, GPU types, batch sizes, sampling procedure, and evaluation metrics. These details are sufficient to understand the reported experimental results.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The paper reports quantitative metrics such as FID, sFID, FDD, and IS, but does not report error bars, confidence intervals, statistical significance tests, or results over multiple random seeds due to limited computational resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resources are explained in A.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses standard image generation benchmarks and does not involve human-subject experiments or private user data collection. To the best of our knowledge, the work follows the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

Justification: The paper does not currently include a broader impact discussion. Since the method improves image generation quality and efficiency, potential impacts include more efficient generative modeling, while potential negative impacts include misuse for synthetic image generation or disinformation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This is foundational research on diffusion model architectures, conducted on standard academic benchmarks (e.g., CelebA-HQ, MS-COCO). Our work does not involve web-scraped or identity-conditioned models, and operates within the same capability envelope as existing openly distributed baselines on these benchmarks. Additional safeguards are therefore not applicable.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Licenses for existing assets are described in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Our paper does not introduce new datasets. The released code and model checkpoints will be accompanied by documentation upon release.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.