Focusing on What Matters: Saliency-Harnessing Accurate Routing for Diffusion MoE

Haoyou Deng[1,2], Keyu Yan[2], Chaojie Mao[2], Xiang Wang[1], Yu Liu[2], Changxin Gao[1], and Nong Sang[1†]

[1] Key Laboratory of Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China

[2] Tongyi Lab, Alibaba Group, Hangzhou, China

{haoyoudeng, wxiang, cgao, nsang}@hust.edu.cn

yankeyu66@aliyun.com, {chaojie.mcj, ly103369}@alibaba-inc.com

Abstract. Mixture-of-Experts (MoE) architectures have emerged as a powerful paradigm for scaling diffusion models in visual generation. Recent advancements have focused on adaptively allocating computational resources across diverse tokens to improve efficiency and performance. However, we identify a routing assignment problem in existing diffusion MoE frameworks: the router fails to accurately allocate more computational resources to salient tokens. Our analysis attributes this failure to the router's reliance on noise-corrupted latent features throughout the denoising process. Such stochastic noise obscures the critical structural and textural information, thereby preventing the router from effectively distinguishing salient tokens. To address this, we propose **SharpMoE**, a post-training framework with a saliency-harnessing accurate routing mechanism, which utilizes clean latent features as a noise-free guidance signal for routing. By bypassing the noise-distorted inputs, SharpMoE provides the router with clear saliency guidance, enabling the identification of salient tokens even in high-noise stages. Furthermore, we introduce a trajectory routing loss to constrain the compute allocation throughout the multi-step denoising trajectory, ensuring precise resource allocation along the generation rollout. Extensive experiments demonstrate that SharpMoE serves as a versatile, plug-and-play solution that further enhances the pretrained, converged MoE models, achieving state-of-the-art performance in visual generation.

Keywords: Image Generation · Diffusion Model · Mixture-of-Experts

1 Introduction

Diffusion models [12] have achieved remarkable advancements in visual generation [21, 22, 24, 34]. Recent research has increasingly focused on scaling these models to billions of parameters, with the goal of enhancing image fidelity and

[†]Corresponding Author

generation quality. Diffusion transformers (DiT) [22] have emerged as a promising framework, marking a notable architectural shift from U-Net backbones to transformer-based designs and demonstrating exceptional scalability potential. Despite these advancements, however, further scaling DiT models to even larger parameters is hindered by the inherent inefficiency associated with dense parameter activation [31].

To push the boundaries of model scale and capability, the Mixture-of-Experts (MoE) paradigm [14,28] has emerged as a widely used framework within the large language models (LLMs) community [4, 16, 18, 33, 35, 42]. MoE employs a router that dynamically assigns a sparse subset of parameters (experts) to each input token and aggregates their outputs to produce the final result. This manner enables a significant expansion of model capacity while maintaining computational efficiency. However, while MoE has achieved substantial success in LLMs, directly applying these established strategies to diffusion models has often resulted in suboptimal performance [37]. This performance gap arises from a fundamental difference in modality: text tokens are discrete and exhibit high semantic density, while visual tokens are spatially correlated and inherently redundant.

To this end, recent research has focused on developing MoE architectures specifically for diffusion-based visual generation. Early efforts in diffusion MoE [9] often relied on the token-choice routing strategy, where each image token is routed to a fixed number of top-ranked experts. However, more recent studies [29, 31, 40] have identified critical differences in computational requirements across image regions. In diffusion generation, regions containing salient tokens, those rich in critical detail, demand greater computational focus (*e.g.*, more experts) than background or redundant areas. This insight highlights the necessity for a dynamic and saliency-aware routing mechanism capable of adaptively assigning computational resources based on varying saliency levels and textural complexities within an image.

Although effective, existing dynamic routing methods, such as DiffMoE [29], suffer from a routing assignment problem: the router fails to accurately allocate more computational resources to salient tokens. To investigate this issue, we utilize the Laplacian operator to extract the textural information of each token as a saliency representation, and then analyze the distribution of experts assigned to tokens with varying saliency levels. The findings, depicted in Fig. 1(a), demonstrate that although existing methods aim to achieve saliency-aware allocation, their actual routing results are largely saliency-insensitive, showing minimal variation in expert allocation across tokens with different saliency levels. We attribute this limitation to the *noisy routing*, where the router is consistently conditioned on noise-corrupted latents throughout the multi-step denoising process. This pervasive noise, especially pronounced at early high-noise timesteps, masks critical structural and textural details. Ultimately, this corruption impairs the router's ability to effectively distinguish salient regions, which leads to inaccurate allocation of computational resources.

To address the aforementioned issue, we introduce **SharpMoE**, a post-training framework that incorporates a saliency-harnessing accurate routing mech-

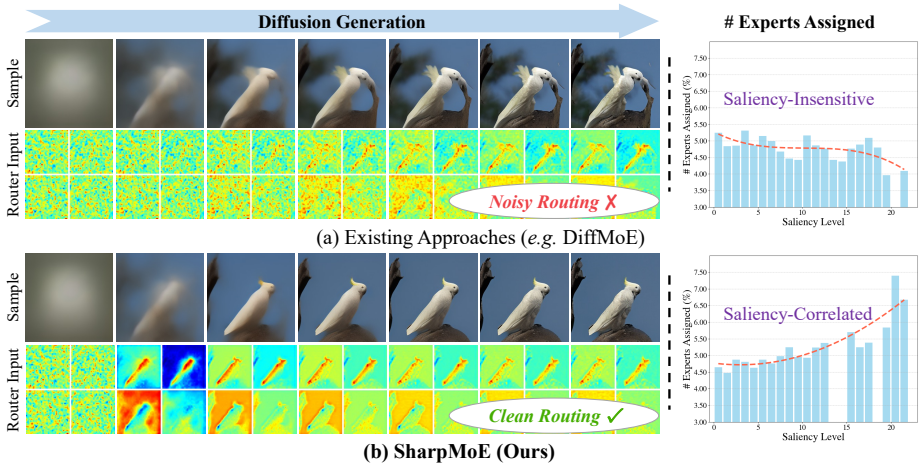


Fig. 1: (Left) Visualization of generated samples and per-channel router inputs. (Right) Distribution of saliency level and the number of assigned experts. (a) Existing methods struggle to differentiate salient tokens due to the use of noise latents for routing, causing a failure in accurate resource assignment. (b) SharpMoE employs clean latents for routing to gain better saliency awareness, thereby achieving a computational allocation that is highly correlated with token saliency.

anism to facilitate *clean routing* for diffusion MoE. The core insight of SharpMoE is to leverage the clean latents (*i.e.*, the $\hat{\boldsymbol{x}}_0$ prediction) from the preceding denoising timestep as the input to the router for the current timestep. This design offers two distinct advantages: (1) *Saliency Awareness*: The predicted clean features explicitly capture and highlight the salient regions of the image. As shown in Fig. 1(b), these features accurately capture the primary object even under the heavy noise present during the early timesteps, delivering robust and well-defined structural guidance to the router. (2) *Temporal Stability*: The noise-free nature of the latent $\hat{\boldsymbol{x}}_0$ ensures a clean and robust input for routing across the entire denoising trajectory. This effectively mitigates the inaccurate resource allocation caused by noise-contaminated inputs in earlier approaches. Encouragingly, while SharpMoE may seem computationally prohibitive due to the full-trajectory training scheme for obtaining $\hat{\boldsymbol{x}}_0$, we demonstrate that it can be efficiently implemented as a post-training algorithm. Our findings reveal that only a limited number of post-training steps (or gradient updates) are sufficient to deliver significant performance gains, even when applied to fully converged pretrained models, highlighting the efficiency and adaptability of SharpMoE.

Moreover, to further promote a saliency-aware allocation of computational resources, we propose the Trajectory Routing Loss, which drives the assignment of computational effort to align with the underlying saliency distribution. Specifically, leveraging the full-trajectory training paradigm in SharpMoE, we quantify the cumulative computational load for each token by aggregating its

expert activations across the denoising sequence. This approach then regulates the computational budget based on saliency, allowing a precise alignment between resource allocation and the visual significance of different regions. By prioritizing salient tokens, the proposed loss concentrates computational capacity on regions with high structural and textural complexity, thereby significantly improving generative fidelity. Extensive experiments conducted on multiple pretrained architectures validate the effectiveness of SharpMoE as a plug-and-play post-training enhancement, further emphasizing the crucial role of clean routing in advancing the capabilities of diffusion MoEs.

In summary, our contributions are fourfold: (1) We identify a critical routing assignment problem in existing diffusion MoEs: the router fails to effectively discriminate salient tokens due to the limitations posed by noisy routing. (2) To address this, we introduce **SharpMoE**, a post-training framework that leverages clean latents as noise-free guidance for the router, effectively capturing critical structural and textural cues for accurate saliency identification. (3) Within SharpMoE, we propose a trajectory routing loss designed to regulate computational resource allocation across the entire generative trajectory, enabling precise saliency-aware routing. (4) Extensive experiments demonstrate that SharpMoE serves as a plug-and-play enhancement to further boost pretrained diffusion MoE models, achieving state-of-the-art performance in visual generation.

2 Related Work

Diffusion Models. Diffusion models [12] have demonstrated remarkable success in visual generation [2, 5, 20, 27, 32, 38]. Early works [23, 24] primarily utilized U-Net [25] backbone optimized via Denoising Diffusion Probabilistic Models (DDPM) [12, 30] objective. More recently, the field has transitioned toward the Diffusion Transformer (DiT) [22] architecture to facilitate model scaling. When combined with the Rectified Flow (RF) training paradigm [19], these DiT-based models [3, 10, 20, 34, 36] have demonstrated superior scalability and synthesis quality, setting new benchmarks for high-fidelity generation.

Mixture of Experts. Mixture-of-Experts (MoE) architectures [15, 28] expand model capacity efficiently by leveraging sparse activation, where only a subset of parameters is activated for each token. While MoE has achieved considerable success in Large Language Models (LLMs), such as DeepSeek-V3 [18] and MiniMax-01 [16], recent efforts [1, 9, 29, 31, 37, 39–41] have focused on adapting MoE to scale dense diffusion models. However, directly migrating MoE designs from LLMs to diffusion frameworks [8, 9] often leads to suboptimal results due to modality differences: text tokens are semantically dense, whereas visual tokens are spatially correlated and redundant. Recent studies have highlighted the heterogeneous complexity of image regions, wherein salient tokens containing critical details demand greater computational resources. This observation has inspired the development of saliency-aware routing mechanisms, such as the expert-choice strategy in EC-DiT [31] and the batch-level pooling approach in

DiffMoE [29] and Expert Race [40]. However, these mechanisms often struggle to achieve precise expert assignments due to the reliance on noisy latents during the denoising process, which obscure the representation of saliency. To address this, we present SharpMoE, which leverages predicted clean latents for routing to provide a robust saliency representation for dynamic expert assignment.

3 Preliminary

Diffusion Models. Diffusion models add Gaussian noise to data and train a neural network to reverse the process. Let $\boldsymbol{x}_0 \sim X_0$ be a sample form the data distribution, and $\boldsymbol{x}_1 \sim X_1$ denote a noise sample, the recent advanced Rectified Flow [19] framework defines the noised data $\boldsymbol{x}_t$ at timestep t as:

$$\boldsymbol{x}_t = t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0. \tag{1}$$

Then, a denoising model is trained to directly regress the velocity field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ by minimizing the Flow Matching [17] objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0\sim X_0,\boldsymbol{x}_1\sim X_1}\left[\|\boldsymbol{v} - \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)\|^2\right], \tag{2}$$

with the target $\boldsymbol{v} = \boldsymbol{x}_1 - \boldsymbol{x}_0$. During generation, the denoising process is formulated as:

$$\boldsymbol{x}_{t+dt} = \boldsymbol{x}_t + dt \cdot \boldsymbol{v}_\theta(\boldsymbol{x}_t, t), \tag{3}$$

where dt is the timestep gap. Therefore, the $\hat{\boldsymbol{x}}_0$ prediction at timestep t is:

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_t - t\boldsymbol{v}_\theta(\boldsymbol{x}_t, t). \tag{4}$$

Mixture of Experts. The Mixture-of-Experts (MoE) adaptively activates a sparse subset of "experts" for each token. In general, a standard MoE layer consists of a router $\mathcal{R}$ and N_E experts $\{E_i\}_{i=1}^{N_E}$, each implemented as a Feed-Forward Network (FFN). Given an input $\boldsymbol{x} \in \mathbb{R}^{B\times S\times D}$, where B, S, and D denote the batch size, token length, and hidden dimension, respectively, the router $\mathcal{R}$ predicts a set of token–expert affinity scores $\boldsymbol{S} \in \mathbb{R}^{B\times S\times N_E}$:

$$\boldsymbol{S} = \mathcal{R}(\boldsymbol{x}). \tag{5}$$

Subsequently, the router selects the experts with the top-k highest scores for computation, and the MoE output is the weighted sum of these experts' outputs:

$$\boldsymbol{G} = \begin{cases} \boldsymbol{S}, & \text{if } \boldsymbol{S} \in \mathrm{TopK}(\boldsymbol{S}, K) \\ 0, & \text{Otherwise} \end{cases}, \quad \mathrm{MoE}(\boldsymbol{x}) = \sum_{i=1}^{N_E} \boldsymbol{G}_i * E_i(\boldsymbol{x}), \tag{6}$$

where $\boldsymbol{G} \in \mathbb{R}^{B\times S\times N_E}$ is the final gating tensor, and $\mathrm{TopK}(\cdot, K)$ is the operation that selects a subset with the K largest value. Notably, the current input $\boldsymbol{x}$ in Eq. 5 is corrupted by the remaining noise during diffusion generation, raising a noisy routing issue where the router fails to accurately differentiate salient tokens, ultimately resulting in incorrect routing assignments.

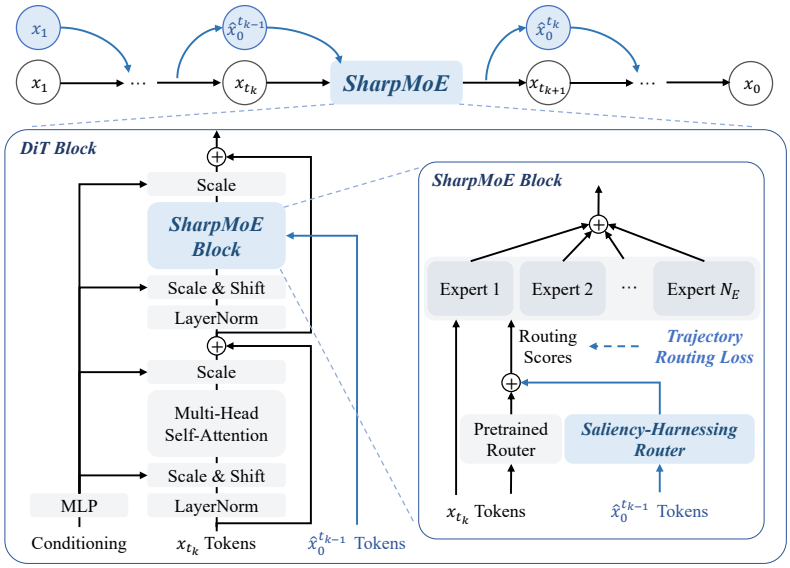


Fig. 2: Overview of SharpMoE architecture. SharpMoE leverages a full-trajectory training scheme where predicted clean latents $\hat{\boldsymbol{x}}_0$ provide saliency guidance for routing. Within each SharpMoE Block, a Saliency-Harnessing Router (taking $\hat{\boldsymbol{x}}_0^{t_{k-1}}$) complements the standard Pretrained Router (taking $\boldsymbol{x}_{t_k}$) for precise expert assignment. The Trajectory Routing Loss is then imposed on the routing scores to align the cumulative compute allocation with the image's saliency distribution across all denoising steps.

4 SharpMoE

4.1 Overview

We build SharpMoE upon the DiT architecture [22], replacing standard feed-forward networks (FFNs) with SharpMoE blocks to facilitate scalable modeling with high efficiency. The core idea of SharpMoE is to transition from *noisy routing* to *clean routing* by harnessing saliency, ensuring that computational resources dynamically focus on salient regions during the denoising process.

As depicted in Fig. 2, SharpMoE introduces three key components to achieve this goal: (1) At a timestep t_k, we depart from the conventional reliance on noisy latents $\boldsymbol{x}_{t_k}$ for routing decisions. Instead, we propose a Saliency-Harnessing Router mechanism (Sec. 4.2) that incorporates the clean prediction $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ from the preceding timestep t_{k-1}, providing robust saliency information. (2) The introduction of $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ creates a recursive dependency across timesteps. To facilitate the availability of these clean latents during training, we develop a Recursive Full-Trajectory Training scheme (Sec. 4.3). Unlike standard single-step denoising, this training strategy enables the model to learn and utilize recursive dependencies across the entire generative trajectory. (3) To regulate cumulative expert allocation, we introduce the Trajectory Routing Loss (Sec. 4.4). By ex-

ploiting the global perspective offered by this training paradigm, this loss aligns the total computational resources with the image's saliency distribution across all timesteps. In the following subsections, we detail each of these components within the SharpMoE framework.

4.2 Saliency-Harnessing Accurate Routing

In diffusion-based generation, visual tokens exhibit non-uniform informational density. Salient tokens, which represent critical object structures and complex textures, necessitate a higher computational budget (*i.e.*, more experts) to ensure generative fidelity. Conversely, redundant background tokens can be processed with fewer experts. While current diffusion MoE frameworks [29, 31] attempt saliency-aware resource allocation, they primarily rely on the current noisy latent $\boldsymbol{x}_{t_k}$ for routing decisions. This reliance gives rise to the *noisy routing* problem: the underlying semantic saliency is heavily obscured by remaining noise, particularly during early timesteps (high noise levels). As a result, routing decisions become erratic and suboptimal, with expert assignments often failing to effectively prioritize computational resources for salient regions.

To tackle the issue of noisy routing, we propose to harness the predicted clean latent $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ as a saliency representation to provide a stable and noise-free guidance signal for routing. This design choice is grounded in two key observations: First, the latent space of the variational autoencoder (VAE) is trained to encode visual information into a semantically dense representation, inherently emphasizing structurally significant regions. As evidenced by Fig. 1, these latent features naturally encapsulate high-level semantic information, such as object regions and textural complexity, both of which are direct indicators of visual saliency. Second, within the diffusion framework, $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ represents the model's denoised projection onto the clean image manifold at timestep t_{k-1}. As visualized in Fig. 1(b), unlike the noise-perturbed $\boldsymbol{x}_{t_k}$ progressively recovers the global structural layout and local details in a denoising manner, $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ provides a stable, noise-free estimation of the image's semantic skeleton, even at early stages where local textures are yet to emerge. By routing based on $\hat{\boldsymbol{x}}_0^{t_{k-1}}$, SharpMoE effectively mitigates the adverse effects of residual noise, enabling the model to accurately anticipate and prioritize salient regions with high precision.

As shown in Fig. 2, the SharpMoE block employs a dual-router mechanism to integrate saliency-aware information into routing assignment, comprising a pretrained router $\mathcal{R}_{pre}$ and a saliency-harnessing router $\mathcal{R}_{sal}$. This architecture is designed as a plug-and-play post-training enhancement for established pretrained diffusion MoE models, enabling a seamless integration of saliency information into existing diffusion MoE blocks. In this setup, the original router from the pretrained backbone is retained as $\mathcal{R}_{pre}$, which assesses the current generation state $\boldsymbol{x}_{t_k}$ and thereby captures the transient requirements of the denoising process at timestep t_k. In parallel, we introduce our saliency-harnessing router $\mathcal{R}_{sal}$. By processing the predicted clean tokens $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ from the preceding step, $\mathcal{R}_{sal}$ embeds saliency-aware guidance into the routing process, aligning expert

Algorithm 1 Recursive Full-Trajectory Training for SharpMoE

Input: SharpMoE model v_θ with saliency-harnessing router $\mathcal{R}_{sal}$, Pre-trained model weights $\mathcal{W}_{pre}$, Rollout steps T, Loss hyperparameter $\lambda_{routing}$.
Initialize:
  $\theta_{\mathcal{R}_{sal}} \leftarrow 0$ {Initialize saliency-harnessing router with zero}
  $\theta_{others} \leftarrow \mathcal{W}_{pre}$ {Initialize other parts with pre-trained weights}
while not converged **do**
  Sample clean data $\boldsymbol{x}_0 \sim X_0$ and noise $\boldsymbol{x}_1 \sim X_1$
  Sample $\{t_k\}_{k=1}^{T} \subset \mathcal{U}(0,1)$ and sort such that $t_1 > t_2 > \cdots > t_T$
  Set $t_1 = 0.999$
  $\mathcal{L}_{total} = 0$
  for $k = 1$ **to** T **do**
    $\boldsymbol{x}_{t_k} = t_k \boldsymbol{x}_1 + (1 - t_k)\boldsymbol{x}_0$ {Initial noisy latent}
    if $k = 1$ **then**
      $\hat{\boldsymbol{x}}_0^{t_0} = \boldsymbol{x}_{t_1}$ {First step: use noise latent as saliency proxy}
    end if
    $\boldsymbol{v}_k = v_\theta(\boldsymbol{x}_{t_k}, t_k, \mathrm{sg}(\hat{\boldsymbol{x}}_0^{t_{k-1}}))$ {Recursive rollout, sg(·): stop-gradient}
    $\mathcal{L}_{fm} = \|(\boldsymbol{x}_1 - \boldsymbol{x}_0) - \boldsymbol{v}_k\|^2$ {Flow Matching objective}
    $\mathcal{L}_{total} \leftarrow \mathcal{L}_{total} + \mathcal{L}_{fm}$
    $\hat{\boldsymbol{x}}_0^{t_k} = \boldsymbol{x}_{t_k} - t_k \boldsymbol{v}_k$ {Derive clean prediction for next step}
    Collect routing scores $\boldsymbol{S}_k$
  end for
  Calculate $\mathcal{L}_{routing}$ with $\{\boldsymbol{S}_k\}$ {Calculate trajectory routing loss}
  $\mathcal{L}_{total} \leftarrow \frac{1}{T}\mathcal{L}_{total} + \lambda_{routing}\mathcal{L}_{routing}$
  Update parameters θ by minimizing $\mathcal{L}_{total}$
end while
return v_θ

allocation with the underlying semantic structure of the image. The final routing scores $\boldsymbol{S}$ are derived by fusing the outputs of both routers:

$$\boldsymbol{S} = \mathcal{R}_{pre}(\boldsymbol{x}_{t_k}) + \mathcal{R}_{sal}(\hat{\boldsymbol{x}}_0^{t_{k-1}}). \tag{7}$$

To ensure the smooth integration of saliency-aware guidance, we initialize the weights of $\mathcal{R}_{sal}$ to zero, allowing the MoE model to progressively incorporate saliency-aware guidance without disrupting its established denoising capabilities in the post-training stage.

4.3 Recursive Full-Trajectory Training

Standard diffusion training typically follows a single-step denoising paradigm. Yet, this approach is inherently incompatible with SharpMoE, as the saliency-harnessing router $\mathcal{R}_{sal}$ requires the predicted clean latent $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ from the preceding denoising step to provide saliency guidance. Since this preceding state is never computed in a standard single-step setting, $\mathcal{R}_{sal}$ becomes impractical to optimize. This intrinsic recursive dependency necessitates a transition from single-step training to a Recursive Full-Trajectory Training scheme.

As depicted at the top of Fig. 2, each training iteration simulates a short-range generation rollout by sampling T consecutive timesteps $\{t_k\}_{k=1}^{T} \subset [0,1]$, where $t_1 > t_2 > \cdots > t_T$. At each timestep t_k, SharpMoE processes the current noisy latent $\boldsymbol{x}_{t_k}$ alongside the predicted clean latent $\hat{\boldsymbol{x}}_0^{t_{k-1}}$ from the preceding step to estimate the velocity field $\boldsymbol{v}_k$. Then we derive the subsequent latent state $\boldsymbol{x}_{t_{k+1}}$ and the clean prediction $\hat{\boldsymbol{x}}_0^{t_k}$:

$$\boldsymbol{x}_{t_{k+1}} = \boldsymbol{x}_{t_k} + (t_{k+1} - t_k) \cdot \boldsymbol{v}_k, \quad \hat{\boldsymbol{x}}_0^{t_k} = \boldsymbol{x}_{t_k} - t_k \cdot \boldsymbol{v}_k. \tag{8}$$

These outputs are subsequently propagated to the next sampling step, enabling a recursive full-trajectory generative rollout throughout the training process.

A practical challenge arises during inference at the initial timestep $t = 1$, where no prior $\hat{\boldsymbol{x}}_0$ is available for the saliency-harnessing router $\mathcal{R}_{sal}$. To resolve this, we use the noise latent $\boldsymbol{x}_1$ as a proxy for the saliency guidance signal. This is motivated by the intuition that during the earliest stage of generation, the object structure and its corresponding saliency are yet to be determined, making the noisy latent a reasonable starting point. To ensure consistency between the training and inference stages, we ideally set the first timestep t_1 of the rollout to 1. In practice, we set $t_1 = 0.999$ as a near-equivalent approximation, as initializing with pure Gaussian noise leads to an unconstrained generation target, which prevents the objective in Eq. 2 from providing meaningful training signals. The complete recursive full-trajectory training scheme is detailed in Algorithm 1.

4.4 Trajectory Routing Loss

To enhance saliency-aware allocation of computational resources, we introduce the Trajectory Routing Loss ($\mathcal{L}_{routing}$), which explicitly aligns the cumulative expert allocation with the saliency distribution. Building upon the full-trajectory training scheme described above, we observe a notable advantage in its ability to provide a holistic perspective on expert allocation throughout the entire generation process. Unlike conventional methods limited to single timesteps, this global view better reflects the sequential nature of diffusion, where single-step snapshots may offer a biased estimate of the total computational load for each token. By considering the entire trajectory, we obtain a more faithful measure of resource distribution. Specifically, for a T-step rollout sequence, we compute the total trajectory-level assignment scores $\mathcal{A}_i$ for the i-th token by aggregating the routing scores of its assigned experts over all steps, as follows:

$$\mathcal{A}_i = \sum_{k=1}^{T} \sum_{l=1}^{L} \sum_{e=1}^{N_E} \mathcal{I}(k,l,e,i) \boldsymbol{S}_{k,l}(e,i), \tag{9}$$

where $\mathcal{I}(k,l,e,i)$ is an indicator function specifying whether the i-th image token is assigned with the e-th expert in the l-th layer at the k-th denoising step, and $\boldsymbol{S}_{k,l}(e,i)$ denotes the corresponding token–expert affinity scores for routing. L is the number of MoE layers, each containing N_E experts.

Meanwhile, we adopt the Laplacian operator to estimate the saliency level of an image. In generative modeling, high-frequency components, comprising intricate textures, sharp edges, and foreground boundaries, are inherently coupled with visual saliency. These regions represent critical structural details that necessitate higher numerical precision and more assigned experts. By calculating the second-order derivatives of the clean image, the Laplacian response effectively isolates areas of high structural density, providing a robust representation for saliency levels. Hence, the target saliency map $\mathcal{M}$ is obtained by passing the clean image $\mathbf{X}_0$ through the Laplacian operator ∇^2:

$$\mathcal{M} = \text{AvgPool}\left(\nabla^2 \mathbf{X}_0\right), \tag{10}$$

where the i-th element $\mathcal{M}_i$ represents the saliency level for the i-th token. To ensure that computational resources are allocated in proportion to the saliency level of each token, we minimize the Kullback-Leibler (KL) divergence between the normalized allocation and saliency:

$$\mathcal{L}_{routing} = D_{KL}\left(\text{softmax}(\mathcal{A}) \parallel \text{softmax}(\mathcal{M})\right). \tag{11}$$

This loss function encourages SharpMoE to prioritize tokens with high saliency throughout the entire generative process. As a result, computational redundancy in less significant background regions is effectively reduced, while the fidelity of crucial foreground details is significantly enhanced. The overall training objective is formulated as a weighted combination of the Flow Matching loss $\mathcal{L}_{fm}$ (Eq. 2) and the trajectory routing loss:

$$\mathcal{L}_{total} = \mathcal{L}_{fm} + \lambda_{routing}\mathcal{L}_{routing}. \tag{12}$$

Here, $\lambda_{routing}$ is a balancing hyperparameter that controls the strength of the routing constraint, which is set to 0.001 in our experiments.

5 Experiment

5.1 Experiment Setup

Baseline and Model Architecture. We compare against Dense-DiT [22], and diffusion MoE approaches, including TC-DiT [9], EC-DiT [31], and DiffMoE [29]. All the methods are evaluated across three standardized model scales (S, B, and L). Building on the pretrained states of these methods, SharpMoE serves as a post-training method by incorporating the proposed saliency-harnessing router into the MoE layers of these diffusion MoE models. Each saliency-harnessing router is designed as a two-layer MLP with SiLU activations. Further architectural details are presented in Sec. A of the supplemental material.

Implementation Detail. Following [29], we perform class-conditional image generation at a resolution of 256×256 using the ImageNet [6] dataset, which contains 1,281,167 training images across 1,000 classes. All models are trained within the

Table 1: Quantitative comparison on ImageNet (256×256). We report FID and IS for models undergoing 100K post-training steps following an initial pre-training phase of 500K steps. All evaluations are conducted under RF with CFG scales of 1.0 and 1.5.

Model	# Activated Params.	# Total Params.	cfg=1.0		cfg=1.5	
			FID50K ↓	IS ↑	FID50K ↓	IS ↑
Dense-DiT-S	33M	33M	55.12	26.17	27.05	58.48
Dense-DiT-B	130M	130M	31.87	47.02	10.07	122.15
Dense-DiT-L	458M	458M	19.02	71.61	4.74	183.54
TC-DiT-S	33M	83M	53.37	27.14	25.82	60.74
+SharpMoE	35M	85M	**46.87**	**31.10**	**20.55**	**73.94**
TC-DiT-B	130M	329M	36.70	41.04	12.82	104.37
+SharpMoE	137M	336M	**31.61**	**48.91**	**9.28**	**130.52**
TC-DiT-L	458M	1.16B	20.53	69.01	5.07	174.98
+SharpMoE	484M	1.18B	**16.63**	**81.29**	**3.72**	**206.93**
EC-DiT-S	33M	83M	50.13	28.54	23.65	64.63
+SharpMoE	35M	85M	**48.78**	**30.15**	**21.81**	**70.56**
EC-DiT-B	130M	329M	30.35	48.90	9.28	126.60
+SharpMoE	137M	336M	**28.10**	**54.14**	**7.67**	**144.50**
EC-DiT-L	458M	1.16B	16.61	78.20	4.09	195.12
+SharpMoE	484M	1.18B	**14.82**	**88.02**	**3.27**	**221.36**
DiffMoE-S	33M	140M	46.55	31.96	20.59	73.46
+SharpMoE	34M	141M	**45.52**	**33.73**	**19.04**	**80.07**
DiffMoE-B	130M	559M	27.50	54.45	8.03	138.49
+SharpMoE	133M	562M	**25.71**	**59.60**	**6.66**	**155.64**
DiffMoE-L	458M	1.98B	15.13	83.62	3.86	203.00
+SharpMoE	470M	1.99B	**13.93**	**93.66**	**3.10**	**228.88**

Rectified Flow (RF) paradigm [19], optimized using AdamW with a learning rate of 1×10^{-4} and a batch size of 256. Besides, we adopt an Exponential Moving Average (EMA) of the model parameters with a decay rate of 0.9999, and all quantitative results reported in this study are computed using the EMA weights. For SharpMoE, we utilize the full-trajectory training scheme with $T = 10$ sampling steps and optimize all network parameters using the loss $\mathcal{L}_{total}$ introduced in Eq. 12. Notably, as TC-DiT distributes computational resources uniformly to each token, $\mathcal{L}_{routing}$ becomes inapplicable, leading to training being performed exclusively with $\mathcal{L}_{fm}$.

Evaluation Metric. We evaluate the image generation quality of all methods using the Fréchet Inception Distance (FID) [7, 11] metric, computed over 50,000 generated samples with 250 sampling steps via Flow Matching Euler. Additionally, we report the Inception Score (IS) [26] to evaluate the diversity of the generated images. A lower FID and a higher IS indicate better performance.

Fig. 3: Samples generated by SharpMoE after 100K post-training steps, based on 500K-pretrained DiffMoE-L, with cfg=4.0.

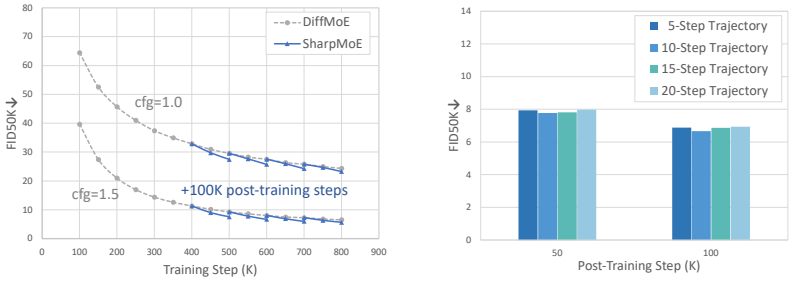
Table 2: Ablation study of each component on ImageNet (256×256), evaluated with CFG scales of 1.0 and 1.5.

Model	cfg=1.0		cfg=1.5	
	FID50K ↓	IS ↑	FID50K ↓	IS ↑
DiffMoE-B	27.50	54.45	8.03	138.49
+ Saliency-Harnessing Routing	25.93	58.33	6.95	152.65
+ Trajectory Routing Loss	**25.71**	**59.60**	**6.66**	**155.64**

5.2 Main Result

We evaluate the performance of SharpMoE against other baselines after 100K post-training steps, with all models initialized from pre-trained checkpoints obtained after 500K training steps. As summarized in Tab. 1, SharpMoE consistently outperforms all competing Diffusion MoE methods, including TC-DiT, EC-DiT, and DiffMoE, across all evaluation metrics, model scales (S, B, and L), and classifier-free guidance (CFG) [13] scales. This consistent superiority demonstrates the broad effectiveness of our saliency-harnessing routing mechanism across diverse MoE-based diffusion architectures. Importantly, these substantial performance improvements are achieved in just 100K post-training iterations, highlighting SharpMoE's efficiency as a versatile, plug-and-play framework to enhance pretrained models, even when they have already converged.

Among all configurations, SharpMoE achieves its strongest performance when applied to the DiffMoE-L backbone, attaining an FID score of 3.10 and an IS of 228.88 using cfg = 1.5. Additionally, even with TC-DiT, which employs a uniform computational resource allocation strategy across all tokens, SharpMoE demonstrates its ability to optimize expert-token assignments by leveraging saliency cues, further improving performance. These quantitative improvements are further validated by the qualitative results presented in Fig. 3, where SharpMoE-



(a) Effect of Pretrained Stage (b) Effect of Full-Trajectory Training Step

Fig. 4: Ablation studies on our critical designs. (a) Effect of Pretrained Stage. SharpMoE consistently boosts various DiffMoE-B checkpoints within only 100K post-training steps. (b) Effect of Training Trajectory Step T. SharpMoE consistently delivers substantial performance gains across various rollout steps, showing high robustness to T.

enhanced models demonstrate exceptional structural fidelity and richer textural details, thereby affirming the effectiveness of the proposed saliency-harnessing routing mechanism.

5.3 Analysis

Effect of Each Component. We conduct an ablation study on ImageNet (256×256) to evaluate the contribution of each component in SharpMoE. The results, summarized in Tab. 2, reveal that integrating the saliency-harnessing routing mechanism leads to a significant performance improvement, reducing the FID from 8.03 to 6.95 at cfg = 1.5. This highlights the critical role of clean latent guidance in enhancing saliency awareness compared to conventional noisy routing methods. Additionally, incorporating the trajectory routing loss further improves the FID to 6.66, underscoring the effectiveness of globally aligning cumulative expert allocation with the saliency distribution of the image. This alignment enables sharper focus of computational resources on the most critical regions, improving overall performance. Together, these complementary components achieve the highest generative fidelity across all CFG scales.

Effect of Pretrained Stage. We evaluate the adaptability of SharpMoE by integrating it into DiffMoE-B checkpoints at different training stages. As shown in Fig. 4(a), SharpMoE consistently achieves a consistent improvement compared to the baseline's original training, regardless of whether it is initialized from a 400K-step or 700K-step pretrained checkpoint. This consistent acceleration in performance underscores SharpMoE's robustness as a plug-and-play enhancement capable of refining expert allocation at any stage of model convergence, even in an already converged state. Remarkably, the fidelity improvements realized within just 100K post-training steps highlight the superior effectiveness of our saliency-harnessing routing as a post-training enhancement.

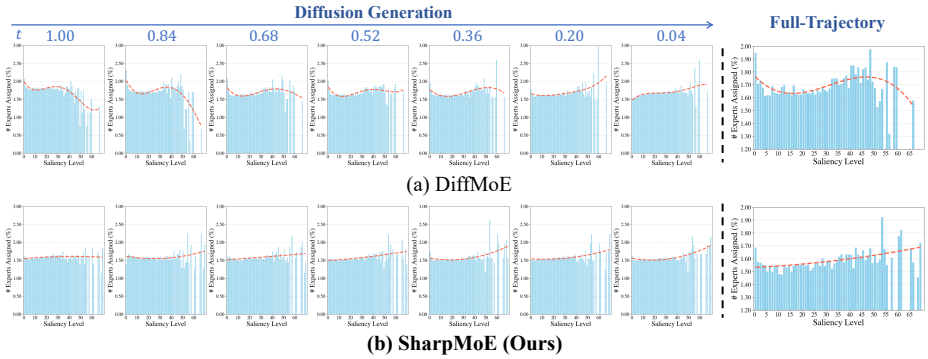


Fig. 5: Aggregated distribution of saliency levels and the number of assigned experts averaged across multiple generated images: (Left) Per-timestep during generation and (Right) Full trajectory. (a) DiffMoE exhibits saliency-insensitive allocation due to the limitations of noisy routing. (b) SharpMoE forms a strong monotonic correlation, prioritizing salient regions, with more notable gains in high-noise stages.

Effect of Full-Trajectory Training Step. We investigate the impact of the rollout step count T within our recursive full-trajectory training scheme. As illustrated in Fig. 4(b), SharpMoE exhibits remarkable robustness to the choice of T, consistently delivering substantial performance gains over the baseline across a wide range of rollout counts (from $T = 5$ to 20). The marginal performance variations across these settings suggest that the saliency-harnessing routing mechanism provides stable and reliable guidance regardless of the specific training step count. This robustness underscores the practicality and adaptability of SharpMoE, as it can be effectively deployed without the need for exhaustive hyperparameter tuning. Among these effective configurations, we empirically select $T = 10$ for our experiments, as it delivers the highest generative fidelity across the spectrum.

Visualization of Expert Allocation. To gain deeper insights into the routing behavior, we visualize the relationship between token saliency and expert assignment of each timestep and the full trajectory in Fig. 5. Using the Laplacian response of the target image as the saliency level, we track the average number of experts assigned to tokens across varying saliency levels within several generated images. Fig. 5(a) illustrates that the baseline DiffMoE exhibits saliency-insensitive allocation, with expert assignments largely uncorrelated with the textural complexity of the tokens, especially at early stages where noise levels are high. This empirically confirms the previously identified routing assignment issue: routers conditioned on noise-corrupted latents struggle to distinguish salient regions from background areas.

In contrast, as shown in Fig. 5(b), SharpMoE establishes a clear monotonic relationship between token saliency and allocated computational resources. The upward-sloping trend indicates that tokens with higher structural or textural richness are consistently assigned more experts. Importantly, the improvement in routing accuracy is most pronounced during high-noise stages, where SharpMoE

successfully harnesses the noise-free latent saliency to guide resource allocation. This alignment underscores the effectiveness of the proposed saliency-harnessing mechanism, which leverages clean latent guidance to provide the router with a stable, noise-free saliency signal. By precisely prioritizing salient tokens, SharpMoE efficiently allocates computational resources to regions critical for generative fidelity, thereby supporting the observed improvements in visual generation.

6 Conclusion

We present SharpMoE, a post-training framework for diffusion MoE that tackles the routing assignment problem: existing routers conditioned on noisy latents struggle to recognize salient tokens, leading to saliency-insensitive compute allocation. Specifically, SharpMoE introduces a saliency-harnessing accurate routing mechanism that employs the clean prediction as the saliency representation, thereby facilitating noise-free routing. Building upon the full-trajectory training scheme, we further propose a trajectory routing loss that aligns the cumulative expert assignment along the denoising rollout with the saliency distribution, enabling saliency-aware resource prioritization. Extensive experiments across multiple diffusion MoEs demonstrate that SharpMoE is plug-and-play, requires only lightweight post-training, and consistently enhances generation quality.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant U22B2053.

References

1. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., et al.: ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
2. Cao, S., Chen, H., Chen, P., Cheng, Y., Cui, Y., Deng, X., Dong, Y., Gong, K., Gu, T., Gu, X., et al.: Hunyuanimage 3.0 technical report. arXiv preprint arXiv:2509.23951 (2025)
3. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al.: Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv preprint arXiv:2310.00426 (2023)
4. Dai, D., Deng, C., Zhao, C., Xu, R., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., et al.: Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. arXiv preprint arXiv:2401.06066 (2024)
5. Deng, H., Yan, K., Mao, C., Wang, X., Liu, Y., Gao, C., Sang, N.: Densegrpo: From sparse to dense reward for flow matching model alignment. arXiv preprint arXiv:2601.20218 (2026)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)

7. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS **34**, 8780–8794 (2021)
8. 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: ICML (2024)
9. Fei, Z., Fan, M., Yu, C., Li, D., Huang, J.: Scaling diffusion transformers to 16 billion parameters. arXiv preprint arXiv:2407.11633 (2024)
10. Hatamizadeh, A., Song, J., Liu, G., Kautz, J., Vahdat, A.: Diffit: Diffusion vision transformers for image generation. In: ECCV. pp. 37–55. Springer (2024)
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS **30** (2017)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS **33**, 6840–6851 (2020)
13. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
14. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)
15. Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., Chen, Z.: Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668 (2020)
16. Li, A., Gong, B., Yang, B., Shan, B., Liu, C., Zhu, C., Zhang, C., Guo, C., Chen, D., Li, D., et al.: Minimax-01: Scaling foundation models with lightning attention. arXiv preprint arXiv:2501.08313 (2025)
17. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
18. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)
19. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
20. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: ECCV. pp. 23–40. Springer (2024)
21. Mao, C., Xie, C.W., Zhong, C., Deng, H., Zhao, J., Xiao, J., Xing, J., Zhang, J., Zhou, J., Zhang, J., et al.: Wan-image: Pushing the boundaries of generative visual intelligence. arXiv preprint arXiv:2604.19858 (2026)
22. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV. pp. 4195–4205 (2023)
23. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
24. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
25. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
26. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. NeurIPS **29** (2016)
27. Seedream, T., Chen, Y., Gao, Y., Gong, L., Guo, M., Guo, Q., Guo, Z., Hou, X., Huang, W., Huang, Y., et al.: Seedream 4.0: Toward next-generation multimodal image generation. arXiv preprint arXiv:2509.20427 (2025)

28. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
29. Shi, M., Yuan, Z., Yang, H., Wang, X., Zheng, M., Tao, X., Zhao, W., Zheng, W., Zhou, J., Lu, J., et al.: Diffmoe: Dynamic token selection for scalable diffusion transformers. arXiv preprint arXiv:2503.14487 (2025)
30. 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)
31. Sun, H., Lei, T., Zhang, B., Li, Y., Huang, H., Pang, R., Dai, B., Du, N.: Ec-dit: Scaling diffusion transformers with adaptive expert-choice routing. arXiv preprint arXiv:2410.02098 (2024)
32. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
33. Wang, L., Gao, H., Zhao, C., Sun, X., Dai, D.: Auxiliary-loss-free load balancing strategy for mixture-of-experts. arXiv preprint arXiv:2408.15664 (2024)
34. Wang, X., Zhang, Z., Zhang, H., Lin, Z., Zhou, Y., Liu, Q., Zhang, S., Li, Y., Liu, S., Zheng, H., et al.: Hbridge: H-shape bridging of heterogeneous experts for unified multimodal understanding and generation. arXiv preprint arXiv:2511.20520 (2025)
35. Wang, Z., Zhu, J., Chen, J.: Remoe: Fully differentiable mixture-of-experts with relu routing. arXiv preprint arXiv:2412.14711 (2024)
36. Wei, Y., Zhang, S., Qing, Z., Yuan, H., Liu, Z., Liu, Y., Zhang, Y., Zhou, J., Shan, H.: Dreamvideo: Composing your dream videos with customized subject and motion. In: CVPR. pp. 6537–6549 (2024)
37. Wei, Y., Zhang, S., Yuan, H., Han, Y., Chen, Z., Wang, J., Zou, D., Liu, X., Zhang, Y., Liu, Y., et al.: Routing matters in moe: Scaling diffusion transformers with explicit routing guidance. arXiv preprint arXiv:2510.24711 (2025)
38. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
39. Xue, Z., Song, G., Guo, Q., Liu, B., Zong, Z., Liu, Y., Luo, P.: Raphael: Text-to-image generation via large mixture of diffusion paths. NeurIPS **36**, 41693–41706 (2023)
40. Yuan, Y., Wang, Z., Huang, Z., Zhu, D., Zhou, X., Yu, J., Min, Q.: Expert race: A flexible routing strategy for scaling diffusion transformer with mixture of experts. arXiv preprint arXiv:2503.16057 (2025)
41. Zhao, W., Han, Y., Tang, J., Wang, K., Song, Y., Huang, G., Wang, F., You, Y.: Dynamic diffusion transformer. arXiv preprint arXiv:2410.03456 (2024)
42. Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A.M., Le, Q.V., Laudon, J., et al.: Mixture-of-experts with expert choice routing. NeurIPS **35**, 7103–7114 (2022)