MoB: Mixture of Block Transformer for Accelerating Video Generation with Dynamic Routing

Anonymous authors
Paper under double-blind review

Abstract

Diffusion Transformers (DiTs) have demonstrated exceptional performance in high-fidelity image and video generation tasks. However, their iterative denoising process introduces substantial computational redundancy within Transformer modules, resulting in prohibitively high computational costs and slow inference speeds. Through comprehensive experimental analysis of existing DiTs, we reveal two key observations: (1) outputs of different Transformer blocks exhibit significant similarity during the denoising process, and (2) block-level redundancy varies dynamically across denoising timesteps. Based on these insights, we propose **Mixture of Blocks (MoB)**, the first framework to introduce block-level dynamic routing for DiT acceleration. The core innovation of MoB lies in a lightweight routing network that dynamically evaluates the importance of each Transformer block based on input prompts. At each denoising step, we propose the Ada-Top-k mechanism which selects relevant blocks by using the k-th largest score as an adaptive threshold, avoiding the winner-take-all problem of traditional soft selection while eliminating 10-20% of redundant computations. To compensate for information loss from skipped blocks, we design a Block Cache mechanism that maintains generation quality by reusing intermediate feature differences from previous timesteps. Furthermore, MoB integrates adaptive timestep skipping and employs knowledge distillation to train the routing network, achieving enhanced inference efficiency and training stability. In addition, we evaluate its generalization ability on image generation tasks using Flux.1. Extensive experiments demonstrate that MoB achieves significant inference acceleration while preserving generation fidelity in both video and image generation tasks, consistently outperforming existing baseline methods in both efficiency and quality.

1 Introduction

Video generation models based on the Diffusion Transformer (DiT) architecture remain computationally expensive (He et al., 2024; Yuan et al., 2024a; Fei et al., 2025; Zhang et al., 2025). These models typically start from a random noise initialization and iteratively denoise over multiple steps. While effective in producing high-fidelity results, this multi-step sampling trajectory incurs substantial computational overhead and poses challenges for real-time or interactive applications. Consequently, accelerating inference in DiT-based models has become a critical research problem.

Early acceleration methods primarily targeted the diffusion sampling process itself, such as DDIM (Song et al., 2022) and DPM-Solver (Lu et al., 2022). Although these approaches reduce the number of sampling steps, they still require multiple iterations and cannot fundamentally eliminate structural redundancy. Distillation-based methods (Zhai et al., 2024) compress hundreds of sampling steps into only a few via a teacher–student framework, but the distillation process remains computationally demanding. Hardware-aware optimizations, such as FlashAttention (Dao et al., 2022), offer composable low-level speedups but provide limited cross-hardware portability and only constrained algorithmic improvements.

Beyond these, architecture-levelstrategies, such as pruning (Wu et al., 2024), low-rank decomposition (Hu et al., 2025), and dynamic routing (Sun et al., 2025; Shi et al., 2025; Xi et al., 2025)—di-

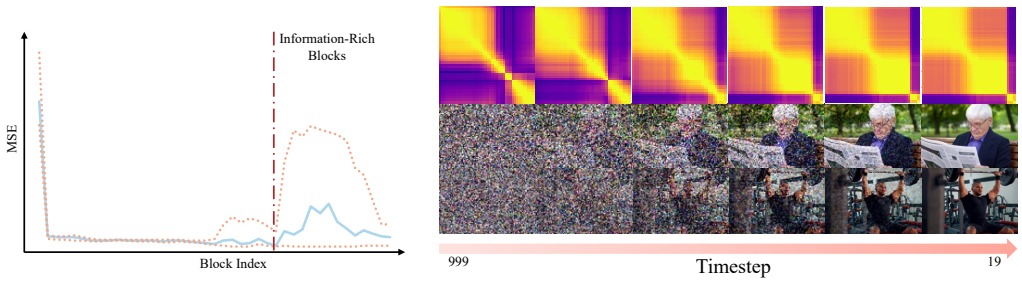


Figure 1: Block output analysis in CogVideoX-5B. **Left:** MSE between outputs obtained by skipping adjacent blocks. The blue solid line indicates the average MSE across all prompts, while the yellow dashed lines denote the maximum and minimum values. Blocks with higher MSE values are regarded as more informative, whereas those with lower MSE values can potentially be skipped without incurring significant performance degradation. **Right:** Cosine similarity between block outputs across timesteps. In the heatmap, both axes represent block indices, with brighter colors indicating higher similarity. The results highlight the presence of block-level redundancy and motivate the use of dynamic routing in MoB.

rectly optimize the generative model at the algorithmic level. Among structural optimization techniques, dynamic routing has typically been applied within Transformer blocks to specific components (e.g., attention (Jin et al., 2025a) in Figure 2(a) or MLP (Fei et al., 2024) in Figure 2(b)), motivated by its similarity to Mixture-of-Experts architectures. In contrast, block-level optimizations have largely relied on pruning or low-rank decomposition. However, these methods neglect the input sensitivity of entire Transformer blocks, thereby limiting the model's ability to adaptively control depth during inference.

Following the approach of Daniel Verdú (2024), which prunes blocks with small input–output mean squared error (MSE) values to reduce model size from 12B to 8B parameters, we conduct analogous experiments on CogVideoX-5B (Yang et al., 2025) using a randomly selected batch of prompts from the Lin et al. (2014). Specifically, n prompts (e.g., 200) are input into CogVideoX-5B, and the generation process is manually terminated after specific Transformer blocks. As shown in Figure 1, for each case, we compute the MSE between the outputs of adjacent blocks. The averaged results, together with their upper and lower bounds across 200 measurements. To further assess the effect of timesteps on block outputs, we also record the outputs of all blocks at multiple timesteps and plot cosine similarity maps across blocks over different denoising stage.

Our findings reveal that, although the contribution of a given Transformer block varies across inputs, there consistently exist block pairs whose output differences remain minimal and, in some cases, negligible. Similar observations are reported in Daniel Verdú (2024), which shows that blocks with small input–output differences exert limited influence on the final generation quality. These results indicate that DiT-based video generation models exhibit substantial computational redundancy at the block level. Moreover, block similarity is relatively low in the early stages of denoising but becomes more pronounced in later stages. This suggests that block-level redundancy increases as denoising progresses, consistent with prior findings on block-level pruning (Zhao et al., 2025; Chen et al., 2024).

Motivated by these insights, we propose the **Mixture of Block Transformer (MoB)**, a DiT-based text-to-video generation framework that employs block-level dynamic routing. As illustrated in Figure 2(c), MoB explicitly selects and activates only the most relevant blocks, thereby reducing inference cost while preserving high-quality generation performance.

MoB introduces a dynamic routing network that computes block-wise relevance scores from input text features. These scores are processed through our **Ada-Top-k** mechanism, which uses the k-th largest score as an adaptive threshold rather than traditional competitive normalization. This rank-based selection strategy activates blocks whose scores exceed the threshold, effectively pruning redundant computations while avoiding the winner-take-all collapse inherent in softmax-based approaches. Concretely, MoB processes input text embeddings to generate relevance scores for each DiT block, then applies Ada-Top-k to determine which blocks should be executed at each iteration of the denoising trajectory.

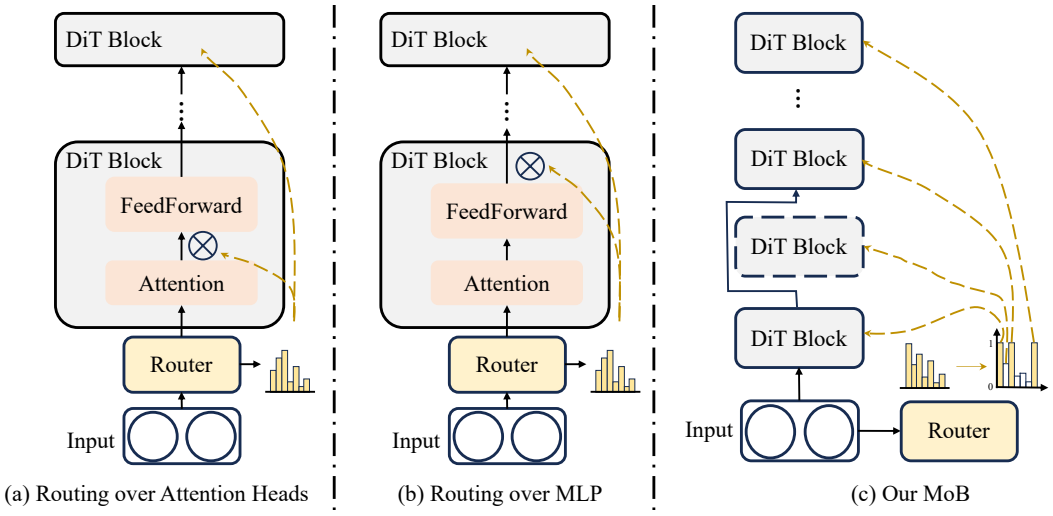


Figure 2: Utilization of three types of dynamic routing within the DiT framework.

To compensate for potential information loss from skipped blocks, we incorporate a **block cache module** that aggregates intermediate representations into a unified output. Furthermore, MoB adopts **knowledge distillation** and **load balancing** strategies, which improve training efficiency and enable effective fine-tuning on small-scale datasets, thereby enhancing the framework's practicality and deployability.

2 METHODOLOGY

We propose the **Mixture of Block Transformer (MoB)**, an optimization framework built upon the CogVideoX-5B architecture. MoB employs a routing network to dynamically select Transformer blocks, complemented by fine-grained design choices to address challenges in practical deployment.

Prior to the denoising process, the text embedding is passed through the routing network, where global pooling followed by a linear projection produces a vector of routing scores, each corresponding to a Transformer block. During inference, Ada-Top-k strategy is proposed to select the indices of the k most relevant blocks, which are then activated.

By integrating information across blocks, MoB effectively leverages the representational capacity of all Transformer blocks, thereby accelerating the video generation process without compromising output quality. Furthermore, MoB incorporates a knowledge distillation mechanism to reduce the reliance on large-scale training data, enabling practical adaptation through fine-tuning on smaller datasets.

In terms of the training objective, MoB introduces load balancing strategies into its loss formulation to ensure stable and efficient optimization. These collective design choices empower MoB to produce high-quality outputs with reduced computational overhead. The detailed architecture of **MoB** is shown in Figure 3. During training, the Ada-Top-k mechanism is used to approximate the skipping of block computations. The output of each block is formulated as the sum of its input and the weighted contribution of the block transformation. Since no block is skipped during training, the Block Cache module (Section 2.2) records only the output differences rather than performing full computations, thereby ensuring sufficient training of the routing network.

2.1 ROUTING NETWORK

In MoB, the dynamic routing network is placed before the denoising network and takes as input the text embeddings $z_{\text{text}} \in \mathbb{R}^{B\times T\times C}$ from the text encoder, where B is the batch size, T is the token sequence length, and C is the embedding dimension. The routing network first applies average pooling along the T-dimension to compress token-level representations into a global representation $z_{\text{cond}} \in \mathbb{R}^{B\times C}$.

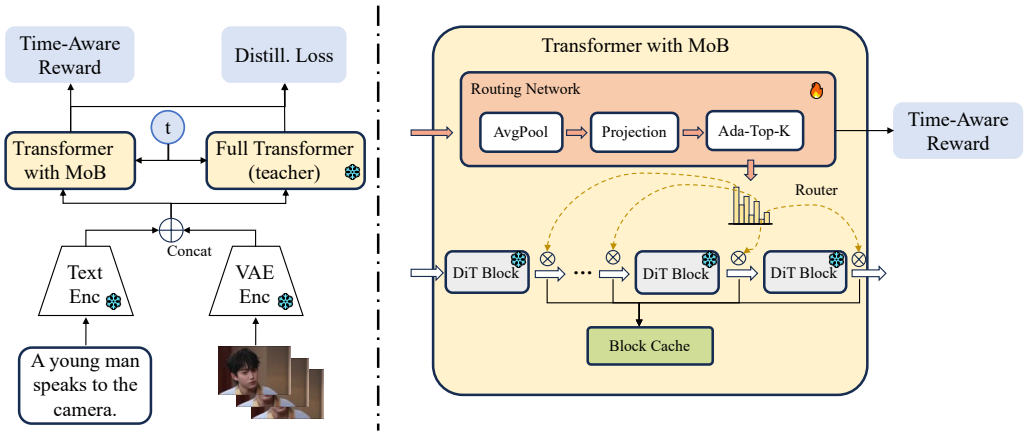


Figure 3: Training Pipeline of MoB. We train only the dynamic routing network while keeping all Transformer block parameters frozen. The routing network employs Ada-Top-k to generate soft selection weights for each block during training, enabling gradient propagation through all blocks. The training objective combines knowledge distillation loss from the teacher model with a time-aware reward term that encourages sparse block activation.

$$z_{\text{cond}} = \frac{1}{T}\sum_{t=1}^{T} z_{\text{text}}[:, t, :] \tag{1}$$

Subsequently, a fully connected layer projects z_{cond} into a space of dimension equal to the number of Transformer blocks N, yielding block-wise importance scores $s \in \mathbb{R}^{B\times N}$, as shown in Equation 2, where W and b are learnable parameters.

$$s = W z_{\text{cond}} + b \tag{2}$$

Ada-Top-k. In MoB, dynamic routing is performed at the Transformer block level, where each block is executed sequentially. Consequently, the conventional MoE approach—normalizing expert scores s into weights via a softmax operation—is not directly applicable. Such normalization would assign extremely small weights to individual blocks, thereby impeding convergence during training. Conversely, directly applying Top-k or Top-p strategies is problematic, as these operations are non-differentiable and lead to unstable gradient flow during optimization.

The standard Soft-Top-k approximates Top-k selection via softmax with temperature scaling T_{temp}. However, lowering the temperature sharpens the Soft-Top-k toward a one-hot distribution, conflicting with the intended k-hot outcome

Algorithm 1 Dynamic Block Routing

Require: Text embeddings $\boldsymbol{z}_{text}$; Transformer with N blocks $\{f_1, \ldots, f_N\}$; Router g; budget k; init $\boldsymbol{h_0}$
Ensure: Output $\boldsymbol{h'}$

```
r ← g(z_text)
h ← h_0
for n = 1 to N do
    if r_n ≈ 1 then
        h ← f_n(h)
    else
        continue
    end if
end for
h' ← h
return h'
```

To address this, we propose the **Ada-Top-k** (Adaptive Top-k), a modified soft top-k mechanism. Ada-Top-k computes the k-th largest block importance score $s_{(k)}$, re-centers all block scores relative to this adaptive threshold, and applies a sigmoid activation function:

$$r_i = \frac{1}{1 + \exp(-(s_i - s_{(k)})/T)}, \quad r_i \in [0, 1] \tag{3}$$

where T is the temperature parameter controlling the sharpness of selection. Unlike softmax-based soft top-k which suffers from competitive normalization, Ada-Top-k employs a rank-based formulation that enables independent block activation—blocks with scores above $s_{(k)}$ are activated while

those below are suppressed, naturally selecting approximately k blocks without winner-take-all collapse.

In summary, the dynamic routing network in MoB generates block importance scores conditioned on input prompts, then employs Ada-Top-k to determine block execution. The parameter k controls the target number of active blocks, enabling flexible trade-offs between computational efficiency and generation quality. Algorithm 1 outlines the integration of this routing mechanism into the DiT framework, where f_n denotes the n-th Transformer block, g represents the routing network, and $h_0 \in \mathbb{R}^{B\times T\times C}$ is the initial noise input.

2.2 Block Cache

During inference with the routing network, some blocks are inevitably skipped to accelerate computation, which results in the omission of certain block-level information. To address this and preserve inference quality, MoB introduces a **block cache module**.

As illustrated in Figure 1(b), although different timesteps influence the overall trend of block-level output similarity, the outputs of certain adjacent blocks remain highly consistent. Exploiting this property, we cache the output difference δ of each block at every timestep. When a block at position i is skipped in the next timestep, the cached δ from block $i-1$ at the previous timestep is used to correct its output, as shown in Figure 4.

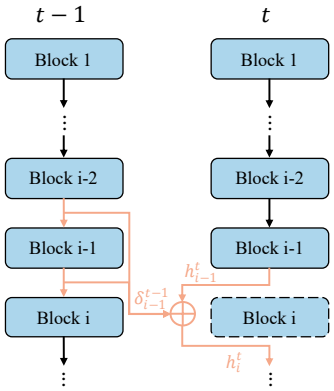


Figure 4: Workflow of the Block Cache Module.

Formally, this procedure is defined in Equations 4, where $h_i^{(t)}$ denotes the output of the i-th block at timestep t. Through this caching mechanism, MoB reduces computational cost while maintaining generation performance.

$$h_i^{(t)} \approx h_{i-1}^{(t)} + \left(h_{i-1}^{(t-1)} - h_{i-2}^{(t-1)}\right) \tag{4}$$

2.3 Other Optimizations

Timestep Skipping. In our experiments, we observe that Transformer computations at certain mid-denoising timesteps exhibit substantial redundancy. Removing all computations at these timesteps has little effect on the final generation quality, a finding consistent with prior work that accelerates inference by skipping timesteps (Zhu et al., 2025).


Figure 5: Generation Results of the Original CogVideoX and Three Timestep Skipping Strategies.

Further analysis shows that, during the mid-denoising stage, the model can maintain stable generation quality provided that multiple skips do not occur consecutively across several timesteps, as illustrated in Figure 5. In the figure, the first row shows the generation results of the original CogVideoX, while the second row corresponds to continuous timestep skipping. The third row applies skipping during the early denoising stages, and the fourth row applies intermittent skipping during the mid-denoising stages. Across all settings, the number of skipped timesteps is kept constant. The results show that intermittent skipping in the mid-denoising stages yields the best generation quality.

Motivated by this observation, we introduce an additional **timestep skipping strategy** to further reduce computational cost. Specifically, we define a fixed interval between skipped timesteps and constrain skipping to occur only within a predefined minimum and maximum timestep range, thereby restricting it to the mid-denoising stages.

Distillation. The objective of MoB is to accelerate video generation by skipping redundant computations while preserving output quality. Accordingly, the training objective is to align the performance of the MoB-augmented model with that of the original model. To reduce training data requirements

and improve training efficiency, MoB incorporates a knowledge distillation strategy based on the teacher–student paradigm.

In this framework, the original text-to-video model serves as the teacher, providing soft supervisory signals to guide the learning of the student model equipped with MoB. Specifically, the student is optimized with a distillation loss that encourages it to mimic either the final outputs or the intermediate feature representations of the teacher.

$$\mathcal{L}_{\text{distill}} = \frac{1}{B} \sum_{b=1}^{B} \| f_{\text{student}}(\boldsymbol{x_b}) - f_{\text{teacher}}(\boldsymbol{x_b}) \|_2^2 \tag{5}$$

where B is the batch size, $\boldsymbol{x_b}$ denotes the b-th input sample, and $f_{\text{student}}(\cdot)$ and $f_{\text{teacher}}(\cdot)$ represent the output logits from the student and teacher models, respectively.

Total Loss. In this study, MoB employs a distillation loss in place of the original task-specific objective. Since the Transformer backbone is kept frozen, the load balancing loss is not required. To further ensure that MoB achieves sufficient acceleration, we introduce a computation-aware reward term, denoted as $\mathcal{L}_{\text{cost}}$.

$$\mathcal{L}_{\text{cost}} = \lambda \cdot \frac{1}{Z} \sum_{i=1}^{N} p_i \cdot c_i \tag{6}$$

where λ is a hyperparameter controlling the strength of the reward, p_i denotes the activation probability of the i-th block, and c_i represents the computational cost of that block. The term $Z = \sum_{i=1}^{N} c_i$ serves as a normalization factor.

Specifically, the overall loss function is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{cost}} \tag{7}$$

where $\mathcal{L}_{\text{distill}}$ measures the discrepancy between the outputs of the student and teacher models, and $\mathcal{L}_{\text{cost}}$ represents the computation-aware cost reward.

3 Experiment

3.1 Setup

Model. We evaluate the performance of MoB on the text-to-video task using three state-of-the-art models: CogVideoX-5B, HunyuanVideo (Kong et al., 2024), and Wan2.1 (Wang et al., 2025). For comparison, we adopt MagCache (Ma et al., 2025), Sparse VideoGen (Xi et al., 2025), and FasterCache (Lv et al., 2024) as baselines. In addition, we evaluated the performance of MoB on the text-to-image task using Flux.1 (Labs, 2024).

Metrics. Following prior work (Wu et al., 2023), we evaluate both generation quality and inference efficiency using the following metrics. For video quality, we adopt two measures: *Frame Consistency*, computed by extracting CLIP embeddings for each frame and reporting the average cosine similarity across all frame pairs, and *Textual Faithfulness*, quantified by the average ImageReward score(Xu et al., 2023) between each generated frame and its corresponding text prompt. In addition, we also evaluate generation quality using PSNR and SSIM. For image quality, we adopt *Textual Faithfulness*, PSNR and SSIM.

For efficiency, we measure inference latency and report the speedup relative to the baseline model.

Datasets. In this study, for evaluation, we randomly sample 1,000 prompts from Lin et al. (2014).

Training Details. Our MoB model is trained over 20,000 epochs using 8 NVIDIA A800-SXM4-80GB GPUs, with all Transformer modules kept freezed except for the routing network. During training, the weight of the computation-aware reward λ is set to 1e-2. The weight decay is set to 1e-4 , gradient clipping is applied with a threshold of 1.0, and the learning rate is initialized to 2e-5, with a cosine learning rate scheduler applied for gradual decay.

Unlike in inference, the routing network does not discretize gating weights during training. Instead, continuous gating weights are applied to all Transformer block outputs, and their weighted aggregation is used as the final output.

Table 1: Comparison of different acceleration methods on CogVideoX, HunyuanVideo and Wan.

Method	Efficiency		Visual Quality			
	Latency (s)↓	Speedup↑	Smooth↑	Text↑	PSNR ↑	SSIM ↑
CogVideoX-5B (49 frames, 480P)						
CogVideoX-5B ($T = 50$)	227.8	1×	0.9453	0.2971	-	-
FasterCache	144.1	1.58×	0.9430	0.2939	13.91	0.572
SVG	134.5	1.69×	0.9272	0.2992	17.73	0.778
Ours	154.9	1.47×	0.9450	0.2972	16.01	0.734
HunyuanVideo (81 frames, 720P)						
HunyuanVideo ($T = 50$)	1784.2	1×	0.9686	0.3020	-	-
MagCache	706.5	2.52×	0.9681	0.3017	17.81	0.614
SVG	1055.8	1.69×	0.9655	0.2749	18.05	0.671
Ours	1427.2	1.25×	0.9672	0.2953	16.78	0.607
Wan 2.1 1.3B (81 frames, 480P)						
Wan 2.1 ($T = 50$)	103.6	1×	0.9747	0.3005	-	-
MagCache	54.8	1.89×	0.9743	0.3002	20.21	0.742
SVG	79.2	1.31×	0.9622	0.2976	18.63	0.711
Ours	71.8	1.44×	0.9702	0.2994	18.54	0.716

Baselines. As for video generation, we benchmark MoB against representative baselines, including the vanilla model, the cache-based methods MagCache and FasterCache, and the dynamic sparsity-aware model Sparse VideoGen. As for image generation, we benchmark against Flux.1-dev.

Parameters. For all baseline methods, we use their publicly released configurations to ensure fair comparison. Since different models contain varying numbers of Transformer blocks, we set 10% of the total blocks as the baseline for MoB's block-skipping strategy and fix the skip interval to 5. These settings balance inference acceleration with generation quality. All inference tasks on the same backbone are performed under a fixed random seed.

3.2 Evaluation

We assess video generation quality by comparing MoB against all baseline methods. The results are presented in Table 1.

MoB demonstrates strong performance in preserving generation quality. On the CogVideoX-5B-T2V model, it achieves a Frame Consistency score of 0.9450 and a Textual Faithfulness score of 0.2969, indicating that generation quality is effectively maintained under accelerated conditions. Representative visualizations are provided in the supplementary figures.

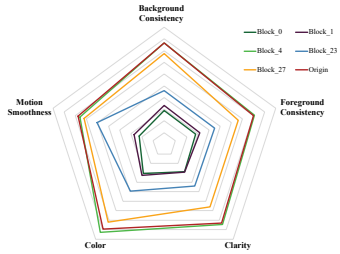


Figure 6: User study on the role of Transformer blocks in CogVideoX-5B.

To assess practical applicability, we conduct an end-to-end performance evaluation on an NVIDIA L20X GPU with CUDA 12.4, comparing MoB against all baselines in terms of latency and speedup. Results show that MoB achieves a 1.2×–1.4× acceleration over the baselines with negligible quality degradation.

For image generation quality, we report comparative results in the appendix. Quantitative results are summarized in Table F.1, and representative samples are shown in Figure F.1. MoB achieves superior generation quality — outperforming all baselines with a Text Faithfulness score of 0.3102, a PSNR of 17.64, and an SSIM of 0.7319.

Importantly, the acceleration mechanism of MoB is orthogonal to existing techniques such as attention sparsification. This suggests that MoB can serve as an independent acceleration strategy and can be further combined with methods such as Yuan et al. (2024b) or Jiang et al. (2024) to yield additional efficiency gains in video generation.

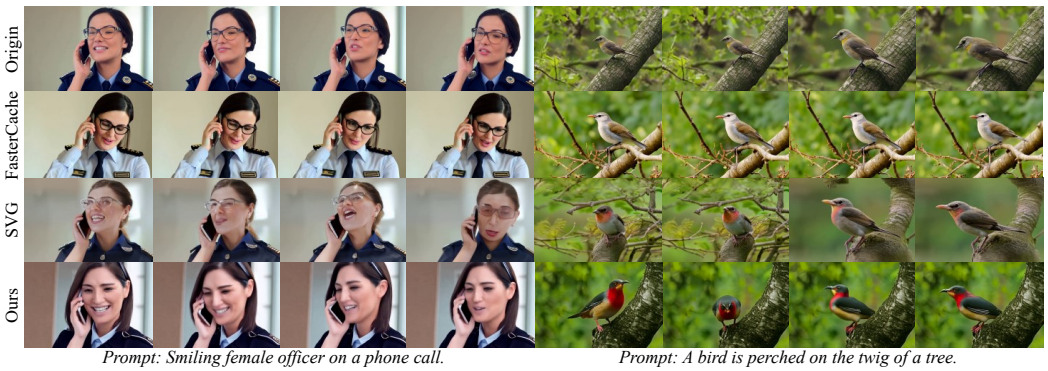


Figure 7: Comparison of Generation Results between MoB and Other Baselines on CogVideoX-5B.

3.3 User study

As discussed in previous sections, we observe that different blocks have varying impacts on the output, as measured by MSE. To further qualitatively analyze the role of each block in CogVideoX-5B and to guide the routing network for targeted dynamic routing, we conducted a user study with 100 prompts from Lin et al. (2014).

Participants evaluated videos generated by skipping individual blocks across five aspects: background quality, foreground quality, clarity, color fidelity, and motion smoothness. The results, shown in Figure 6, indicate that different blocks exhibit distinct influences on model performance, while some blocks contribute relatively little to overall generation quality.

3.4 Ablation study

Table 2: Effect of Top-k in routing network

Top-k	Latency (s)↓	Speedup↑	Smooth↑	Text↑
Origin	230.1	1×	0.9453	0.2958
40	211.3	1.08×	0.9454	0.2973
38	194.1	1.18×	0.9431	0.2967
36	179.8	1.27×	0.9329	0.2817

Effect of Top-k in routing network. To examine the performance limits of MoB's inference acceleration, we conduct an ablation study, with results summarized in Table 2. The Top-k parameter specifies the number of Transformer blocks selected to participate in the denoising process during inference, as defined in the corresponding equation.

As shown in Table 2, the Top-k setting strongly affects both inference speed and generation quality. While preserving satisfactory visual quality, MoB achieves up to 1.3× speedup. To further illustrate the qualitative impact of varying Top-k values, we present representative visual examples in the accompanying figure, which demonstrate perceptual variations across different configurations.

Table 3: Effect of the Interval Between Two Skips

Interval	Latency (s)↓	Speedup↑	Smooth↑	Text↑
Origin	230.1	1×	0.9453	0.2958
2	175.1	1.34×	0.9301	0.2714
3	195.8	1.17×	0.9316	0.2958
4	202.5	1.13×	0.9363	0.2957
5	210.3	1.09×	0.9410	0.2958

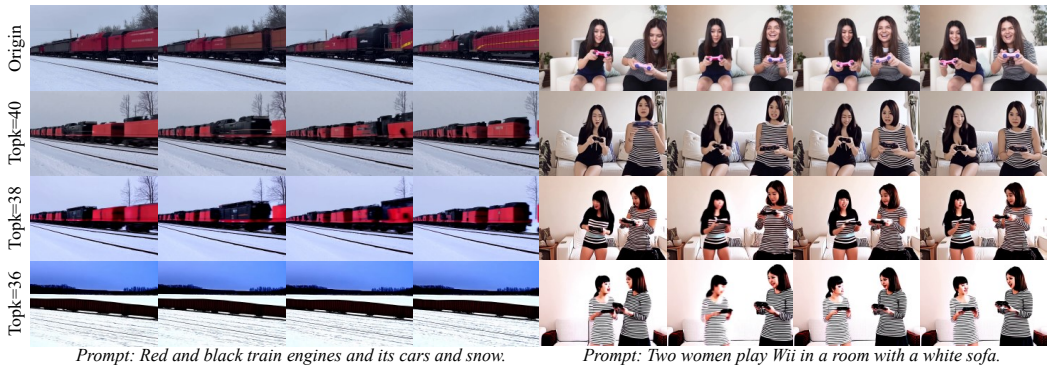


Figure 8: Effect of Top-k in routing network.

Figure 9: Effect of the Interval Between Two Skips.

Effect of the Number of Skipped Timesteps. To analyze the effect of the timestep-skipping strategy on efficiency and its impact on generation quality, we conduct an ablation study on the number of skipped timesteps. As discussed previously, when skipping occurs in the mid-denoising stage and is applied in a non-consecutive manner, the degradation in video quality is relatively small.

We define the total number of inference steps as T. In this experiment, the mid-denoising stage is empirically set to the interval $[\frac{3}{10}T, \frac{8}{10}T]$. The number of skipped timesteps is controlled by varying the interval between two skips. Experiments are conducted on CogVideoX-5B with $T = 50$, and the results are summarized in Table 3.

Dynamic Top-k Setting across Timestep. We evaluate three dynamic Top-k scheduling strategies, including the original scheme, as shown in Figure G.1, to validate the effectiveness of our proposed method.

4 Conclusion

We propose MoB, a DiT-based acceleration framework for video generation. By exploiting computational redundancy across Transformer blocks, MoB performs block-level dynamic routing to eliminate unnecessary computations. This reduces computational cost while preserving generation capability, and the framework can also be integrated with other acceleration methods.

Limitations. MoB trains the routing network based on block-level redundancy. For models with fewer Transformer blocks and stronger computational coupling, training effectiveness may be limited, and block skipping could have a larger negative impact on the final results.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Tianlong Chen, and Yu Cheng. Accelerating vision diffusion transformers with skip branches. *CoRR*, abs/2411.17616, 2024. URL https://doi.org/10.48550/arXiv.2411.17616.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

Javier Martín Daniel Verdú. Flux.1 lite: Distilling flux1.dev for efficient text-to-image generation. 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters, 2024. URL https://arxiv.org/abs/2407.11633.

Zhengcong Fei, Debang Li, Di Qiu, Changqian Yu, and Mingyuan Fan. Ingredients: Blending custom photos with video diffusion transformers. *arXiv preprint arXiv:2501.01790*, 2025.

Ruoyu Feng, Tiankai Hang, Tianyu He, Kai Qiu, Qi Dai, Jianmin Bao, Zhibo Chen, and Chong Luo. Videodit: Bridging image diffusion transformers for streamlined video generation, 2025. URL https://openreview.net/forum?id=lvgsPjRtLM.

Xuanhua He, Quande Liu, Shengju Qian, Xin Wang, Tao Hu, Ke Cao, Keyu Yan, Man Zhou, and Jie Zhang. Id-animator: Zero-shot identity-preserving human video generation. *arXiv preprint arXiv:2404.15275*, 2024.

Teng Hu, Jiangning Zhang, Ran Yi, Hongrui Huang, Yabiao Wang, and Lizhuang Ma. SaRA: High-efficient diffusion model fine-tuning with progressive sparse low-rank adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wGVOxplEbf.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention, 2024. URL https://arxiv.org/abs/2407.02490.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng YAN. Moh: Multi-head attention as mixture-of-head attention, 2025a. URL https://openreview.net/forum?id=VOVFvaxgD0.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. Moh: Multi-head attention as mixture-of-head attention, 2025b. URL https://arxiv.org/abs/2410.11842.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Daquan Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models. *CoRR*, abs/2412.03603, 2024. URL https://doi.org/10.48550/arXiv.2412.03603.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers, 2025. URL https://arxiv.org/abs/2503.06923.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps, 2022. URL https://arxiv.org/abs/2206.00927.

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *CoRR*, abs/2410.19355, 2024. URL https://doi.org/10.48550/arXiv.2410.19355.

Zehong Ma, Longhui Wei, Feng Wang, Shiliang Zhang, and Qi Tian. Magcache: Fast video generation with magnitude-aware cache, 2025. URL https://arxiv.org/abs/2506.09045.

Minglei Shi, Ziyang Yuan, Haotian Yang, Xintao Wang, Mingwu Zheng, Xin Tao, Wenliang Zhao, Wenzhao Zheng, Jie Zhou, Jiwen Lu, Pengfei Wan, Di Zhang, and Kun Gai. Diffmoe: Dynamic token selection for scalable diffusion transformers, 2025. URL https://arxiv.org/abs/2503.14487.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. URL https://arxiv.org/abs/2209.14792.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Zhao Jin, Jingyi Liao, Shunyu Liu, and Dacheng Tao. Vorta: Efficient video diffusion via routing sparse attention, 2025. URL https://arxiv.org/abs/2505.18809.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *CoRR*, abs/2503.20314, March 2025. URL https://doi.org/10.48550/arXiv.2503.20314.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, pp. 7589–7599, 2023. URL https://doi.org/10.1109/ICCV51070.2023.00701.

Yiming Wu, Huan Wang, Zhenghao Chen, and Dong Xu. Individual content and motion dynamics preserved pruning for video diffusion models, Nov 2024.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity, 2025. URL https://arxiv.org/abs/2502.01776.

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. In *NeurIPS*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025. URL https://arxiv.org/abs/2408.06072.

Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space, 2023. URL https://arxiv.org/abs/2302.07685.

Shenghai Yuan, Jinfa Huang, Xianyi He, Yunyuan Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. Identity-preserving text-to-video generation by frequency decomposition. *arXiv preprint arXiv:2411.17440*, 2024a.

Zhihang Yuan, Hanling Zhang, Pu Lu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattn: Attention compression for diffusion transformer models, 2024b. URL https://arxiv.org/abs/2406.08552.

Yuanhao Zhai, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, David Doermann, Junsong Yuan, and Lijuan Wang. Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation, 2024. URL https://arxiv.org/abs/2406.06890.

Yuechen Zhang, Yaoyang Liu, Bin Xia, Bohao Peng, Zexin Yan, Eric Lo, and Jiaya Jia. Magic mirror: Id-preserved video generation in video diffusion transformers. *arXiv preprint arXiv:2501.03931*, 2025.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=taHwqSrbrb.

Shangwen Zhu, Han Zhang, Zhantao Yang, Qianyu Peng, Zhao Pu, Huangji Wang, and Fan Cheng. Accelerating diffusion sampling via exploiting local transition coherence, 2025. URL https://arxiv.org/abs/2503.09675.

A ADA-TOP-k ROUTING STRATEGY

As shown in Figure A.1, the weight distributions of three routing strategies are visualized across the 42 Transformer blocks of CogVideoX-5B under the same input. The results show that, while maintaining differentiability, the Ada-Top-k strategy enables better adaptation of the routing network for acceleration. From a mathematical perspective, the Ada-Top-k, Soft-Top-k, and Top-k routing strategies differ primarily in their treatment of normalization and scaling factors. However, Ada-Top-k is the only strategy among the three that is both differentiable and stably yields a k-hot output. During training, given the specific requirements of our block-skipping routing network—particularly its distinction from conventional Mixture-of-Experts (MoE) architectures—we impose two key criteria on the routing mechanism: (1) the routing computation must be differentiable to enable end-to-end training; and (2) the routing output must be k-hot to closely approximate the inference-time scenario in which only a subset of blocks is activated. Given these constraints, we deliberately excluded Top-k (which is non-differentiable) and Soft-Top-k (which yields soft, often near-one-hot distributions rather than exact k-hot selections). Ada-Top-k uniquely satisfies both conditions, making it the most suitable choice for our framework.

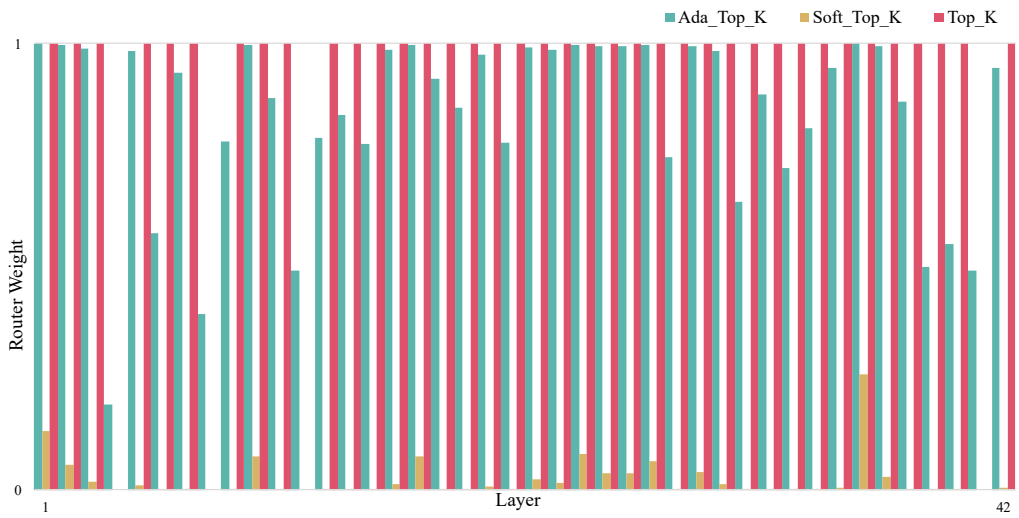


Figure A.1: Output Weight Distributions of Ada-Top-k vs. Other Routing Strategies.

B INFERENCE LATENCY AND MEMORY FOOTPRINT OF MOB

In terms of time overhead, using CogVideoX-5B with 42 Transformer blocks as an example, the additional time overhead of the routing network accounts for less than 3% of the original model's inference time. Furthermore, each block skipped by the routing network results in an approximate 2% acceleration.

As for space overhead, the MoB routing network is designed to be lightweight, introducing a negligible number of parameters relative to the original model. During inference, the model caches only the intermediate outputs from the previous timestep, discarding earlier ones to mitigate excessive memory consumption. Consequently, in terms of inference memory overhead, the MoB strategy increases GPU memory consumption by approximately 1%. This efficient resource utilization ensures that the benefits of improved routing are achieved without imposing significant additional memory overhead.

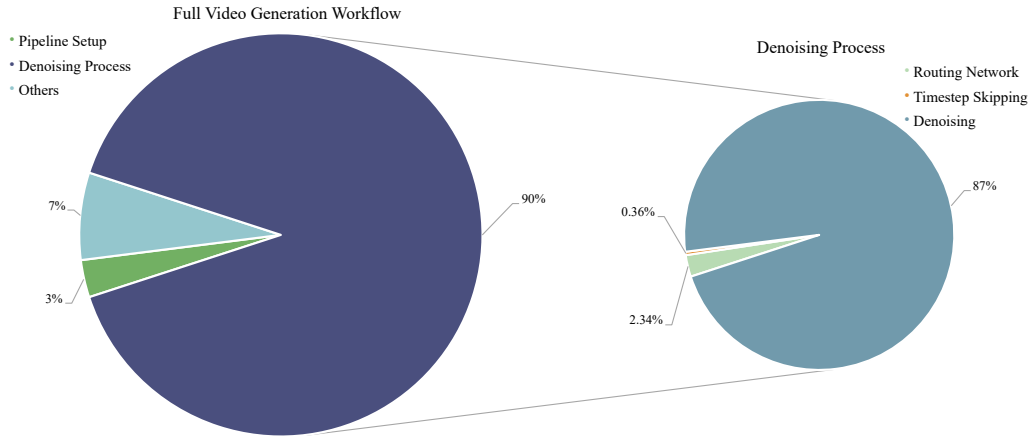


Figure B.1: Time budget allocation across the full inference pipeline with MoB.

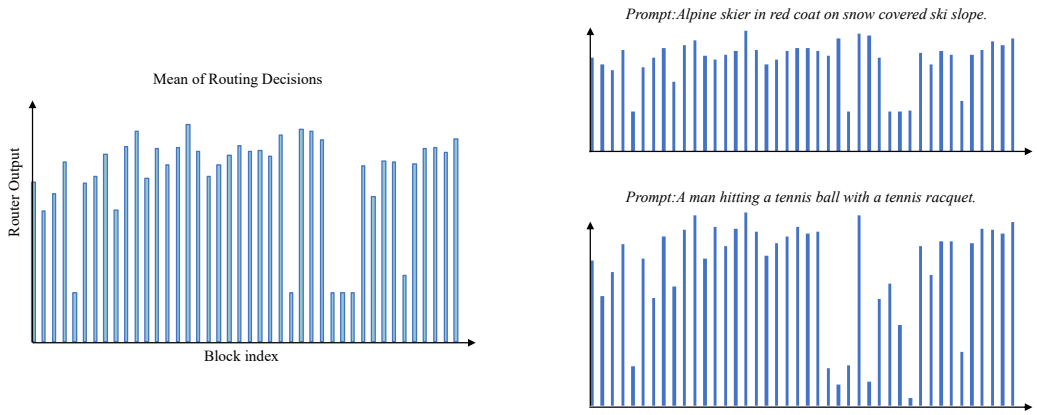


Figure C.1: Visualization of routing decisions.

C Visualization of routing decisions

D Generating long-duration videos using CogVideoX-5B w/o MoB

E Statistical Analysis of Block-Wise Contributions in CogVideoX-5B

To analyze the functional contribution of each block, we manually skip individual blocks in CogVideoX when generating videos. Representative cases are shown in Figure E.1.

F Transfer of MoB to text-to-image generation on Flux.1-dev

We conducted the experiment shown in the Figure F.1 on Flux.1-dev, where the baseline Flux uses 10 denoising steps and contains 19 Transformer blocks. In the MoB-augmented variant, we configure the router to skip 2 blocks, resulting in only 17 active blocks during inference. We further conducted a quantitative evaluation of MoB on Flux.1-dev, comparing its performance with several representative baselines, as shown in Table F.1.

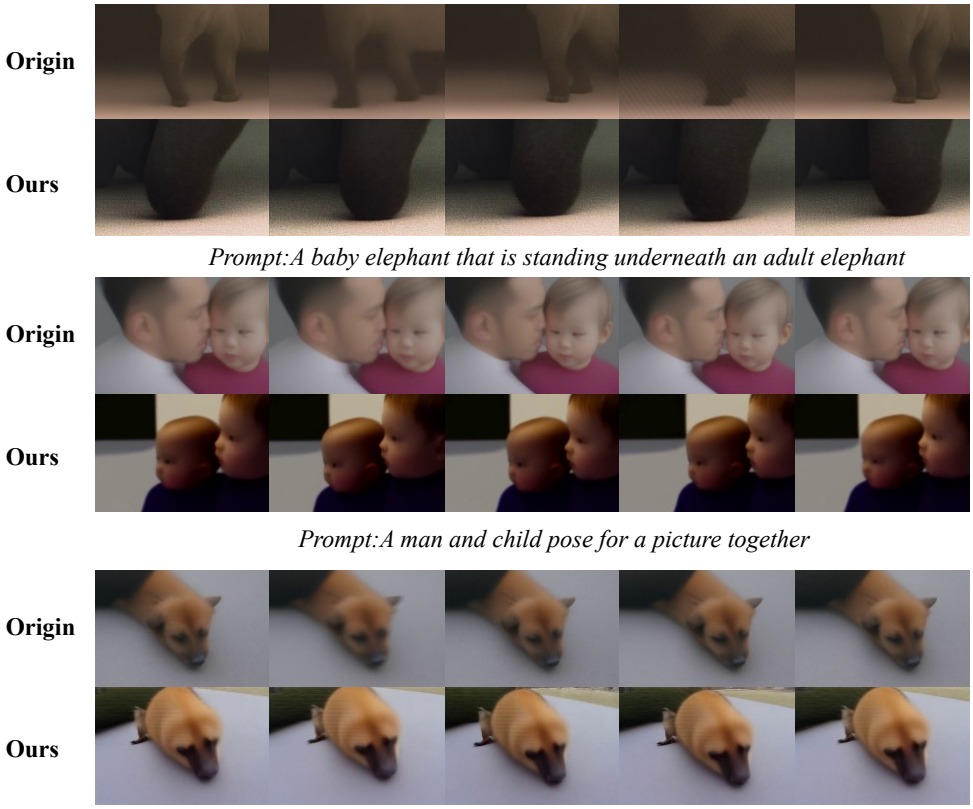


Figure D.1: Generating long-duration videos using CogVideoX-5B w/o MoB.

Table F.1: Evaluation of MoB and Baseline Methods on Flux.1-dev for T2I Acceleration.

Method	Efficiency		Visual Quality		
	Latency (ms)↓	Speedup↑	Text↑	PSNR ↑	SSIM ↑
Flux.1-dev (1024P)					
Flux.1-dev ($T = 28$)	582	1×	**0.3108**	-	-
MagCache	277	**2.10**×	0.3087	17.39	0.6952
TaylorSeer(Liu et al., 2025)	362	1.61×	0.2831	13.98	0.6179
Ours	362	1.65×	0.3102	**17.64**	**0.7319**

G Dynamic Top-k Setting Across Timesteps

In this section, we investigate whether dynamically varying the Top-k parameter across timesteps can improve generation quality. Figure G.1 compares three routing strategies.

Random-k refers to performing dynamic routing at each timestep by feeding slightly perturbed inputs into the routing network to induce variation in its outputs. This experiment uses a fixed setting of $k = 40$, as lower Top-k values substantially degrade temporal consistency, making it difficult to assess the effectiveness of the strategy.

Two-stage-k denotes a strategy in which different Top-k settings are applied across timesteps: during the first half of the denoising process, only 50% of the target number of blocks are skipped, whereas in the second half the full skip ratio is applied. This setup allows us to examine whether delaying aggressive skipping yields better results.

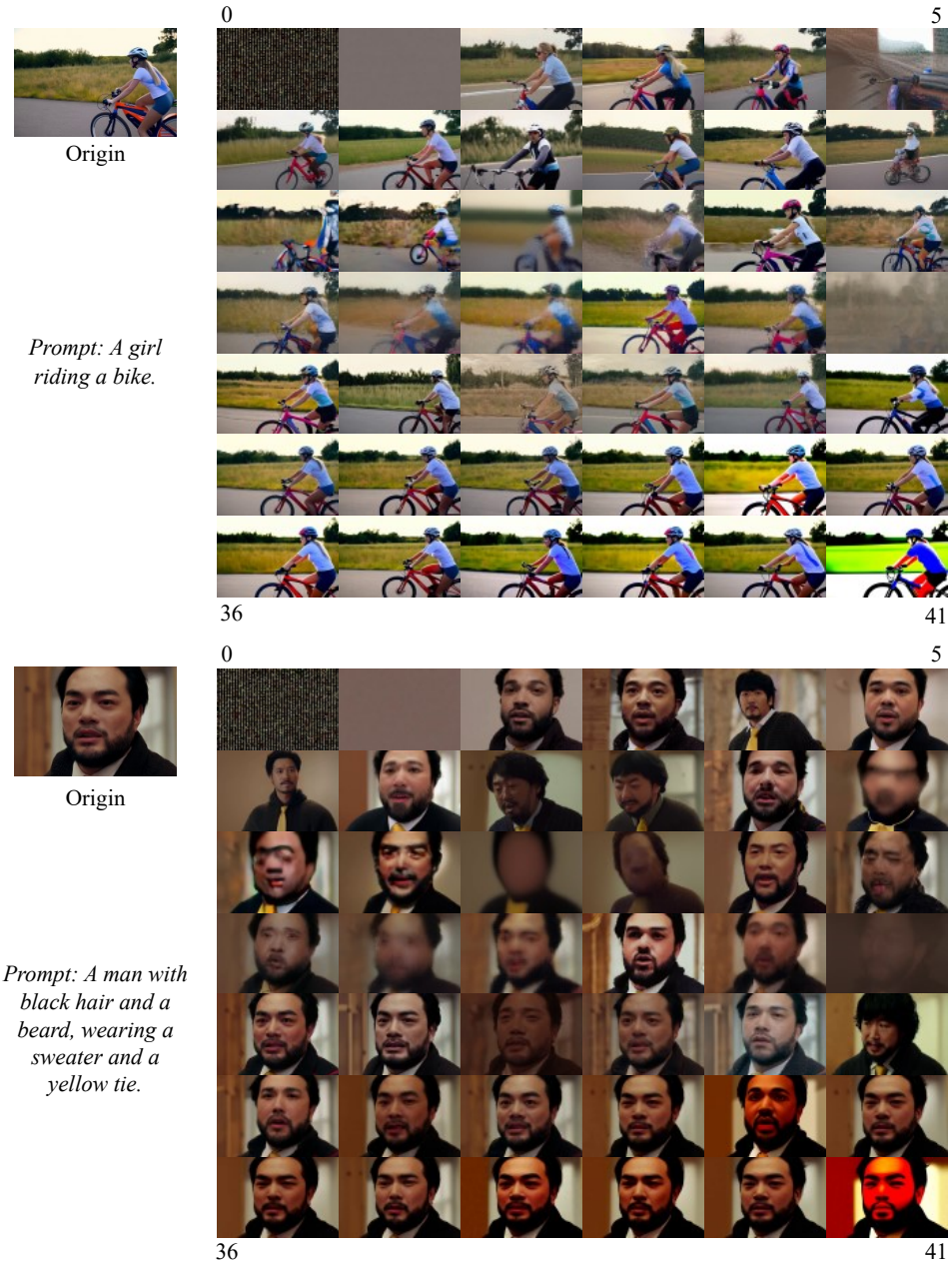


Figure E.1: Visualization of Transformer block outputs across different timesteps.

Ours corresponds to the full-pipeline skipping strategy proposed in the main paper, using a fixed $k = 38$ across all timesteps. For a fair comparison, both Two-stage-k and Ours share the same target Top-k configuration.

The results show that skipping different blocks across timesteps leads to noticeable degradation in video consistency. Although the Two-stage-k strategy does not yield significantly worse results, it reduces the total number of block-skipping operations by approximately one quarter, thereby weakening its acceleration effect. As a result, we do not adopt this strategy in our final design.

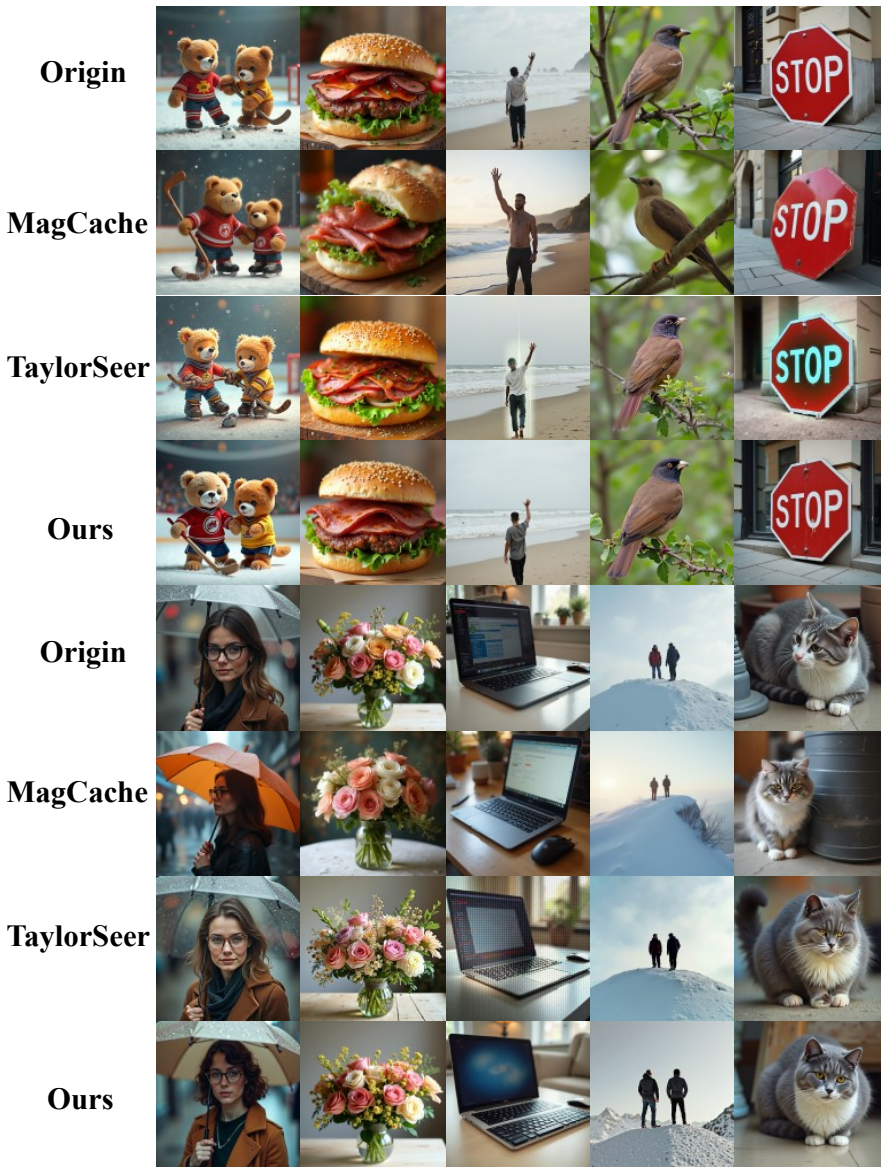


Figure F.1: Comparison of MoB and Baseline Methods on Flux.1-dev.

H Evaluation of MoB on HunyuanVideo and Wan2.1

I Related Work

I.1 Diffusion Models for Video Generation

Compared to earlier approaches such as Generative Adversarial Networks (GANs), diffusion models have exhibited superior capabilities in high-quality video generation tasks. Early video diffusion models enhanced pre-trained image diffusion models by incorporating temporal convolution layers and temporal attention mechanisms to improve interframe consistency. For instance, Singer et al. (2022) extended image diffusion models by introducing temporal modeling, leveraging learned coherence along the time axis to adapt text-to-image generation into text-to-video synthesis. Chen et al. (2023) further advanced the field by scaling up the dataset used for video diffusion models, enabling video generation conditioned on various inputs, including text, images, and motion-adaptive signals.

Figure G.1: Comparison of Different Top-k Router Policies.

However, these methods remain constrained by the scalability limitations of the U-Net framework, which makes them less effective in generating longer-duration videos.

In contrast, Diffusion Transformer (DiT)-based models, such as Yu et al. (2023) and Feng et al. (2025), are better suited for capturing complex spatio-temporal dependencies and can generate videos lasting ten seconds or longer. Brooks et al. (2024) further extended DiT for joint spatio-temporal modeling and leveraged large-scale pre-training, achieving the generation of high-quality videos exceeding one minute in length. These advances highlight the advantages of DiT-based models over traditional U-Net-based architectures, particularly in terms of video length and quality.

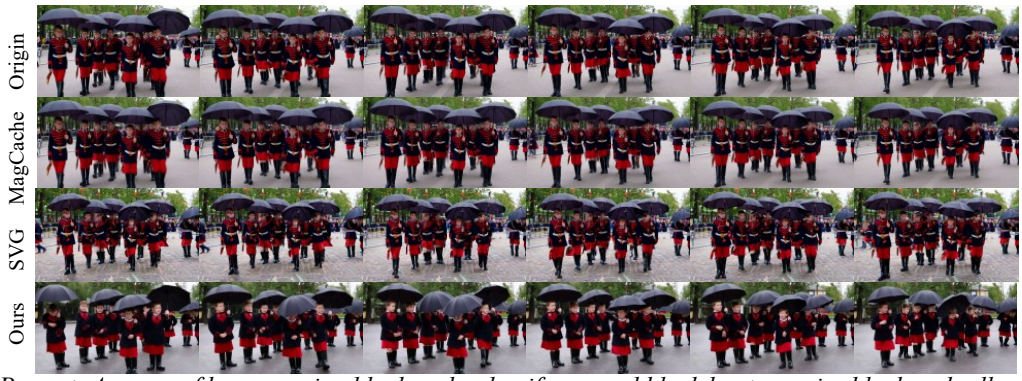
I.2 Mixture of Expert

In recent years, Transformer-based large language models (LLMs) have exhibited remarkable capabilities, largely driven by their extensive scale, vast training datasets, and substantial computational resources. However, the significant time and financial costs associated with training such models have motivated researchers to explore more efficient strategies to ensure their sustainable development.

The Mixture of Experts (MoE) paradigm was initially introduced in Jacobs et al. (1991) and Jordan & Jacobs (1994). The advent of sparsely gated MoE and its integration into large Transformer-based LLMs have demonstrated its effectiveness in reducing both computational time and memory consumption.

Building upon this idea, Jin et al. (2025b) incorporate this mechanism into the attention module, where each attention head was treated as an expert, and a routing mechanism was employed to dynamically select the Top-k heads for computing weighted attention outputs. Inspired by this, we observe that different DiT blocks contribute unequally to the final generation process. Based on this observation, we propose treating each DiT block as an expert and introducing a routing network that dynamically selects the Top-k blocks for participation in generation, thereby optimizing both training and inference efficiency.

Prompt: A dog enjoying playing on the beach in the sun.



Prompt: A group of boys wearing black and red uniforms and black boots carries black umbrellas.

Prompt: A person in the air on a snowboard.

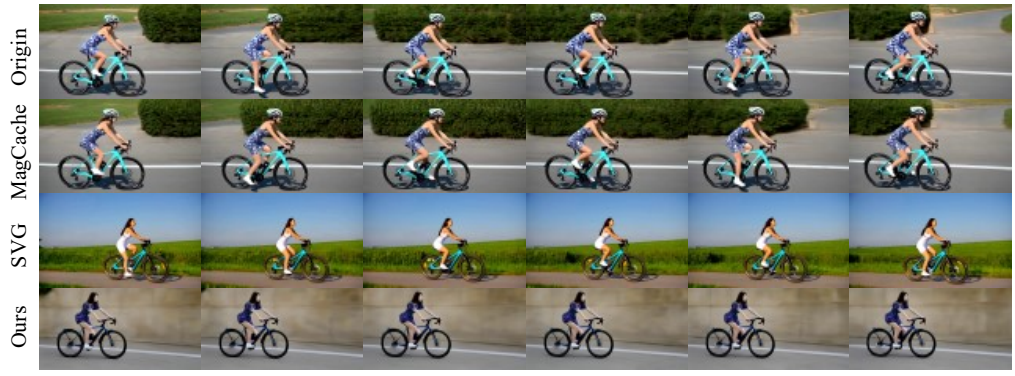
Prompt: Cows graze peacefully in a field with a dormant volcano in the distance.

Figure H.1: Results of MoB and Other Acceleration Methods on HunyuanVideo.

Prompt: 2 zebras standing next to each other in plaines.

Prompt: The desert is made and ready to be eaten.

Prompt: Two cats sleeping with a remote control near each of them.



Prompt: The woman is riding a bike in a dress.

Figure H.2: Results of MoB and Other Acceleration Methods on Wan2.1.