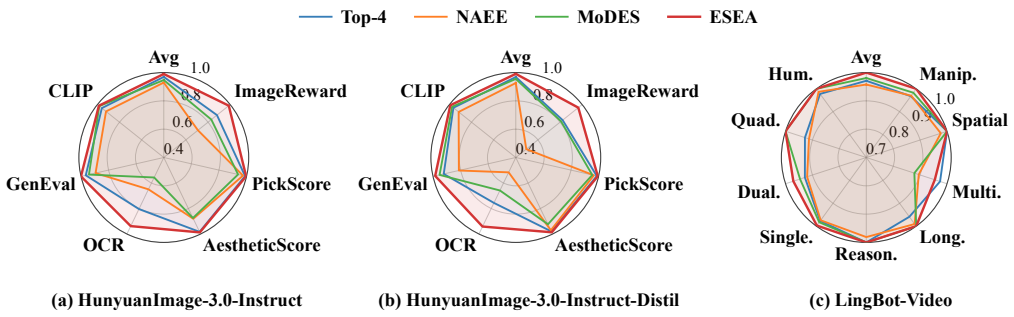
ESEA: ERROR-AWARE SPARSE EXPERT ALLOCATION FOR EFFICIENT DIFFUSION MOE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) has emerged as a promising paradigm for scaling diffusion models, yet activating a fixed number of experts for all tokens leaves computational redundancy. Existing expert skipping methods mainly target LLMs and overlook the iterative denoising nature of diffusion models, leading to degraded generation quality when directly applied to diffusion moe models. We analyze the underlying error mechanisms and identify three insights: (i) approximation error depends on expert similarity; (ii) local errors exhibit timestep–layer heterogeneity and cross-prompt consistency; (iii) the impacts of local errors on global error vary across timesteps. Based on these insights, we propose ESEA (**E**rror-aware **S**parse **E**xpert **A**llocation), a training-free method for diffusion MoE inference. ESEA first adjusts routing weights based on expert similarity to reduce approximation error, then weights calibrated 3D local errors tensor to reflect their timestep-dependent impacts on global error. Finally, dynamic programming minimizes the weighted local error, jointly allocating experts across timesteps and layers under a fixed total expert budget. The routing weight adjustment strategy and expert allocation are determined offline, introducing negligible overhead during inference. Extensive experiments on HunyuanImage-3.0-Instruct, its distilled variant, and LingBot-Video demonstrate that ESEA achieves the best performance among the compared methods under the same expert budget.



(a) HunyuanImage-3.0-Instruct (b) HunyuanImage-3.0-Instruct-Distil (c) LingBot-Video

Figure 1: Comparison of expert skipping methods across three diffusion MoE models at an average Top-k budget of 4. Scores are normalized to the corresponding no expert skipping baseline.

1 INTRODUCTION

Diffusion models have made substantial progress in image (Cai et al., 2025; Zhao et al., 2026; Mao et al., 2026) and video generation (Seedance et al., 2026; Zhu et al., 2026; Chen et al., 2026), enabling high-quality visual content synthesis while highlighting the need to scale model capacity efficiently as generation tasks become increasingly complex. Mixture-of-Experts (MoE) architectures have been widely adopted in large language models (LLMs) (Yang et al., 2025; Xu et al., 2026; Team et al., 2026; Xiao et al., 2026; Zeng et al., 2026), offering an effective approach to balancing model capacity and computational cost through sparse conditional computation. Specifically, MoE organizes feed-forward computation into multiple independent experts and uses a lightweight router

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

to activate only a small subset for each token, allowing the total parameter count to grow without a proportional increase in per-token computation. Recent work has extended this paradigm to diffusion-based generative models (Team, 2025; Ma et al., 2026; Shi et al., 2026), demonstrating its potential for scaling visual generation efficiently. For example, LingBot-Video (Ma et al., 2026), with 30B total parameters and 3B active parameters, approaches the training loss of a 14B dense model while achieving a $2.59\times$ inference speedup over a depth-matched 14B dense baseline at a sequence length of 1M tokens.

Despite these advantages, standard MoE routing activates a fixed number of experts for every token, incurring unnecessary computation and limiting inference efficiency. To mitigate this issue, prior work (Lu et al., 2024; Huang et al., 2024; 2025; Zeng et al., 2024; Huang et al., 2026; Wu et al., 2026) has proposed Dynamic Expert Skipping methods that adaptively reduce the number of active experts during inference. However, directly transferring these methods, which are primarily designed for LLMs and vision-language models (VLMs), to diffusion models can severely degrade generation quality, as shown in Figure 1, largely because they do not adequately account for the iterative nature of denoising and the computational demands that vary across timesteps.

Diffusion models generate visual content through iterative denoising, typically progressing from global structure formation to local detail refinement as the noise level decreases. Feature distributions, routing behaviors, and expert computation requirements evolve throughout this process, meaning that the appropriate number of active experts may vary substantially across timesteps, even within the same network layer. Consequently, ignoring this variation can lead to overly aggressive expert skipping at sensitive stages, compromising generation quality, while retaining excessive computation at more redundant stages and limiting acceleration. Local errors introduced by expert skipping can propagate through subsequent denoising steps, potentially accumulating and becoming amplified along the trajectory. In addition, diffusion inference follows a prefill-only computation pattern, making methods (Oncescu et al., 2025; Vankov et al., 2026; Wu et al., 2026) designed to accelerate the decoding stage in LLMs and VLMs inapplicable to this setting.

To address this problem, we first conduct an in-depth analysis of the error mechanisms underlying dynamic expert skipping in diffusion MoE models and identify three key insights corresponding to approximation error, local error, and global error: (i) The approximation error introduced by Top-k reduction arises from using retained experts to approximate discarded experts and depends on expert similarity. (ii) Local errors exhibit pronounced timestep–layer heterogeneity and maintain consistent patterns across prompts. Specifically, the same Top-k configuration can induce substantially different local errors across denoising timesteps and layers. (iii) Local errors at different timesteps have different impacts on global error, reflecting their timestep-dependent importance. Even local errors of the same magnitude can affect the final output differently, depending on when they are introduced and how they propagate through subsequent denoising steps.

To this end, we propose ESEA (**E**rror-aware **S**parse **E**xpert **A**llocation),a training-free and error-aware expert allocation method for diffusion MoE inference. In response to the first insight, we use the precomputed similarity to determine whether the retained experts' routing weights are rescaled to compensate for discarded contributions. To capture timestep–layer heterogeneity, we construct a three-dimensional local error tensor over denoising timesteps, network layers, and candidate Top-k configurations through offline calibration, and formulate it as a discrete allocation problem under a fixed total expert budget. To account for the timestep-dependent impact of local errors on global error, ESEA further weights these errors according to the noise level at each timestep,aligning with the denoising process. By solving this problem with dynamic programming, we obtain the Top-k configuration that minimizes the errors under the given budget, reducing expert computation while controlling generation quality loss. At inference time, the Top-k configuration and the the routing weight adjustment strategy for each timestep and layer are retrieved from the precomputed configuration table, introducing negligible computational overhead.

In summary, our key contributions are as follows:

- We systematically analyze the error mechanisms underlying dynamic expert skipping in diffusion MoE models, revealing the role of expert similarity in approximation error, pronounced timestep–layer heterogeneity and cross-prompt consistency in local errors, and the timestep-dependent impacts of local errors on global error.

- We propose ESEA (**E**rror-aware **S**parse **E**xpert **A**llocation), a training-free and error-aware expert allocation method for diffusion MoE inference. ESEA reduces approximation error through similarity-based routing weight adjustment and minimizes timestep-weighted local errors via dynamic programming, jointly allocating experts across timesteps and layers under a fixed total expert budget.
- Comprehensive empirical evaluation across HunyuanImage3.0-Instruct, its distilled variant, and LingBot-Video demonstrates that ESEA achieves the best generation quality among the compared methods under the same expert budget.

2 Related Work

Efficient MoE. Existing work on efficient MoE models can be broadly divided into static model compression and dynamic expert skipping. Static compression reduces model size by pruning redundant experts (Lu et al., 2024) or merging experts with similar functions (Chen et al., 2024). Quantization and low-rank decomposition (Hu et al., 2025; Li et al., 2025) further reduce the memory required to store expert weights. While effective in reducing model size, these methods often depend on calibration data or task-specific priors. The removed experts and lost parameter information cannot be restored during inference, which may limit model capacity and generalization.

Dynamic expert skipping improves inference efficiency by reducing unnecessary expert computation. NAEE (Lu et al., 2024) and Top-p routing (Huang et al., 2024) select experts using routing weights and cumulative probabilities, respectively. MC-MoE (Huang et al., 2025) preserves expert computation for important tokens, while AdaMoE (Zeng et al., 2024) introduces null experts to vary the number of active experts. MoDES (Huang et al., 2026) combines routing weights and layer sensitivity with modality-specific skipping thresholds, whereas SERE (Wu et al., 2026) reroutes tokens to similar experts to reduce distinct expert accesses. These methods primarily target LLMs and VLMs, overlooking error propagation during iterative denoising. In contrast, ESEA reduces approximation error and jointly allocates experts across timesteps and layers through local and global error modeling under a fixed total expert budget.

3 Preliminary

Mixture-of-Experts (MoE). MoE layers implement conditional computation through multiple parallel experts, enabling sparse computation by selecting only a small subset of experts for each token. Let the l-th MoE layer contain M routed experts, denoted by $\mathcal{E}^{(l)} = \{E_1^{(l)}, E_2^{(l)}, \dots, E_M^{(l)}\}$, each implemented as an independent multilayer perceptron (MLP). Given an input token representation $x^{(l)} \in \mathbb{R}^d$, where d denotes the hidden dimension, a lightweight router $\mathcal{R}^{(l)}$ produces routing logits $r^{(l)} = \mathcal{R}^{(l)}(x^{(l)}) \in \mathbb{R}^M$ and applies softmax to obtain expert routing probabilities:

$$\pi_m^{(l)} = \frac{\exp\left(r_m^{(l)}\right)}{\sum_{j=1}^{M} \exp\left(r_j^{(l)}\right)}, \qquad m = 1, \dots, M, \tag{1}$$

where $\pi_m^{(l)}$ denotes the routing probability assigned to expert $E_m^{(l)}$ for the current token. By default, the model selects K experts for each token, with $\mathcal{S}_K^{(l)}$ denoting the index set of the K experts with the highest routing probabilities. Top-K routing normalizes the probabilities of the selected experts, yielding the following aggregation weights:

$$w_m^{(l)} = \frac{\pi_m^{(l)}}{\sum_{j \in \mathcal{S}_K^{(l)}} \pi_j^{(l)}}, \qquad m \in \mathcal{S}_K^{(l)}. \tag{2}$$

The outputs of the selected experts are aggregated using these weights to obtain the routed expert output under the default Top-K configuration:

$$y^{(l+1)} = \sum_{m \in \mathcal{S}_K^{(l)}} w_m^{(l)} E_m^{(l)}\left(x^{(l)}\right). \tag{3}$$

By activating only a small subset of experts for each token, MoE can scale its total parameter count while keeping per-token expert computation low.

4 Error Analysis of Dynamic Expert Skipping

To motivate our error-aware expert allocation strategy, we analyze dynamic expert skipping in diffusion MoE models from three perspectives: approximation error, local error, and global error. We establish a theoretical link between expert similarity and approximation error, characterize the timestep–layer heterogeneity of local error, and examine how these local errors propagate through iterative denoising and contribute to global error in the final generated output.

4.1 Expert Similarity and Approximation Error

Consider reducing the number of active experts from K to k, where $1 \leq k < K$. We omit layer indices for simplicity and denote each expert output by $e_m = E_m(x)$. Let $\mathcal{S}_k \subset \mathcal{S}_K$ and $\mathcal{D}_k = \mathcal{S}_K \setminus \mathcal{S}_k$ denote the retained and discarded expert indices, respectively. The original output in Eq. (3) can then be decomposed as

$$y = \sum_{i \in \mathcal{S}_k} w_i e_i + \sum_{j \in \mathcal{D}_k} w_j e_j. \tag{4}$$

Under Top-k routing, the discarded contributions are removed and the retained weights are normalized as in Eq. (2). This normalization can be viewed as transferring the discarded routing weight to the retained experts and using their outputs to approximate those of the discarded experts, as formalized below.

Proposition 4.1 (Expert Weight Redistribution)**.** *Let $\rho_k = \sum_{i \in \mathcal{S}_k} w_i$ and $\delta_k = \sum_{j \in \mathcal{D}_k} w_j$ be the total routing weights of the retained and discarded experts, respectively. With $\gamma_{ij} = w_i w_j / \rho_k$ denoting the routing weight transferred from discarded expert j to retained expert i, the output y' under reduced Top-k routing can be expressed as*

$$y' = \sum_{i \in \mathcal{S}_k} \left(w_i e_i + \sum_{j \in \mathcal{D}_k} \gamma_{ij} e_i \right). \tag{5}$$

The detailed proof of Proposition 4.1 is provided in Appendix A. This proposition shows that Top-k reduction redistributes the discarded routing weight among the retained experts in proportion to their original weights. For a retained expert i, each discarded expert j transfers a fraction w_i/ρ_k of its routing weight to it, so the weighted output $\gamma_{ij} e_j$ is replaced by $\gamma_{ij} e_i$, while the original weighted output $w_i e_i$ remains unchanged.

The resulting approximation error arises from using the retained experts to approximate the discarded experts. Based on the weight redistribution in Proposition 4.1, this error can be expressed as

$$\epsilon = y - y' = \sum_{i \in \mathcal{S}_k} \sum_{j \in \mathcal{D}_k} \gamma_{ij} (e_j - e_i). \tag{6}$$

Eq. (6) highlights the role of expert similarity in determining the approximation error of Top-k reduction. When the similarity between retained and discarded experts is high, redistributing the discarded routing weights can compensate for the missing expert contributions. When this similarity is low, however, redistribution may fail to provide effective compensation and instead increase the approximation error.

Insight 1. The approximation error introduced by Top-k reduction arises from using retained experts to approximate discarded experts and depends on expert similarity.

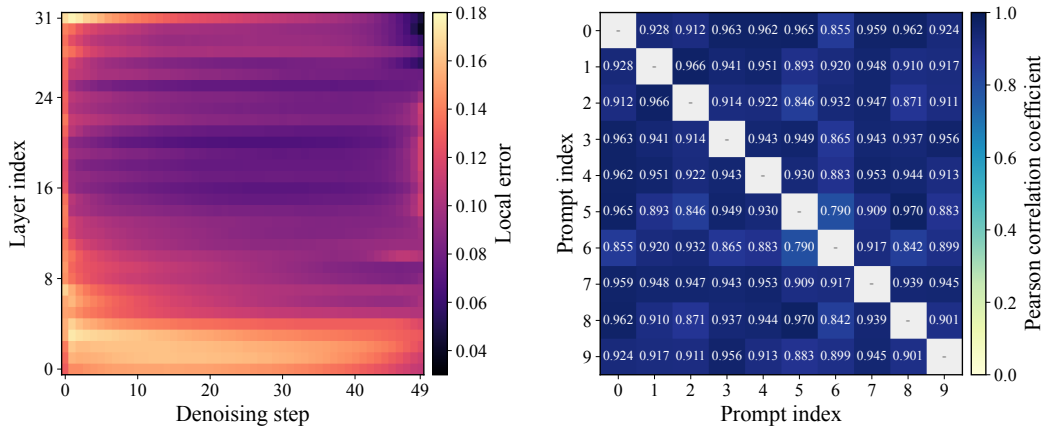
4.2 Timestep–Layer Heterogeneity of Local Error

The preceding analysis characterizes the approximation error under a fixed Top-k configuration. However, the local errors introduced by different Top-k configurations vary across denoising timesteps and network layers. To be specific, we compare the outputs under different Top-k configurations with the default Top-K output and measure local error using one minus the mean token-wise cosine similarity.

Figure 2(a) shows the local error distribution across denoising timesteps and layers with Top-4, averaged over 10 prompts. Even with the same number of active experts, local errors vary substantially across denoising timesteps and network layers, revealing pronounced timestep–layer heterogeneity. A uniform Top-k configuration therefore cannot fully accommodate these varying sensitivities, motivating joint expert budget allocation across timesteps and layers.

Besides, Figure 2(b) presents the pairwise Pearson correlations between local error profiles from different prompts. The high mean correlation of 0.921 reveals consistent timestep–layer error patterns across different prompts, supporting the use of offline calibration to guide expert allocation.

Insight 2. Local errors exhibit pronounced timestep–layer heterogeneity and maintain consistent patterns across prompts.



(a) Timestep–layer local error.

(b) Across-prompt error correlation.

Figure 2: Local error analysis: (a) timestep–layer distribution of local error; (b) pairwise Pearson correlations between local error profiles across prompts.

4.3 Timestep-Dependent Global Error

Section 4.2 shows that local error varies across layers and timesteps, but local error is not equivalent to global error. Local errors introduced by expert skipping propagate through subsequent denoising steps and may accumulate and be amplified, so errors of the same magnitude can affect the final output differently depending on when they occur.

Table 3 in Appendix B reports OCR scores (Liu et al., 2026) for different allocation strategies under the same total expert budget. Specifically, we divide the 50-step sampling process into the first 25 steps (F25) and the last 25 steps (L25) in denoising order, swap the expert budgets between the two halves while keeping the same average number of active experts, and use OCR scores to assess final generation quality.

Table 3 highlights the timestep-dependent nature of global error, excessive expert reduction in either half degrades model performance.Too few activated experts in the early stage can disrupt global structure and introduce errors that accumulate and become amplified through subsequent steps, while too few in the later stage can impair detail recovery and final quality.

Insight 3. Local errors at different timesteps have different impacts on global error, reflecting their timestep-dependent importance.

5 METHODOLOGY

Based on the error analysis in Section 4, we propose ESEA, a training-free and error-aware expert allocation method for efficient diffusion MoE inference, as illustrated in Figure 3. ESEA first reduces approximation error through similarity-based Top-k strategy, then constructs a local error tensor through offline calibration to capture timestep–layer heterogeneity. To account for the timestep-dependent impact of local errors on global error, it weights these errors according to the noise level. Finally, dynamic programming minimizes the weighted local error, jointly allocating experts across timesteps and layers under a fixed total expert budget.

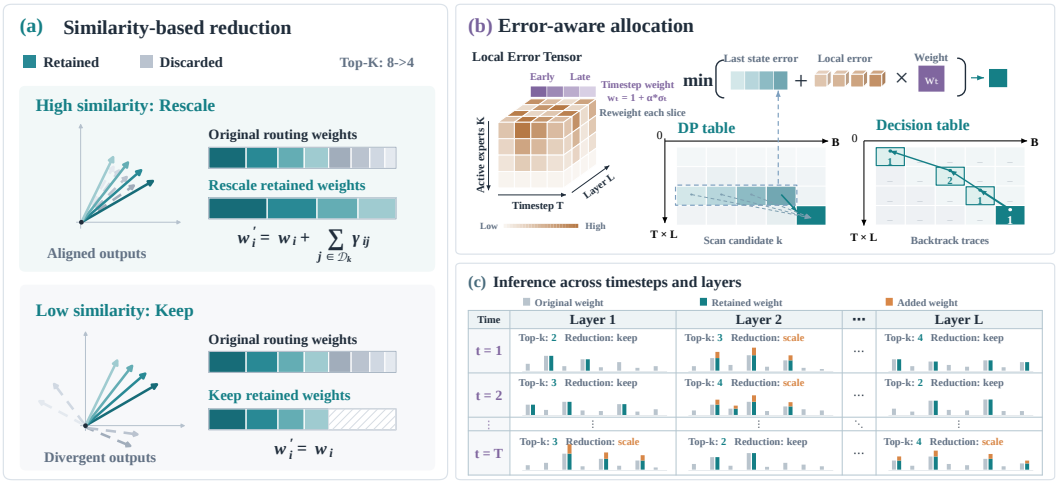


Figure 3: Overview of ESEA. During calibration, ESEA determines routing weight adjustments using expert similarity and jointly allocates experts across timesteps and layers by minimizing timestep-weighted local errors with dynamic programming under a fixed total expert budget. During inference, the precomputed Top-k configurations and weight adjustment decisions are applied directly.

5.1 SIMILARITY-BASED TOP-K REDUCTION

As discussed in Section 4.1, the approximation error introduced by Top-k reduction arises from using retained experts to approximate discarded experts and depends on expert similarity. We therefore propose a similarity-based Top-k reduction strategy: when discarded experts are highly similar to the retained experts, we rescale the retained experts' routing weights; otherwise, we keep their original routing weights unchanged.This strategy is determined during offline calibration to avoid the overhead of computing token-wise expert similarities during inference.

We first run the model with its default Top-K routing to collect calibration inputs at each denoising timestep t and layer l, then feed these inputs to all experts to compute their pairwise expert similarities. For a given Top-k configuration, we use these precomputed similarities to average over all pairs of retained and discarded experts for each token, obtaining a token-level score. Averaging these scores across tokens yields the layer-level similarity $s_{t,l}$ between retained and discarded experts.

$$y' = \begin{cases} \sum_{i \in \mathcal{S}_k} \left(w_i e_i + \sum_{j \in \mathcal{D}_k} \gamma_{ij} e_i \right), & \text{if } s_{t,l} \geq 0.9, \\ \sum_{i \in \mathcal{S}_k} w_i e_i, & \text{if } s_{t,l} < 0.9. \end{cases} \tag{7}$$

Specifically, during inference, the precomputed similarity $s_{t,l}$ determines whether the retained experts' routing weights are rescaled to compensate for discarded contributions, as shown in Eq. (7). This offline decision is shared by all tokens without introducing any additional online overhead, with an empirical comparison of the resulting approximation errors (Eq. (6)) provided in Appendix C.

5.2 LOCAL ERROR-AWARE TOP-K ALLOCATION

Although the Top-k reduction strategy has been determined, a uniform Top-k configuration is not necessarily optimal. As discussed in Insight 2 in Section 4.2, local errors exhibit pronounced timestep–layer heterogeneity and maintain consistent patterns across prompts. This motivates us to formulate it as a discrete allocation problem under a fixed total expert budget and solve it in two stages: local error tensor construction and optimal configuration search.

Stage 1: Local Error Tensor Construction. To capture this timestep–layer heterogeneity, we construct a 3D local error tensor $\mathbf{S} \in \mathbb{R}^{T\times L\times K}$. Each entry $S[t, l, k]$ quantifies the local error introduced by reducing the default Top-K configuration to Top-k at timestep t and layer l. Specifically, this error is computed as one minus the mean token-wise cosine similarity between the reduced output $y_{t,l}^{(k)}$ and the default output $y_{t,l}^{(K)}$:

$$S[t, l, k] = 1 - \cos\left(y_{t,l}^{(k)}, y_{t,l}^{(K)}\right). \tag{8}$$

Moreover, the consistent patterns across prompts observed in Insight 2 suggest that these error profiles are relatively stable across prompts. We therefore average local errors over the calibration set to obtain a reusable error tensor for online expert allocation.

Stage 2: Optimal Configuration Search. Given the local error tensor $\mathbf{S}$, our objective is to identify an optimal Top-k configuration across timesteps and layers under a fixed total expert budget. We define the total cost of a configuration as the sum of its local errors, leading to the following optimization problem:

$$k_{t,l}^{\star} = \arg\min_{\{k_{t,l}\}} \sum_{t=0}^{T-1}\sum_{l=0}^{L-1} S[t, l, k_{t,l}], \quad \text{s.t.} \quad \sum_{t=0}^{T-1}\sum_{l=0}^{L-1} k_{t,l} = B, \quad k_{t,l} \in \{1, \ldots, K\}. \tag{9}$$

where $B = \bar{k}TL$ denotes the total expert budget and $\bar{k}$ is the target average number of active experts.This discrete allocation problem can be efficiently solved using dynamic programming. We flatten the timestep–layer pairs into $N = TL$ decision units, with (t_n, l_n) denoting the pair associated with the n-th unit. Let $D_n(b)$ denote the minimum cumulative local error achievable by the first n units using exactly b units of expert budget. Initializing $D_0(0) = 0$ and all other states to $+\infty$, the recurrence is given by

$$D_n(b) = \min_{1\le k\le \min(K,b)} \{D_{n-1}(b - k) + S[t_n, l_n, k]\}. \tag{10}$$

For each state, we store the minimizing k in a decision table $P_n(b)$. The minimum cumulative local error under the target budget is then given by $D_N(B)$. Starting from $b_N = B$, the optimal configuration is recovered by backtracking:

$$k_{t_n,l_n}^{\star} = P_n(b_n), \qquad b_{n-1} = b_n - k_{t_n,l_n}^{\star}, \quad n = N, \ldots, 1. \tag{11}$$

With $N = TL$ decision units, the DP algorithm considers at most K choices per budget state, yielding time complexity $O(NBK)$ and space complexity $O(NB)$. Given the consistent error patterns across prompts observed in Insight 2, we construct the error tensor using only a small set of calibration prompts, so the offline search incurs negligible overhead.

5.3 GLOBAL ERROR-AWARE TIMESTEP WEIGHTING

The preceding DP minimizes cumulative local error, but this objective does not directly reflect timestep-dependent impact on global error, as discussed in Insight 3 in Section 4.3. This motivates global error-aware timestep weighting to better reflect the differing contributions of local errors during dynamic programming.

To estimate this timestep-dependent importance, we use the noise level as a reference for the denoising stage. Early high-noise timesteps primarily establish global structure, where perturbations can affect subsequent generation, while later low-noise timesteps refine local details. For the flow-shifted sampling schedule, the noise level is defined as

$$u_t = 1 - \frac{t}{T}, \qquad \sigma_t = \frac{s u_t}{1 + (s-1)u_t}, \qquad t = 0, \ldots, T-1, \tag{12}$$

where s is the flow-shift parameter.We then define the timestep weight and the weighted local error tensor as

$$w_t = 1 + \alpha\sigma_t, \qquad \widetilde{S}[t,l,k] = w_t S[t,l,k], \tag{13}$$

Since σ_t decreases throughout denoising, earlier timesteps receive larger weights, encouraging the allocation to preserve more expert computation during global structure formation. However, overly aggressive weighting can leave insufficient budget for later timesteps, impairing detail reconstruction or causing blurred outputs. The parameter α therefore controls the balance between early structure preservation and late detail refinement.

In practice, we select the optimal α through grid search over $[0, 1]$ with a step size of 0.2. By replacing $\mathbf{S}$ with $\widetilde{\mathbf{S}}$ in Eq. 13, we extend the objective from minimizing cumulative local error to minimizing cumulative timestep-dependent global error under the original expert budget:

$$k^{\star}_{t,l} = \arg\min_{\{k_{t,l}\}} \sum_{t=0}^{T-1}\sum_{l=0}^{L-1} w_t S[t,l,k_{t,l}], \quad \text{s.t.} \quad \sum_{t=0}^{T-1}\sum_{l=0}^{L-1} k_{t,l} = B, \quad k_{t,l} \in \{1,\dots,K\}. \tag{14}$$

6 Experiment

6.1 Experimental Setup

Base Models and Baselines. To evaluate the efficacy of our proposed method, we select two representative diffusion MoE models: HunyuanImage-3.0-80B-A13B (Team, 2025) for image generation and LingBot-Video-30B-A3B (Ma et al., 2026) for video generation. Additionally, as there are no expert skipping baselines for diffusion MoE models, we select two representative methods: NAEE (Lu et al., 2024) and MoDES (Huang et al., 2026), both of which we re-implement and adapt strictly following their official settings.

Evaluation Metrics. For HunyuanImage-3.0-Instruct and its distilled variant, we use ImageReward (Xu et al., 2023) to assess overall generation quality, alongside PickScore (Kirstain et al., 2023) and AestheticScore (Verb, 2024) for human preference and visual aesthetics, respectively. We further evaluate text rendering accuracy with OCR (Liu et al., 2026), compositional instruction following with GenEval (Ghosh et al., 2023), and text–image alignment with CLIPScore (Hessel et al., 2021). For LingBot-Video, we follow the official evaluation protocol on RBench (Deng et al., 2026), covering five task categories and four embodiment categories. Speedup is computed relative to the no expert skipping baselines.

Implementation Details. All experiments are implemented in PyTorch and conducted on NVIDIA A800 40GB GPUs for HunyuanImage-3.0 and NVIDIA H20 80GB GPUs for LingBot-Video. Further details regarding evaluation and method implementation are provided in Appendix E.

6.2 Main Result

As shown in Tables 1 and 2, ESEA consistently outperforms uniform Top-4, NAEE, and MoDES in overall generation quality under the same average expert budget. On HunyuanImage-3.0-Instruct, ESEA achieves an ImageReward of 1.0973, compared with 0.9785 for uniform Top-4, 0.7874 for NAEE, and 0.9221 for MoDES, remaining close to the original Top-8 model (1.1114). Beyond overall generation quality, ESEA achieves the highest scores among the compared expert skipping methods on all evaluation metrics. Similar gains are observed on the distilled variant, where ESEA improves ImageReward from 0.8438 to 0.9887 over uniform Top-4. Notably, both NAEE and MoDES obtain lower overall scores than uniform Top-4, suggesting that dynamic expert skipping without accounting for timestep-dependent sensitivity can further degrade generation quality.

On LingBot-Video, ESEA achieves an average RBench score of 0.835, outperforming uniform Top-4 (0.805), NAEE (0.794), and MoDES (0.813). ESEA obtains the best or tied-best scores among the compared expert skipping methods in eight of the nine evaluated categories. Additional results under different Top-k configurations and qualitative comparisons are provided in Appendix D and Appendix G, respectively.

Table 1: Quantitative evaluation of generation quality and inference efficiency on HunyuanImage-3.0-Instruct and its distilled variant. All reduced methods target an average Top-k of 4.

Method	Overall	Image Quality		Text Alignment			Acceleration
	ImageReward↑	PickScore↑	AestheticScore↑	OCR↑	GenEval↑	CLIP↑	Speedup↑
(a) HunyuanImage-3.0-Instruct (80B-A13B)							
Top-8	1.1114	22.3786	5.6999	0.7932	0.8619	29.5973	1.0000×
Top-4	0.9785	22.1368	5.6027	0.6379	0.8320	28.4033	**1.8000×**
NAEE	0.7874	21.7987	5.0172	0.5152	0.7709	27.2839	1.7888×
MoDES	0.9221	20.9976	4.9846	0.4415	0.8121	29.0843	1.7817×
ESEA	**1.0973**	**22.3053**	**5.6267**	**0.7440**	**0.8670**	**29.1966**	1.7962×
(b) HunyuanImage-3.0-Instruct-Distil (80B-A13B)							
Top-8	1.0248	22.1205	5.6707	0.7450	0.8889	28.5997	1.0000×
Top-4	0.8438	21.8349	5.5565	0.5632	0.8197	27.5620	**1.8000×**
NAEE	0.5071	21.3650	5.5051	0.3847	0.7221	26.2040	1.7888×
MoDES	0.8293	21.1046	5.2378	0.4906	0.8465	27.8956	1.7597×
ESEA	**0.9887**	**21.9920**	**5.5938**	**0.7013**	**0.8748**	**28.3670**	1.7824×

Table 2: RBench performance and inference efficiency on LingBot-Video (30B-A3B). All reduced methods use an average Top-k of 4.

LingBot-Video (30B-A3B)											
Method	Overall	Task Scores↑					Embodiment Scores↑				Acceleration
	Avg↑	Manip.	Spatial	Multi.	Long.	Reason.	Single.	Dual.	Quad.	Hum.	Speedup↑
Top-8	0.829	0.928	0.860	0.922	0.947	0.897	0.694	0.723	0.734	0.760	1.0000×
Top-4	0.805	0.898	0.860	**0.898**	0.908	0.912	0.678	0.671	0.681	0.743	**1.8000×**
NAEE	0.794	0.900	0.840	0.825	0.938	0.880	0.676	0.665	0.670	0.750	1.6452×
MoDES	0.813	0.912	0.860	0.810	0.956	0.910	0.682	0.684	0.742	0.762	1.7173×
ESEA	**0.835**	**0.940**	**0.860**	0.880	**0.959**	**0.945**	**0.692**	**0.702**	**0.748**	**0.786**	1.7832×

6.3 Ablation Study

We ablate three components of ESEA: similarity-based Top-k reduction, local error-aware Top-k allocation, and global error-aware timestep weighting. As shown in Tables 6 and 7 in Appendix F, similarity-based reduction improves overall generation quality over uniform Top-4 by adjusting routing weights according to expert similarity. Local error-aware allocation yields additional gains by capturing timestep–layer heterogeneity in local errors, while global error-aware weighting further improves performance by accounting for their timestep-dependent impacts on global error. Across HunyuanImage-3.0-Instruct, its distilled variant, and LingBot-Video, the complete method achieves the best overall performance among all ablation configurations.

7 Conclusion

We present ESEA, a training-free and error-aware expert allocation method for diffusion MoE inference. Our analysis reveals the role of expert similarity in approximation error, timestep–layer heterogeneity in local errors, and the timestep-dependent impacts of local errors on global error. Based on these insights, ESEA combines similarity-based routing weight adjustment with dynamic programming to minimize timestep-weighted local errors, jointly allocating experts across timesteps and layers under a fixed total expert budget. The resulting configurations are determined offline and retrieved during inference with negligible overhead. Experiments on HunyuanImage-3.0-Instruct, its distilled variant, and LingBot-Video demonstrate that ESEA achieves the best overall generation quality among the compared expert skipping methods under the same expert budget.

AI USE STATEMENT

AI tools were used to aid in the writing and polishing of the manuscript. Specifically, we used AI to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that AI was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of AI were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis.

The authors take full responsibility for the content of the manuscript, including any text generated or polished by AI. We have ensured that the AI-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects were involved, and no animal experiments were conducted. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure that the results presented in this paper are reproducible. All datasets used in our experiments, as described in Appendix E.1, are publicly available, ensuring consistent and reproducible evaluation results. The experimental setup, including method implementation, baseline adaptations, evaluation protocols, and generation settings, is described in detail in Section 6.1 and Appendix E. We believe these measures will enable other researchers to reproduce our work and further advance the field.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Guibin Chen, Dixuan Lin, Jiangping Yang, Youqiang Zhang, Zhengcong Fei, Debang Li, Sheng Chen, Chaofeng Ao, Nuo Pang, Yiming Wang, et al. Skyreels-v4: Multi-modal video-audio generation, inpainting and editing model. *arXiv preprint arXiv:2602.21818*, 2026.

I Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, Chun-Yi Lee, et al. Retraining-free merging of sparse moe via hierarchical clustering. *arXiv preprint arXiv:2410.08589*, 2024.

Yufan Deng, Zilin Pan, Hongyu Zhang, Xiaojie Li, Ruoqing Hu, Yufei Ding, Yiming Zou, Yan Zeng, and Daquan Zhou. Rethinking video generation model for the embodied world. In *Forty-third International Conference on Machine Learning*, 2026.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Xing Hu, Zhixuan Chen, Dawei Yang, Zukang Xu, Chen Xu, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. *arXiv preprint arXiv:2505.03804*, 2025.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder task needs more experts: Dynamic routing in moe models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895, 2024.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts llms gains more. In *International Conference on Learning Representations*, volume 2025, pp. 68628–68648, 2025.

Yushi Huang, Zining Wang, Zhihang Yuan, Yifu Ding, Ruihao Gong, Jinyang Guo, Xianglong Liu, and Jun Zhang. Modes: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30205–30215, 2026.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Wei Li, Lujun Li, You-Liang Huang, Mark G Lee, Shengjie Sun, Wei Xue, and Yike Guo. Structured mixture-of-experts llms compression via singular value decomposition. 2025.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6159–6172, 2024.

Shuailei Ma, Jiaqi Liao, Xinyang Wang, Jingjing Wang, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, et al. Scaling mixture-of-experts video pretraining for embodied intelligence. *arXiv preprint arXiv:2607.07675*, 2026.

Chaojie Mao, Chen-Wei Xie, Chongyang Zhong, Haoyou Deng, Jiaxing Zhao, Jie Xiao, Jinbo Xing, Jingfeng Zhang, Jingren Zhou, Jingyi Zhang, et al. Wan-image: Pushing the boundaries of generative visual intelligence. *arXiv preprint arXiv:2604.19858*, 2026.

Costin-Andrei Oncescu, Qingyang Wu, Wai Tong Chung, Robert Wu, Bryan Gopal, Junxiong Wang, Tri Dao, and Ben Athiwaratkun. Opportunistic expert activation: Batch-aware expert routing for faster decode without retraining. *arXiv preprint arXiv:2511.02237*, 2025.

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

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Yangming Shi, Shixiang Zhu, Tao Shen, Zhimiao Yu, Dengsheng Chen, Taicai Chen, Yunfei Yang, Juan Zhou, Chen Cheng, Liang Ma, et al. Mamoda2. 5: Enhancing unified multimodal model with dit-moe. *arXiv preprint arXiv:2605.02641*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Tencent Hunyuan Foundation Model Team. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Daniil Vankov, Nikita Ivkin, Kyle Ulrich, Xiang Song, Ashish Khetan, and George Karypis. Xshare: Collaborative in-batch expert sharing for faster moe inference. *arXiv preprint arXiv:2602.07265*, 2026.

Verb. Aesthetic Predictor V2.5. GitHub repository, 2024. URL `https://github.com/discus0434/aesthetic-predictor-v2-5`.

Juntong Wu, Jialiang Cheng, Fuyu Lv, Dan Ou, and Li Yuan. Sere: Similarity-based expert re-routing for efficient batch decoding in moe models. In *International Conference on Learning Representations*, volume 2026, pp. 56694–56720, 2026.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6223–6235, 2024.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026.

Jiashu Zhu, Yanhao Zheng, Ruitian Tian, Rujing Dang, Shen Zhang, Bingze Song, Jiachen Lei, Ruimin Lin, Jiahong Wu, and Xiangxiang Chu. Dreamx-creator: Democratizing native audio-video generation at 2k resolution. *arXiv preprint arXiv:2608.31106*, 2026.

A Detailed Proof of Proposition 4.1

Restatement of Proposition 4.1 (Expert Weight Redistribution). *Let $\rho_k = \sum_{i\in\mathcal{S}_k} w_i$ and $\delta_k = \sum_{j\in\mathcal{D}_k} w_j$ be the total routing weights of the retained and discarded experts, respectively. With $\gamma_{ij} = w_i w_j/\rho_k$ denoting the routing weight transferred from discarded expert j to retained expert i, the output y' under reduced Top-k routing can be expressed as*

$$y' = \sum_{i\in\mathcal{S}_k} \left(w_i e_i + \sum_{j\in\mathcal{D}_k} \gamma_{ij} e_i \right). \tag{15}$$

Proof. Consider the standard MoE forward pass, whose output can be decomposed into the contributions of the retained and discarded experts:

$$y = \sum_{i\in\mathcal{S}_k} w_i e_i + \sum_{j\in\mathcal{D}_k} w_j e_j. \tag{16}$$

Reducing Top-K to Top-k proceeds in two stages: truncating the discarded experts and normalizing the routing weights of the retained experts. Truncation removes the contributions of the experts in $\mathcal{D}_k$ while leaving the retained weights unchanged, giving

$$y_{\text{trunc}} = y - \sum_{j \in \mathcal{D}_k} w_j e_j = \sum_{i \in \mathcal{S}_k} w_i e_i. \tag{17}$$

Normalization then increases each retained weight by a factor of $1/\rho_k$, so that the retained weights sum to one:

$$\widetilde{w}_i = \frac{w_i}{\rho_k} = w_i + \frac{\delta_k}{\rho_k} w_i, \qquad i \in \mathcal{S}_k. \tag{18}$$

The total increase is $\sum_{i \in \mathcal{S}_k} (\widetilde{w}_i - w_i) = \delta_k$, showing that normalization redistributes the discarded routing weight to the retained experts in proportion to their original weights. For a retained expert $i \in \mathcal{S}_k$, expanding δ_k over the discarded experts reveals where its additional routing weight comes from:

$$\widetilde{w}_i - w_i = \frac{\delta_k}{\rho_k} w_i = \sum_{j \in \mathcal{D}_k} \frac{w_i w_j}{\rho_k} = \sum_{j \in \mathcal{D}_k} \gamma_{ij}, \qquad \gamma_{ij} = \frac{w_i w_j}{\rho_k}. \tag{19}$$

Under this decomposition, each discarded expert j transfers a fraction w_i/ρ_k of its routing weight to expert i. The entire weight of expert j is distributed among the retained experts, since

$$\sum_{i \in \mathcal{S}_k} \gamma_{ij} = \frac{w_j}{\rho_k} \sum_{i \in \mathcal{S}_k} w_i = w_j. \tag{20}$$

The transferred weight γ_{ij} originally multiplies the discarded expert's output e_j, but after redistribution it multiplies the retained expert's output e_i, replacing the weighted output $\gamma_{ij} e_j$ with $\gamma_{ij} e_i$. Substituting $\widetilde{w}_i = w_i + \sum_{j \in \mathcal{D}_k} \gamma_{ij}$ into the reduced MoE output yields the expression in Eq. (5):

$$\begin{aligned} y' &= \sum_{i \in \mathcal{S}_k} \widetilde{w}_i e_i \\ &= \sum_{i \in \mathcal{S}_k} \left(w_i e_i + \sum_{j \in \mathcal{D}_k} \gamma_{ij} e_i \right). \end{aligned} \tag{21}$$

□

B TIMESTEP-DEPENDENT EXPERT ALLOCATION

F25 and L25 denote the first 25 and last 25 steps of the 50-step denoising process, respectively.A moderate shift toward early computation improves OCR scores from 0.6379 with uniform Top-4 to 0.7274 with F25 Top-5/L25 Top-3, whereas further shifting to F25 Top-7/L25 Top-1 lowers the score to 0.1927.

Table 3: OCR scores for different expert allocation configurations at an average Top-k of 4.

F25 Top-k	L25 Top-k	Avg. Top-k	OCR score ↑
8	8	8	0.7932
1	7	4	0.0532
2	6	4	0.1928
3	5	4	0.5118
4	4	4	0.6379
5	**3**	**4**	**0.7274**
6	2	4	0.7227
7	1	4	0.1927

C Empirical Comparison of Approximation Errors

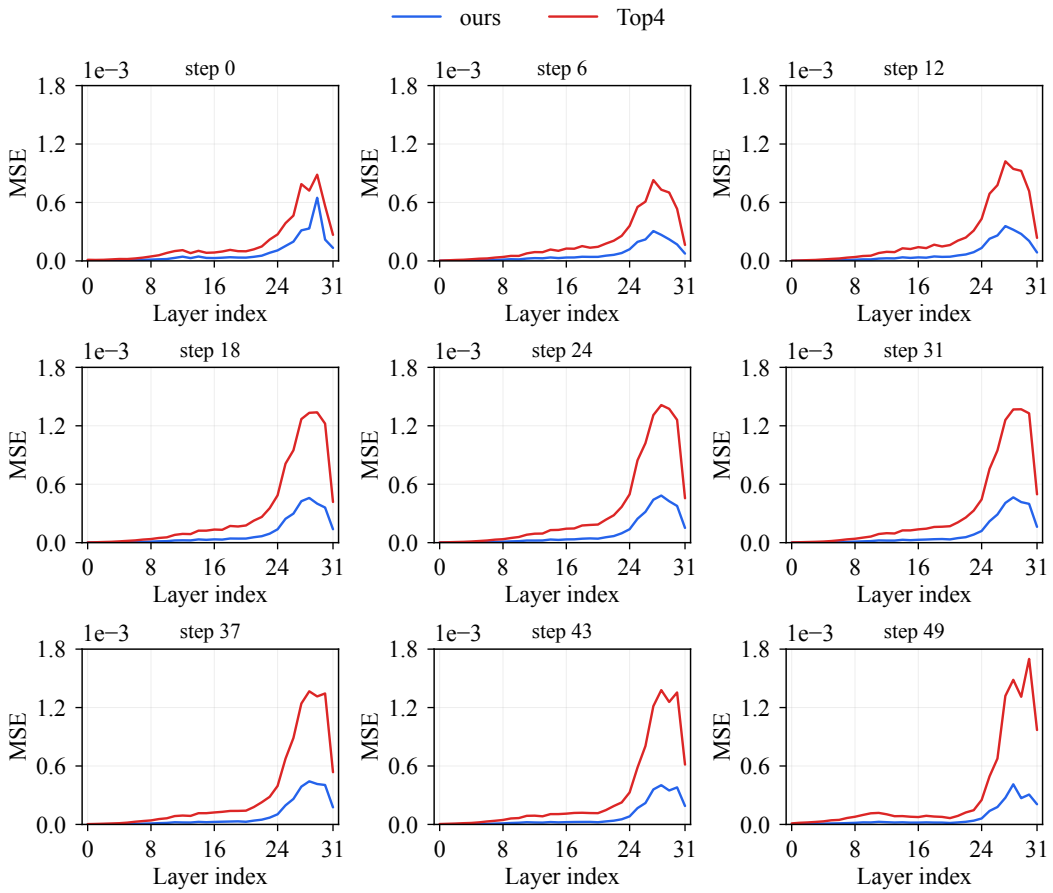


Figure 4: Layer-wise approximation errors of our similarity-based Top-4 reduction and standard Top-4 routing. MSE is measured relative to the original output without expert skipping.

We compare the layer-wise approximation errors of similarity-based Top-4 reduction and standard Top-4 routing at different denoising steps, measuring MSE relative to the original output without expert skipping. As shown in Figure 4, similarity-based reduction consistently lowers approximation errors across the sampled timesteps. The reductions are particularly pronounced in deeper layers, where standard Top-4 routing exhibits large error peaks. These results demonstrate that adjusting routing weights according to expert similarity effectively reduces approximation error at the same number of active experts.

D Additional Results at Different Expert Budgets

We further evaluate ESEA at average Top-k budgets of 5 and 3 on HunyuanImage-3.0-Instruct, its distilled variant, and LingBot-Video. As shown in Tables 4 and 5, ESEA consistently improves overall generation quality over uniform Top-k routing under the same expert budget. The gains are more pronounced at the Top-3 budget, where ESEA increases ImageReward from 0.4660 to 0.9314 on the distilled image model and the average RBench score from 0.755 to 0.835 on LingBot-Video. These results demonstrate the effectiveness of ESEA across different expert budgets, particularly in preserving generation quality under more aggressive expert skipping.

Table 4: Generation quality at average Top-k budgets of 5 and 3 on HunyuanImage-3.0-Instruct and its distilled variant, with the original Top-8 model as a reference.

Method	Overall	Image Quality		Text Alignment		
	ImageReward↑	PickScore↑	AestheticScore↑	OCR↑	GenEval↑	CLIP↑
(a) HunyuanImage-3.0-Instruct (80B-A13B)						
Top-8	1.1114	22.3786	5.6999	0.7932	0.8619	29.5973
Top-5	1.0298	22.2754	5.6119	0.7152	0.8603	28.8836
ESEA (Top-5)	**1.1263**	**22.3704**	**5.6551**	**0.7761**	**0.8736**	**29.4374**
Top-3	0.8555	21.8807	**5.5327**	0.4765	0.7724	27.5820
ESEA (Top-3)	**1.1048**	**22.0807**	5.5045	**0.7024**	**0.8450**	**29.1341**
(b) HunyuanImage-3.0-Instruct-Distil (80B-A13B)						
Top-8	1.0248	22.1205	5.6707	0.7450	0.8889	28.5997
Top-5	0.9249	22.0005	5.6084	0.6915	0.8556	28.3148
ESEA (Top-5)	**0.9936**	**22.0789**	**5.6526**	**0.7339**	**0.8887**	**28.3989**
Top-3	0.4660	21.2465	5.3951	0.2851	0.7382	25.9148
ESEA (Top-3)	**0.9314**	**21.7520**	**5.5323**	**0.6531**	**0.8622**	**27.8377**

Table 5: RBench performance at average Top-k budgets of 5 and 3 on LingBot-Video (30B-A3B), with the original Top-8 model as a reference.

LingBot-Video (30B-A3B)										
Method	Overall	Task Scores↑					Embodiment Scores↑			
	Avg↑	Manip.	Spatial	Multi.	Long.	Reason.	Single.	Dual.	Quad.	Hum.
Top-8	0.829	0.928	0.860	0.922	0.947	0.897	0.694	0.723	0.734	0.760
Top-5	0.801	0.910	**0.820**	0.855	0.935	0.872	0.672	0.684	0.710	0.755
ESEA (Top-5)	**0.818**	**0.920**	0.780	**0.890**	**0.944**	**0.912**	**0.698**	**0.692**	**0.754**	**0.773**
Top-3	0.755	0.855	0.830	0.760	0.893	0.891	0.637	0.626	0.610	0.696
ESEA (Top-3)	**0.835**	**0.928**	**0.880**	**0.908**	**0.934**	**0.930**	**0.720**	**0.688**	**0.774**	**0.753**

E Additional Experimental Details

E.1 Evaluation

For HunyuanImage-3.0, we evaluate both the Instruct and Instruct-Distil variants at a resolution of 1024×1024 with 50 and 8 NFEs, respectively. Each evaluation uses a subset of 300 test prompts sampled randomly from the corresponding test split. We use the official GenEval test prompts (Ghosh et al., 2023) to evaluate compositional instruction following, while ImageReward, AestheticScore, and CLIPScore are evaluated on DrawBench (Saharia et al., 2022). PickScore is evaluated on Pick-a-Pic (Kirstain et al., 2023), and OCR uses the visual text rendering test split provided by Flow-GRPO (Liu et al., 2026).

For LingBot-Video, we generate 121-frame videos at 24 FPS with 40 NFEs, with the output resolution matching that of the corresponding RBench reference images. And we evaluate on the complete RBench benchmark (Deng et al., 2026), which contains 650 text–image prompts. The task-oriented split consists of 250 prompts, with 50 prompts for each of five categories: common manipulation, spatial relationship, multi-entity collaboration, long-horizon planning, and visual reasoning. The embodiment-oriented split consists of 400 prompts, with 100 prompts for each of four robot morphologies: single-arm, dual-arm, humanoid, and quadruped robots. In addition, we use Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025) as the VLM evaluator.

E.2 Method Implementation

We strictly follow the expert-skipping rules in the original papers, adopting the generalized Top-K NAEE implementation described in MoDES (Huang et al., 2026), with details below.

MoDES. Following MoDES (Huang et al., 2026), we combine an offline-calibrated layer importance factor $\alpha^{(l)}$ with each token's routing weight $w_m^{(l)}$, skipping experts for which $\alpha^{(l)} w_m^{(l)} < \tau$. In our diffusion adaptation, routed tokens are treated as a single modality, reducing the original dual-modality threshold search to a one-dimensional search over a single threshold τ shared across layers and denoising timesteps. Layer importance is calibrated by disabling one layer's routed experts throughout the denoising trajectory, measuring the MSE between the resulting final latent and the unmodified model's final latent, and normalizing these errors across layers.

NAEE. We implement NAEE (Lu et al., 2024) using the Top-K extension in MoDES (Huang et al., 2026). For each token, the original K selected experts are ordered by decreasing routing weight, denoted by $w_{(1)}^{(l)} \geq \cdots \geq w_{(K)}^{(l)}$, and the largest low-weight suffix satisfying

$$\sum_{r=i}^{K} w_{(r)}^{(l)} < \beta^{(l)} \sum_{r=1}^{K} w_{(r)}^{(l)}, \qquad 2 \leq i \leq K, \tag{22}$$

is skipped, ensuring that at least one expert remains active; if no suffix satisfies this condition, all K experts are retained. The layer-specific thresholds $\beta^{(l)}$ are optimized by genetic search under the target average expert budget constraint.

F Component Ablation Results

Table 6: Component ablation on HunyuanImage-3.0-Instruct and its distilled variant at an average Top-k of 4. Components: (1) Similarity-Based Top-k Reduction; (2) Local Error-Aware Top-k Allocation; (3) Global Error-Aware Timestep Weighting. Disabling all components gives uniform Top-4, while enabling all three gives ESEA.

Components			Overall	Image Quality		Text Alignment		
(1)	(2)	(3)	ImageReward↑	PickScore↑	AestheticScore↑	OCR↑	GenEval↑	CLIP↑
(a) HunyuanImage-3.0-Instruct (80B-A13B)								
–	–	–	0.9785	22.1368	5.6027	0.6379	0.8320	28.4033
✓	–	–	1.0532	22.2324	5.6063	0.7225	0.8466	29.0580
✓	✓	–	1.0895	22.2971	5.6167	0.7411	**0.8702**	29.1875
✓	✓	✓	**1.0973**	**22.3053**	**5.6267**	**0.7440**	0.8670	**29.1966**
(b) HunyuanImage-3.0-Instruct-Distil (80B-A13B)								
–	–	–	0.8438	21.8349	5.5565	0.5632	0.8197	27.5620
✓	–	–	0.9426	21.9229	5.5358	**0.7151**	0.8716	28.1332
✓	✓	–	0.9693	21.9800	5.5348	0.6982	0.8614	28.2034
✓	✓	✓	**0.9887**	**21.9920**	**5.5938**	0.7013	**0.8748**	**28.3670**

Table 7: Component ablation on LingBot-Video (30B-A3B) at an average Top-k of 4. Components (1)–(3) follow Table 6.

LingBot-Video (30B-A3B)											
Components			Overall	Task Scores↑					Embodiment Scores↑			
(1)	(2)	(3)	Avg↑	Manip.	Spatial	Multi.	Long.	Reason.	Single.	Dual.	Quad.	Hum.
–	–	–	0.805	0.898	0.860	0.898	0.908	0.912	0.678	0.671	0.681	0.743
✓	–	–	0.827	0.925	0.860	0.880	0.940	0.915	0.709	**0.706**	0.746	0.766
✓	✓	–	0.828	0.928	0.840	**0.902**	0.948	0.932	**0.711**	0.678	0.746	0.768
✓	✓	✓	**0.835**	**0.940**	**0.860**	0.880	**0.959**	**0.945**	0.692	0.702	**0.748**	**0.786**

G QUALITATIVE COMPARISONS

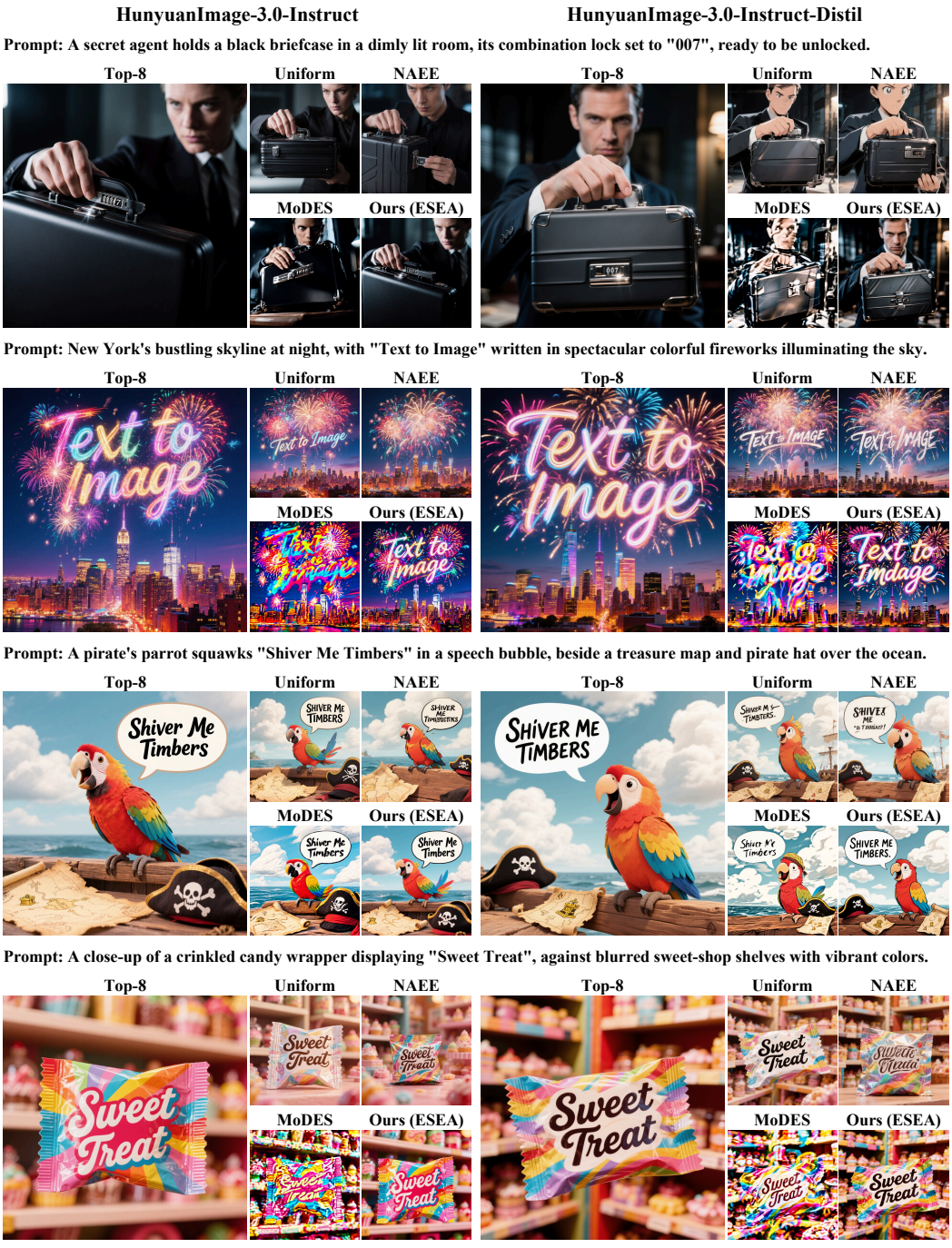


Figure 5: Qualitative comparisons on HunyuanImage-3.0-Instruct (left) and its distilled variant (right).

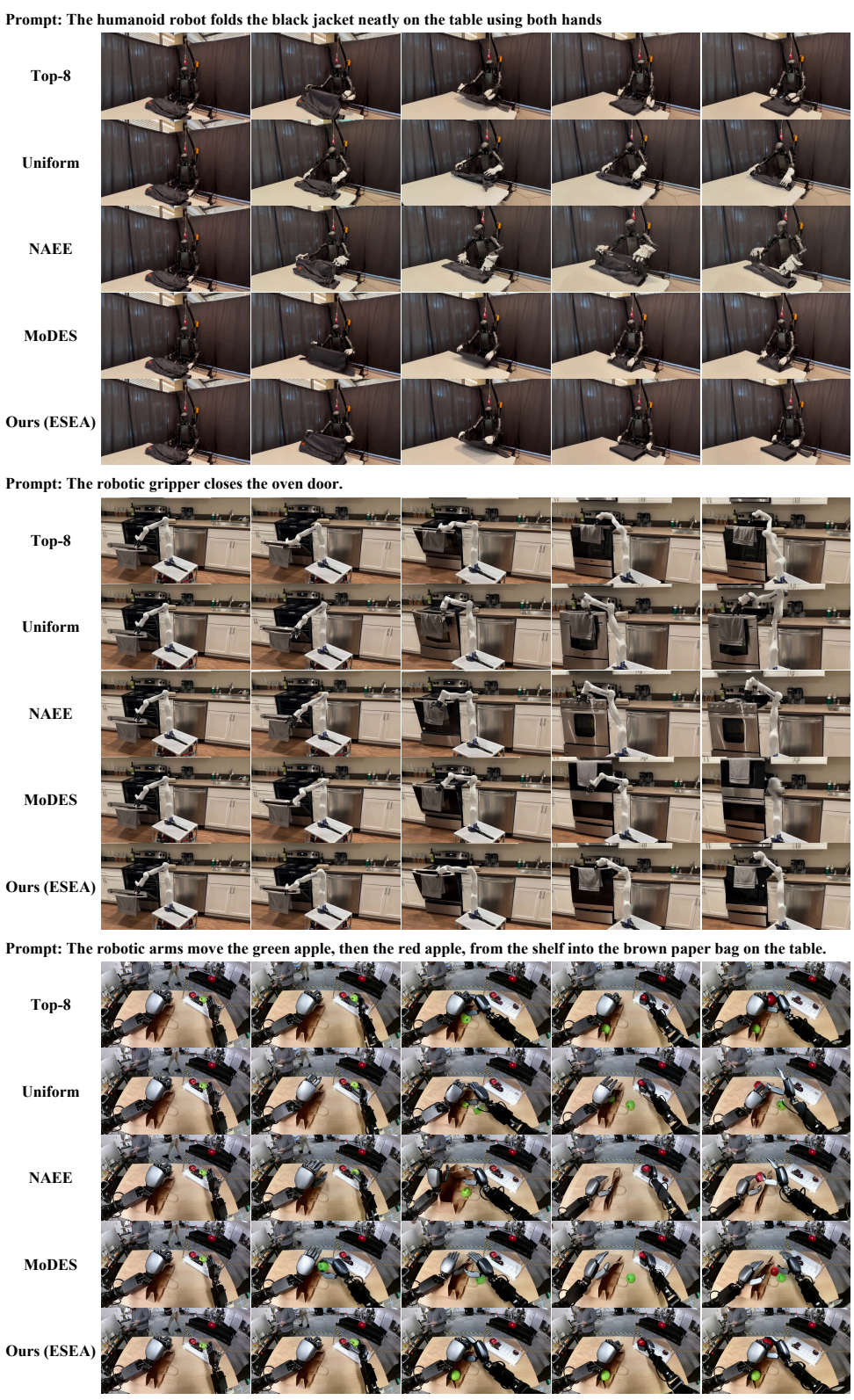


Figure 6: Qualitative comparisons. The frames are uniformly sampled from the generated video.