Breaking the Uniformity Trap: Scaling Video Diffusion Model via SplitMoE

Yu Xu[1,2]*, **Yuxin Zhang**[2]†, **Xiao Yang**[3], **Haotian Yang**[2]
Yizhi Wang[2], **Xinwei Huang**[2], **Minxuan Lin**[2]
Angtian Wang[2], **Chongyang Ma**[2], **Fan Tang**[4]
[1]University of Chinese Academy of Sciences, [2]ByteDance
[3]Canva Research
[4]University of Science and Technology Beijing
†Corresponding author

Abstract

Mixture-of-Experts (MoE), popularized by large language models, is a promising paradigm for scaling visual generative models. However, conventional token-wise MoE routes tokens independently within a homogeneous expert pool and regularizes expert usage toward uniformity, making it poorly matched to video data that is spatiotemporally redundant and semantically long-tailed. We show that existing visual MoEs fall into a uniformity trap: semantically under-organized routing, compounded by uniform expert-usage regularization, scatters coherent patches across disparate experts, causing routing fragmentation and structural distortion. To address this, we propose SplitMoE, a split-role sparse architecture that breaks the shackles of uniformity. To accommodate the inherent semantic imbalance, we explicitly bifurcate the expert pool into semantic experts and generic experts, with semantic experts capturing high-level semantic abstraction and generic experts preserving residual visual information and flexible generative capacity. Leveraging prototype-guided routing and pull-push regularization, SplitMoE enables tokens to cluster naturally by semantic attributes rather than arbitrary balancing constraints. Extensive results show that under an equivalent activated-parameter budget, SplitMoE outperforms traditional load-balanced MoEs in convergence speed, routing coherence, and video generation quality across standard benchmarks. By revealing an emergent coarse-to-fine denoising logic, SplitMoE provides the community with a modality-aware scaling path, serving as a critical reference for building large-scale video world models.

1 Introduction

Diffusion Transformers (DiTs) [24] have made significant advances in high-fidelity video generation [7, 8, 14, 20], yet scaling their parameters to capture complex motion remains computationally prohibitive. Mixture-of-Experts (MoE) provides a promising path toward scalable modeling by decoupling total model capacity from active FLOPs. However, standard MoE architectures are developed for language models [3, 4, 15]; nevertheless, current visual generative models [5, 31, 44] often naively adopt these language-centric designs, taking their cross-modal effectiveness for granted and assuming that mechanisms optimized for discrete syntax will naturally generalize to the highly redundant and heterogeneous spatiotemporal tokens of video.

In this study, we revisit one of the fundamental settings of standard MoE: load-balancing mechanisms which enforce tokens' statistical uniformity across available experts when training. We argue that such

*Work done during an internship at ByteDance.

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

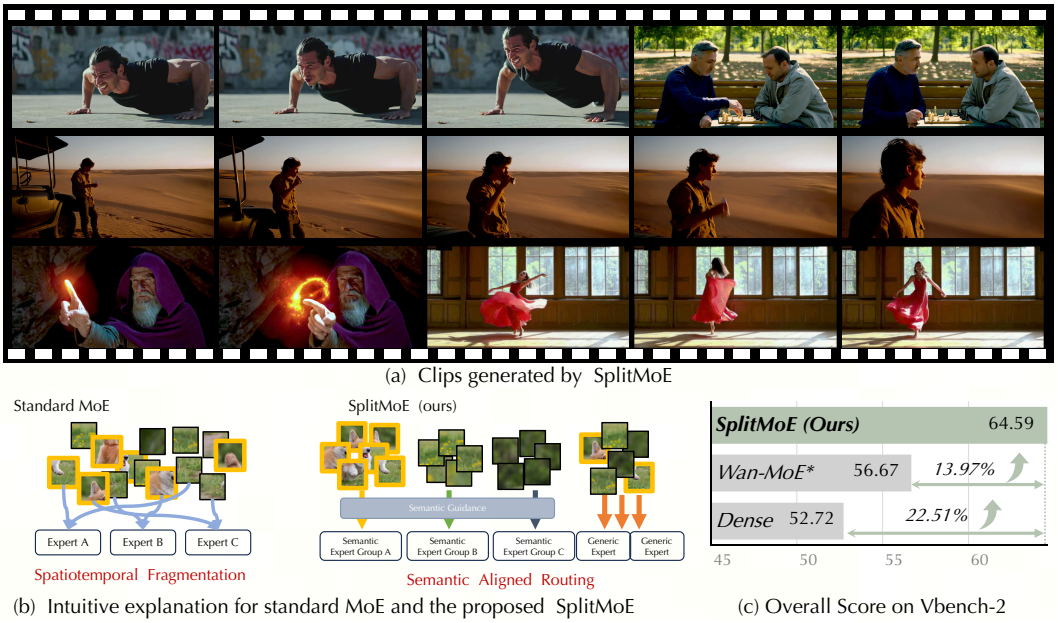
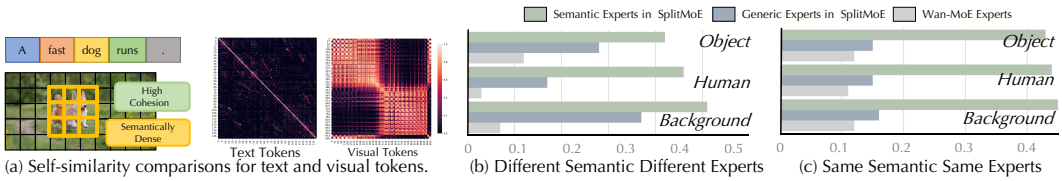


Figure 1: SplitMoE scales video diffusion models by introducing semantic-aligned expert partitioning, reducing spatiotemporal fragmentation, and generating higher-quality videos than dense fine-tuning and WAN with a standard MoE.

unique distributional characteristics are poorly aligned with the unstructured and imbalanced nature of video data. While language tokens are often organized into distinguishable semantic units by discrete syntax, video tokens are highly imbalanced across different regions. Unconstrained sparse routing in video diffusion features can therefore produce *spatiotemporal fragmentation*: neighboring patches from the same coherent object may be dispatched to different experts, while visually redundant background patches may dominate multiple experts (as illustrated in Fig. 1(b)).

To investigate such a "uniformity trap", we compare the self-similarity patterns of language and visual tokens in Fig. 2(a). Unlike language tokens exhibiting relatively high entropy and low cohesion, visual tokens are densely correlated and spatially continuous. Directly routing such tokens leads the experts to redundantly specialize in ubiquitous visual patterns rather than distinct semantic entities. We further group video tokens with off-the-shelf semantic segmentation masks and analyze their routed experts in Fig. 2. Two complementary properties are measured: semantic distinctiveness (Fig. 2(b)) evaluates whether different semantic classes are assigned to more category-specific experts, and intra-sample routing dominance (Fig. 2(c)) evaluates whether tokens from the same semantic region are consistently routed to the same expert. High values in both metrics indicate routing that is discriminative across semantic classes and coherent within each semantic region. Standard MoE has consistently low semantic distinctiveness and routing dominance across most semantic classes, indicating that tokens from the same region are often dispersed across experts. Enforcing strict uniform allocation may therefore amplify spatiotemporal fragmentation, since tokens within semantically coherent regions are forced into uniform routing patterns.

Motivated by these observations, we propose SplitMoE, a structured routing framework that separates semantic abstraction from general visual residual modeling. As shown in Fig. 1(b), SplitMoE partitions experts into a semantic branch and a generic branch. The semantic branch is guided to route tokens from the same semantic region to consistent expert groups and to assign different semantic regions to more distinguishable expert groups, thereby improving semantic coherence while avoiding expert collapse. This effect is supported by the green bars in Fig. 2: compared with standard MoE, our semantic branch achieves higher routing dominance across all semantic classes and stronger semantic distinctiveness for most categories. This design addresses the fragmentation observed in common video MoE routing, where semantically similar regions may be split across unrelated experts. As shown in Fig. 1(c), semantically coherent token assignment encourages experts to develop more specialized capabilities, leading to better generation performance. Meanwhile, the generic



(a) Self-similarity comparisons for text and visual tokens. (b) Different Semantic Different Experts (c) Same Semantic Same Experts

Figure 2: Motivation for semantic-aware Video-MoE routing. Visual tokens exhibit much stronger cohesion than text tokens, making uniform MoE routing prone to fragmented assignments. Compared with standard MoE, SplitMoE routes tokens from different semantic regions to more distinguishable experts while assigning tokens within the same semantic region to more consistent expert groups.

branch retains flexible capacity for residual variations that are not well captured by discrete semantic grouping. In this way, SplitMoE avoids forcing all visual tokens into a single uniform routing space. To ensure meaningful specialization and prevent semantic collapse, we introduce a Prototype-Guided Semantic Routing mechanism. We use prototypes as a bridge between visual model features and DiT visual tokens, guiding the router toward semantic-aware expert assignment.

More importantly, we demonstrate that this role-aware decoupling aligns with the semantic-to-detail generation order of the diffusion process [42] through Emergent Temporal Specialization. Without relying on any explicit timestep-conditioned routing constraints, SplitMoE naturally routes more capacity to Semantic Experts during the early, high-noise structural phase, and shifts more expert weights to Generic Experts during the final denoising stages. This emergent behavior validates the intuition behind our semantic-generic partition and establishes an effective paradigm for modeling heterogeneous video tokens. In summary, our core contributions are threefold:

- We propose SplitMoE, a sparse architecture tailored for video generation. By decoupling visual modeling into Semantic and Generic Experts, SplitMoE addresses the inherent cohesion of video tokens and mitigates spatiotemporal fragmentation and temporal flickering.
- We introduce Prototype-Guided Semantic Routing, which uses learnable prototypes to bridge reconstructive visual features and DiT visual tokens. It enables semantic-aware expert assignment, reduces expert homogenization, and promotes meaningful specialization while preserving flexible residual modeling.
- Under matched activated-parameter budgets, **SplitMoE** outperforms standard MoE and remains competitive with other baselines of comparable activated parameter counts. We provide empirical insights into Video-MoE routing dynamics and timestep analyses, which reveal a coarse-to-fine routing pattern consistent with denoising.

2 Related Work

Large-scale text-to-video generation. Text-to-video generation has advanced rapidly with the scaling of diffusion models and transformer architectures. Closed-source systems [8, 14, 18, 19, 21, 22, 26, 27, 30] have set high standards in visual fidelity, physical plausibility, and long-video generation. Meanwhile, open-source models have shifted from 3D U-Nets [1] to Diffusion Transformers [24], with recent systems [11, 13, 25, 34, 35] combining DiT backbones, advanced autoencoders, and multimodal language models to approach proprietary performance. Despite this progress, open-source models still suffer from spatio-temporal semantic inconsistency, structural distortion, and temporal jitter, motivating more effective semantic representation and token-level control during generation.

Mixture of experts for visual generation. MoE effectively scales LLMs [3, 4, 15] by offering large capacity with manageable inference cost through sparse routing. This paradigm has extended to visual generation, where recent works [2, 5, 31, 40, 44] show that sparse architectures improve diffusion transformers for image synthesis. For video generation, existing methods [35] adopt coarse timestep-level MoE routing, assigning one expert to all tokens at each denoising stage. While effective, this overlooks the spatio-temporal heterogeneity within video frames. Token-level routing can address this limitation but introduces another challenge: conventional load-balancing objectives [6] uniformly scatter tokens across experts, disrupting strong video semantic correlations and temporal consistency.

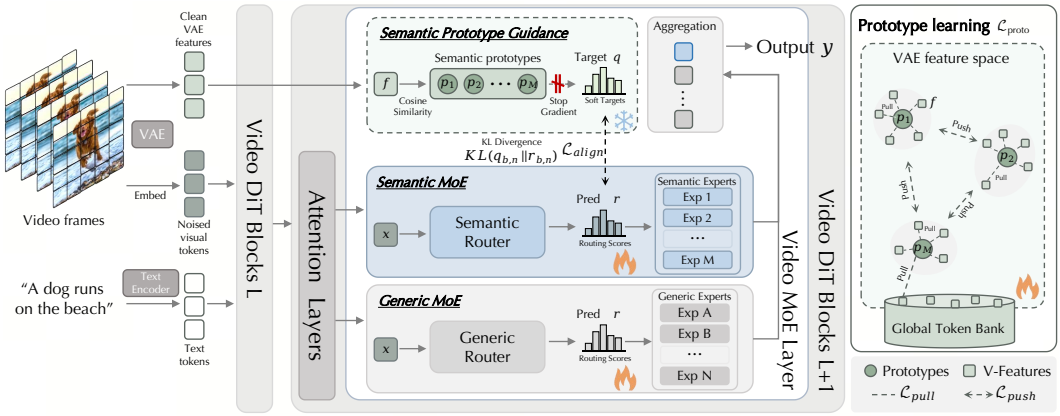


Figure 3: Pipeline of SplitMoE. SplitMoE splits each Video MoE layer into Semantic and Generic MoE branches, where VAE-prototype induced soft targets guide semantic routing while the generic branch preserves flexible residual modeling capacity. Learnable prototypes are optimized in the VAE feature space with pull-push regularization and a global token bank, encouraging semantic experts to form diverse and well-covered visual concepts.

Recent works have addressed homogeneous routing in static image generation. ProMoE [38] uses prototype-guided routing to segregate conditional and unconditional tokens, while Mammoth-Moda2 [29] decouples generation and understanding experts in a unified autoregressive-diffusion framework. However, they focus on conditioning types or task modalities rather than video structure. In contrast, SplitMoE targets the long-tailed spatio-temporal redundancy of video by splitting experts into semantic and generic pools, decoupling high-level structural abstraction from localized high-frequency reconstruction.

3 Method

3.1 Preliminaries

A standard Mixture-of-Experts (MoE) layer [28] replaces a dense feed-forward network with a set of experts $\mathcal{E} = \{E_1, \ldots, E_M\}$ and a router that activates only a small subset of experts for each token:

$$\mathbf{y} = \sum_{i \in \mathcal{I}(\mathbf{x})} g_i(\mathbf{x}) E_i(\mathbf{x}), \qquad |\mathcal{I}(\mathbf{x})| = K \ll M, \tag{1}$$

where $\mathcal{I}(\mathbf{x})$ denotes the set of selected experts and $g_i(\mathbf{x})$ represents the corresponding routing weight. Most existing MoE architectures originate from language modeling [4], where load-balancing objectives are commonly employed. Specifically, an auxiliary loss is typically added during training to penalize routing skew, forcing a uniform distribution of tokens across all available experts to prevent expert collapse and maximize capacity utilization.

3.2 SplitMoE: Split-Role Visual Experts

We propose SplitMoE, which explicitly decouples the MoE layer into specialized functional roles. By utilizing continuous VAE features as a teacher signal to anchor tokens to learnable semantic prototypes, our approach separates high-level semantic abstraction from low-level generic reconstruction.

Decoupled expert partitioning. We partition the standard expert pool $\mathcal{E}$ into two disjoint subsets, specifically the semantic experts $\mathcal{E}^{\text{sem}}$ and the generic experts $\mathcal{E}^{\text{gen}}$, formulated as:

$$\mathcal{E} = \mathcal{E}^{\text{sem}} \cup \mathcal{E}^{\text{gen}}, \qquad \mathcal{E}^{\text{sem}} \cap \mathcal{E}^{\text{gen}} = \emptyset, \qquad M = M_s + M_g. \tag{2}$$

Similar to an artist sketching a structural outline before rendering generic visual elements, semantic experts model high-level structures, while generic experts capture local appearance and generative residuals. This bifurcation alleviates capacity interference from forcing a single isotropic expert pool to jointly optimize low-frequency semantics and high-frequency generic features.

Given a patchified video token $\mathbf{x} \in \mathbb{R}^C$, the router initially projects the token to compute the semantic-generic affinities:

$$\mathbf{h} = \phi(\mathbf{W}_1\mathbf{x}), \qquad z_e = \alpha \left\langle \frac{\mathbf{h}}{\|\mathbf{h}\|_2}, \frac{\mathbf{w}_e}{\|\mathbf{w}_e\|_2} \right\rangle, \qquad s_e = \sigma(z_e), \tag{3}$$

where ϕ denotes the GELU activation function and α represents a learnable, clipped scale that controls the sharpness of the sigmoid affinities. Unlike a global softmax operation that forces competition among all experts, the utilization of independent sigmoid scores s_e allows a single token to be highly compatible with both a semantic expert and a generic expert simultaneously.

The logits and scores are divided into the respective semantic and generic groups, enabling the independent selection of the top K_s and K_g experts:

$$\mathcal{I}^{\text{sem}} = \text{TopK}(\tilde{\mathbf{s}}^{\text{sem}}, K_s), \qquad \mathcal{I}^{\text{gen}} = \text{TopK}(\tilde{\mathbf{s}}^{\text{gen}}, K_g), \qquad \mathcal{I} = \mathcal{I}^{\text{sem}} \cup \mathcal{I}^{\text{gen}}. \tag{4}$$

Here, $\tilde{\mathbf{s}}$ represents the biased routing scores (detailed in Sec. 3.3), which are utilized solely for the discrete expert selection step and may incorporate exploration noise or non-gradient balancing biases. However, to maintain routing fidelity, the final mixture weights are consistently computed in a dynamic manner from the clean raw scores:

$$g_e^{\text{sem}} = \frac{s_e}{\sum_{j \in \mathcal{I}^{\text{sem}}} s_j + \epsilon}, \qquad g_e^{\text{gen}} = \frac{s_e}{\sum_{j \in \mathcal{I}^{\text{gen}}} s_j + \epsilon}. \tag{5}$$

The final MoE output aggregates the specialized contributions:

$$\mathbf{y} = \sum_{e \in \mathcal{I}^{\text{sem}}} g_e^{\text{sem}} E_e(\mathbf{x}) + \sum_{e \in \mathcal{I}^{\text{gen}}} g_e^{\text{gen}} E_e(\mathbf{x}). \tag{6}$$

Crucially, a fixed active expert budget is maintained by ensuring that $K = K_s + K_g$ equals the Top-K value of the standard MoE baseline (*e.g.*, $K_s = 1$ and $K_g = 1$ for a Top-2 routing setup), ensuring the performance gains are not driven by an inflated computational budget, but rather by role-aware capacity allocation.

3.3 Prototype-Guided Semantic Routing

We introduce a set of learnable prototypes $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{M_s}\}$ that act as a semantic bottleneck between continuous VAE features and discrete semantic experts. Inspired by SRA2 [36], we use clean VAE features as a cost-free guidance space for the rich visual priors without relying on external representation models. Each prototype $\mathbf{p}_m$ serves as a semantic anchor for one semantic expert in the VAE feature space. To obtain a stable prototype topology, we decouple router learning from prototype optimization through two objectives.

Router alignment ($\mathcal{L}_{\text{align}}$). For the i-th visual token, we compute a teacher assignment $\mathbf{q}_i$ by comparing its corresponding clean VAE feature $\mathbf{f}_i$ with the semantic prototypes using temperature-scaled cosine similarity. The resulting $\mathbf{q}_i$ is treated as a fixed target. Meanwhile, the router takes the DiT visual token $\mathbf{x}_i$ as input and predicts a semantic routing distribution $\mathbf{r}_i$ from the logits. We align this router prediction with the VAE-prototype induced soft target using Kullback–Leibler divergence:

$$\mathcal{L}_{\text{align}} = \frac{1}{N} \sum_{i=1}^{N} \text{KL}(\mathbf{q}_i \,\|\, \mathbf{r}_i). \tag{7}$$

This objective trains the router to assign visually similar tokens to consistent semantic prototypes, while the stop-gradient operation prevents the router objective from directly distorting the prototype topology.

Prototype optimization ($\mathcal{L}_{\text{proto}}$). The prototypes are explicitly optimized to track the underlying visual feature manifold. We formulate this process by treating prototypes as particles on a hypersphere governed by two complementary forces, attraction (pull) and repulsion (push) [37]:

$$\mathcal{L}_{\text{proto}} = \underbrace{-\frac{1}{N}\sum_{i=1}^{N} \tau_c \log \sum_{m=1}^{M_s} e^{\frac{\cos(\mathbf{f}_i, \mathbf{p}_m)}{\tau_c}} + \frac{1}{M_s}\sum_{m=1}^{M_s}\left(1 - \max_{\bar{\mathbf{f}} \in \mathcal{B}} \cos(\mathbf{p}_m, \bar{\mathbf{f}})\right)}_{\text{Pull: keep prototypes on current and historical VAE features}} + \underbrace{\frac{1}{M_s(M_s-1)} \sum_{j \neq k} \max(0, \cos(\mathbf{p}_j, \mathbf{p}_k) - \delta)}_{\text{Push: separate prototypes to avoid redundancy}} \tag{8}$$

Specifically, the **pull** term attracts prototypes toward both current mini-batch features and recent historical features stored in the circular token bank $\mathcal{B}$ [10], ensuring that prototypes remain close to meaningful regions of the VAE feature manifold and reducing dead modes. Conversely, the **push** term imposes inter-prototype repulsion with margin δ, preventing multiple prototypes from collapsing onto redundant background-dominated regions. Together, these two terms encourage prototypes to form separated and well-covered centers over the visual feature distribution. While ProMoE [38] also introduces prototype-based routing, we use prototypes as a bridge between DiT router logits and reconstructive VAE features, rather than relying on direct prototypical matching within DiT layers alone.

Conceptually, $\mathcal{L}_{\text{align}}$ aligns the router predictions with VAE-induced semantic assignments, while $\mathcal{L}_{\text{proto}}$ learns a stable and diverse prototype topology.

Group-aware load balancing. Following DeepSeek-V3 [16], we employ a loss-free load-balancing strategy for the generic experts. Importantly, semantic experts avoid explicit load balancing, instead achieving adaptive balance naturally through semantic alignment.

3.4 Overall Objective

The full training objective is formulated as $\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{proto}}\mathcal{L}_{\text{proto}}$, where $\mathcal{L}_{\text{flow}}$ denotes flow matching loss. Notably, the proposed group-aware load balancing introduces no auxiliary optimization loss. The mechanism exclusively updates the non-gradient routing biases for the expert selection process.

4 Experiments

4.1 Implementation Details

We upcycle the low-noise checkpoint of Wan 2.2 [35] by replacing the dense feed-forward networks at even-numbered layers from 15 to 35 with sparse MoE layers, while keeping all other components unchanged. This alternating design balances stable global feature extraction with expert specialization [17]. Each MoE layer contains one continuously active shared expert and 100 routed experts, partitioned into 20 semantic and 80 generic experts. For each token, a Top-8 router activates K_s=2 semantic experts and K_g=6 generic experts. By matching the active intermediate dimension of the dense baseline, our model activates 14B of its 27B total parameters per forward pass, maintaining computational parity with the dense model. We initialize experts from the dense FFN to preserve pre-trained generative priors and train all models under the same 80k-step optimization setting.

4.2 Experimental Setup

Same-source baseline setup. To isolate component contributions, we compare same-source variants sharing the Wan 2.2 low-noise checkpoint, training data, step count, and inference settings. Unless specified, all MoE variants share the same configuration as in Sec. 4.1. *Dense Wan2.2-FT* finetunes the original dense model identically. *Wan2.2-MoE* converts Wan 2.2 into a standard MoE (100 generic experts, Top-8 routing), representing our model without the semantic branch. *SplitMoE w/o ProtoGuidance (PG)* retains the 20/80 semantic-generic partition and dual-track routing but omits VAE teacher features, semantic prototypes, $\mathcal{L}_{\text{align}}$, and $\mathcal{L}_{\text{proto}}$. *SplitMoE w/o Pull* and *SplitMoE w/o Push* respectively remove the attractive and repulsive terms of $\mathcal{L}_{\text{proto}}$. *Full SplitMoE* incorporates all proposed components. This protocol effectively disentangles dense fine-tuning, sparse scaling, role-aware expert partitioning, and prototype-guided semantic specialization.

Baseline methods and evaluation metrics. We compare SplitMoE against a wide range of SOTA text-to-video generation models, including CogVideoX-1.5 [41], Mochi [33], HunyuanVideo [13], LongCat-Video [34], LTX-2 [9], Wan2.2 [35] and OmniWeaving (think) [23]. All videos are generated using the default inference settings of each respective model with 81 frames.

We evaluate SplitMoE on two complementary benchmarks. VBench-2.0 [43], which is an upgraded version of VBench [12], assesses intrinsic faithfulness across five capability dimensions: Creativity, Commonsense, Controllability, Human Fidelity, and Physics, using a combination of state-of-the-art VLMs/LLMs and specialist anomaly detectors. T2V-CompBench [32] targets compositional



Figure 4: Qualitative comparison with baseline methods. Zoom in for better comparison. We provide more examples and comparisons with more baseline methods in the video demo.

Table 1: Text-to-Video evaluation results on VBench-2 and T2V-CompBench, best scores in **bold** and second scores with underline. A14B denotes MoE with 14B activated parameters.

		VBench-2					T2V-CompBench		
Model name	#Params.	Creativity↑	Common Sense↑	Control-lability↑	Human Fidelity↑	Physics↑	Consist attr.↑	Inter-action↑	Nu-meracy↑
Dense Wan2.2-FT	14B	49.57%	55.92%	30.37%	73.08%	54.66%	77.51%	61.08%	36.71%
Wan2.2-MoE	A14B	53.88%	57.15%	35.51%	73.49%	63.32%	78.82%	63.89%	38.55%
SplitMoE w/o PG	A14B	51.05%	63.22%	34.83%	74.04%	55.89%	78.46%	61.96%	38.57%
SplitMoE w/o Push	A14B	50.54%	62.75%	33.92%	73.60%	56.44%	77.29%	59.05%	37.15%
SplitMoE w/o Pull	A14B	52.23%	61.10%	35.67%	73.81%	56.83%	78.14%	61.15%	38.32%
CogVideoX-1.5	5B	43.66%	58.19%	29.57%	72.14%	63.23%	61.64%	60.69%	37.06%
Mochi	10B	39.84%	61.08%	30.12%	78.91%	57.61%	59.73%	53.81%	27.18%
HunyuanVideo	13B	41.84%	63.44%	28.60%	82.41%	60.20%	78.58%	63.31%	25.05%
LongCat-Video	13.6B	54.73%	**70.94%**	44.79%	80.20%	59.92%	73.38%	61.76%	**47.87%**
LTX-2	14B	50.39%	67.09%	38.09%	73.69%	**76.71%**	69.93%	47.98%	25.83%
Wan2.2	A14B	58.35%	62.45%	**48.73%**	75.33%	69.07%	83.19%	73.29%	16.96%
OmniWeaving	8.3B+7B	50.12%	61.89%	40.21%	81.15%	67.21%	83.98%	**73.94%**	37.61%
Full SplitMoE (Ours)	A14B	**58.46%**	64.89%	45.72%	**84.47%**	69.41%	**84.63%**	69.98%	44.01%

generation quality via MLLM-based, detection-based, and tracking-based metrics across three dimensions: Consistent Attribute, Interaction, and Numeracy.

4.3 Quantitative Comparison

Table 1 reports quantitative results on VBench-2 and T2V-CompBench. Our method achieves competitive performance across multiple evaluation dimensions. Compared with Wan2.2, it improves Creativity and Human Fidelity, suggesting that the proposed routing design benefits diverse content generation and human-centric video synthesis. On T2V-CompBench, our method obtains competitive compositional-alignment results, indicating that the introduced MoE structure preserves text-video consistency under compositional prompts. VBench scores for LongCat-Video and HunyuanVideo follow the LongCat-Video paper, and T2V-CompBench scores for CogVideoX-1.5 and Mochi follow the T2V-CompBench benchmark. Since independently developed models may differ in training data, scale, compute budget and post-training recipes, we use controlled same-source ablations to isolate the architectural contribution. The following section therefore focuses on MoE-specific comparisons.

Table 2: Speed comparison.

Method	Ours	Baseline MoE	Dense Model
Total Parameters	27B	27B	14B
Activated Parameters	14B	14B	14B
Training Speed	6.84s/it	6.69s/it	5.66s/it
Inference Speed	6.08s/it	6.12s/it	5.76s/it

4.4 Qualitative Comparisons

For a fair comparison, we use examples from VBench2. In Fig. 4, most baselines follow text prompts well, with Wan2.2 showing high visual quality and LTX-2 producing realistic effects. In the watermelon-cutting case, however, baselines struggle with complex object interactions: Wan2.2, LongCat-Video, and LTX-2 show local distortions as the action progresses, while the dashed boxes highlight unnatural deformation and fusion of hand morphology and watermelon geometry. OmniWeaving further suffers from abrupt content changes and inter-frame blurring. In contrast, our method maintains temporal consistency, stable physical boundaries, and structural details throughout the action. The horse case evaluates complex motion transitions from standing to running, where baselines produce multi-leg artifacts and structural anomalies under large spatiotemporal changes. Our method yields a smoother transition and preserves accurate physiological structure, benefiting from our split experts and semantic routing.

4.5 Ablation Studies

Computational overhead analysis. Tab. 2 compares training and inference speed under an iso-activated-parameter budget of 14B. Both our method and the baseline MoE show a slight per-iteration latency overhead over the dense model, mainly due to routing operations and memory bandwidth bottlenecks. Compared with the baseline MoE, our architecture adds negligible overhead and maintains highly comparable speed.

Efficacy of MoE Upcycling. We evaluate sparse capacity scaling by comparing SplitMoE with a dense fine-tuning baseline. As shown in the first row of Table 1, Dense Wan2.2-FT fine-tunes the low-noise Wan2.2 model on the same training data and evaluates it on VBench-2. Under the same active-parameter budget, the full SplitMoE outperforms this dense baseline across all evaluation dimensions, indicating that MoE upcycling raises capacity and the performance ceiling without increasing per-token activated parameters. The validation diffusion loss in Fig. 5(a) further supports this trend: SplitMoE converges faster, reaches comparable loss with roughly 70% of the training steps, and maintains lower loss throughout training. These results show that our upcycling strategy preserves dense-model optimization stability while improving representational capacity through structure-aware sparse routing. Notably, whereas the official Wan2.2 uses a high-/low-noise dual-model ensemble, Dense Wan2.2-FT is trained only from the low-noise checkpoint and thus naturally underperforms the full ensemble due to missing high-noise priors and reduced capacity.

Effect of structured semantic routing. We compare SplitMoE with two MoE baselines using similar expert configurations and parameter budgets. As shown in Table 1 and Fig. 4, Wan2.2-MoE converts the low-noise Wan2.2 model into a standard MoE and fine-tunes it on the same data, while SplitMoE w/o PG keeps the same 20–80 expert partition as SplitMoE but removes prototype guidance. Their similar performance suggests that expert partitioning alone brings limited gains without explicit semantic guidance. In the left example in Fig. 4, the watermelon struggles to maintain its complete semantics. In contrast, the full SplitMoE performs better on semantics-sensitive dimensions such as Human Fidelity and Physics, indicating that prototype-guided semantic routing improves expert specialization beyond naive sparse scaling. Fig. 5(b) supports this conclusion, as removing the semantic branch yields higher validation diffusion loss than the full model.

Coupled effects of pull-push prototype guidance. We ablate the two complementary terms in $\mathcal{L}_{\text{proto}}$ to test whether prototype guidance needs balanced attraction and repulsion. As shown in Table 1, using only one term does not reliably improve quantitative metrics and can even underperform the variant without prototype guidance, since partial supervision may introduce a biased routing

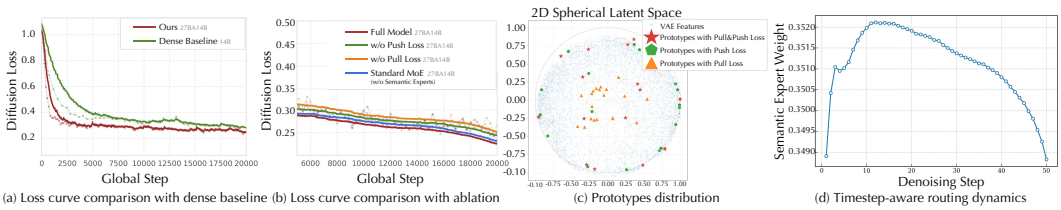


Figure 5: Ablation study analysis.

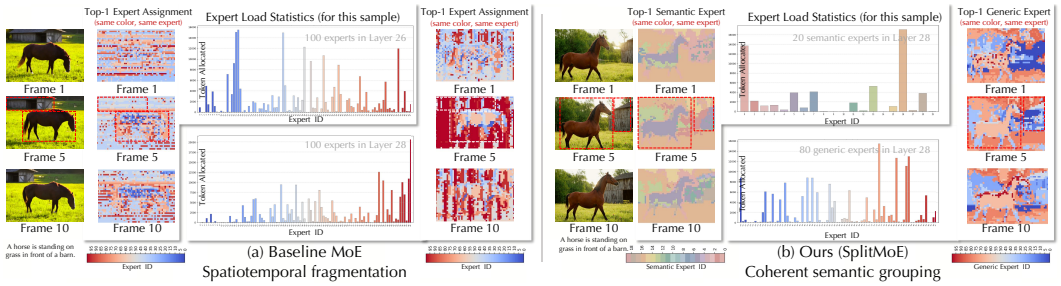


Figure 6: Comparison of spatial routing and expert loads. (a) Baseline MoE scatters correlated patches across experts under strict capacity constraints, causing fragmented routing and poor spatiotemporal consistency. (b) SplitMoE routes coherent entities to Semantic Experts and residual textures to Generic Experts, achieving stable utilization without enforcing uniformity.

prior rather than a stable semantic partition. Fig. 5(b) shows the same trend in validation diffusion loss, where removing either term worsens optimization against the full model. The VAE-space visualization in Fig. 5(c) explains these failure modes. Without push, pull attracts prototypes toward dominant visual regions, preventing full coverage of the VAE manifold and limiting the semantic experts' representation space. Without pull, push drives many prototypes away from the data manifold, producing underused or dead semantic experts and weakening semantic routing. Thus, pull and push form a coupled mechanism: pull anchors prototypes to meaningful visual semantics, while push prevents redundant collapse. Together, they yield a balanced prototype layout that supports effective semantic expert specialization.

Routing pathologies in video MoE. Consistent with Fig. 2, where standard MoE shows low semantic distinctiveness and intra-region routing dominance, Fig. 6(a) visualizes how this mismatch appears spatially and temporally. The Top-1 expert maps across frames show that coherent regions such as the horse and barn are not routed consistently. As highlighted by the dashed boxes, standard routing produces three artifacts: *(i) Temporal jitter*, where similar regions switch experts across frames; *(ii) Spatial striping*, where homogeneous areas are artificially split to satisfy load constraints; and *(iii) Semantic fragmentation*, where adjacent tokens from the same object scatter across unrelated experts. These results indicate that rigid token-level balancing is ill-suited to video tokens. In contrast, Fig. 6(b) shows that SplitMoE assigns coherent object regions to stable semantic experts while using the generic branch to absorb residual textures and backgrounds, thereby reducing jitter, striping, and fragmentation.

Timestep-aware routing dynamics. We further track the routing weights assigned to semantic experts over a 50-step denoising trajectory in Fig. 5(d). The semantic weight peaks in the early high-noise stage, around steps 10–15, when the model establishes global layout, object placement, and coarse semantic structure. It then gradually decreases as denoising shifts toward local appearance and detail refinement. This coarse-to-fine pattern shows that the semantic branch is not redundant sparse capacity; instead, it learns a stage-dependent role, allocating semantic computation to the phases where high-level structure formation is most needed.

5 Conclusion

In this work, we identify the *uniformity trap* as a key limitation of LLM-style MoE routing for video diffusion and propose SplitMoE, a split-role MoE that separates semantic abstraction from general visual residual modeling. With semantic-generic expert partitioning, prototype-guided routing, and pull-push regularization, SplitMoE improves convergence, routing coherence, and generation quality under matched activated-parameter budgets. Its timestep-aware routing reveals an emergent coarse-to-fine pattern, suggesting modality-aware sparse routing as a useful direction. Limitations include reliance on frozen visual features and sparse-routing overhead, motivating future work on adaptive partitioning, efficient distributed routing, and broader video generation tasks.

Acknowledgments and Disclosure of Funding

Funding and support. This work was conducted during Yu Xu's internship at ByteDance. The authors received no additional funding or in-kind support for this work.

Competing interests. Beyond the affiliations listed in the author block and Yu Xu's internship at ByteDance, the authors declare no other competing financial interests.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[2] Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. HunyuanImage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

[3] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[4] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

[5] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024.

[6] Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. MegaBlocks: Efficient sparse training with mixture-of-experts. *Proceedings of Machine Learning and Systems*, 5:288–304, 2023.

[7] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

[8] Google DeepMind. Veo: A state-of-the-art video generation model. https://deepmind.google/technologies/veo/, 2024. Accessed: 2026-02-22.

[9] Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. LTX-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[11] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[12] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[13] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[14] Kuaishou. Kling AI. https://klingai.kuaishou.com/, 2024.

[15] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[16] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[17] Yahui Liu, Yang Yue, Jingyuan Zhang, Chenxi Sun, Yang Zhou, Wencong Zeng, Ruiming Tang, and Guorui Zhou. Efficient training of diffusion mixture-of-experts models: A practical recipe. *arXiv preprint arXiv:2512.01252*, 2025.

[18] Luma Labs. Dream Machine. https://lumalabs.ai/dream-machine, June 2024. Accessed: 2026-02-22.

[19] MiniMax. Hailuo AI. https://hailuoai.com/video, September 2024. Accessed: 2026-02-22.

[20] OpenAI. Sora. https://openai.com/sora/, 2024.

[21] OpenAI. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, 2024.

[22] OpenAI. Sora2. https://openai.com/index/sora-2/, 2025.

[23] Kaihang Pan, Qi Tian, Jianwei Zhang, Weijie Kong, Jiangfeng Xiong, Yanxin Long, Shixue Zhang, Haiyi Qiu, Tan Wang, Zheqi Lv, et al. OmniWeaving: Towards unified video generation with free-form composition and reasoning. *arXiv preprint arXiv:2603.24458*, 2026.

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

[25] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. Open-Sora 2.0: Training a commercial-level video generation model in $200k. *arXiv preprint arXiv:2503.09642*, 2025.

[26] Runway. Gen-2: Generate novel videos with text, images or video clips. https://runwayml.com/research/gen-2, 2023. Accessed: 2026-02-22.

[27] Runway. Gen-3. https://runwayml.com/, June 2024. Accessed: 2026-02-22.

[28] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[29] Tao Shen, Xin Wan, Taicai Chen, Rui Zhang, Junwen Pan, Dawei Lu, Fanding Lei, Zhilin Lu, Yunfei Yang, Chen Cheng, et al. MammothModa2: A unified AR-diffusion framework for multimodal understanding and generation. *arXiv preprint arXiv:2511.18262*, 2025.

[30] ShengShu-AI. Vidu. https://www.vidu.studio/, July 2024. Accessed: 2026-02-22.

[31] Minglei Shi, Ziyang Yuan, Haotian Yang, Xintao Wang, Mingwu Zheng, Xin Tao, Wenliang Zhao, Wenzhao Zheng, Jie Zhou, Jiwen Lu, et al. DiffMoE: Dynamic token selection for scalable diffusion transformers. *arXiv preprint arXiv:2503.14487*, 2025.

[32] Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2V-CompBench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8406–8416, 2025.

[33] Genmo Team. Mochi 1. https://github.com/genmoai/models, 2024.

[34] Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, et al. LongCat-Video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

[35] Team Wan, 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, Mengyang Feng, 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. *arXiv preprint arXiv:2503.20314*, 2025.

[36] Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. SRA 2: Variational autoencoder self-representation alignment for efficient diffusion training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 32978–32987, 2026.

[37] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[38] Yujie Wei, Shiwei Zhang, Hangjie Yuan, Yujin Han, Zhekai Chen, Jiayu Wang, Difan Zou, Xihui Liu, Yingya Zhang, Yu Liu, et al. Routing matters in MoE: Scaling diffusion transformers with explicit routing guidance. *arXiv preprint arXiv:2510.24711*, 2025.

[39] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, 2021.

[40] Yu Xu, Hongbin Yan, Juan Cao, Yiji Cheng, Tiankai Hang, Runze He, Zijin Yin, Shiyi Zhang, Yuxin Zhang, Jintao Li, et al. TAG-MoE: Task-aware gating for unified generative mixture-of-experts. *arXiv preprint arXiv:2601.08881*, 2026.

[41] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[42] Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Prompt spectrum for attribute-aware personalization of diffusion models. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.

[43] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

[44] Youwei Zheng, Yuxi Ren, Xin Xia, Xuefeng Xiao, and Xiaohua Xie. Dense2MoE: Restructuring diffusion transformer to MoE for efficient text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18661–18670, 2025.

A Technical appendices and supplementary material

A.1 Architectural Details

We instantiate SplitMoE by upcycling the feed-forward networks of the Wan 2.2 low-noise model. Specifically, the dense FFNs at the selected layers are replaced with sparse MoE layers, while the attention blocks, normalization layers, text-conditioning modules, and other non-MoE parameters are kept unchanged. Each MoE layer contains one continuously active Shared Expert and 100 routed experts, consisting of 20 Semantic Experts and 80 Generic Experts. For each token, the router activates K=8 routed experts in total, with K_s=2 selected from the semantic expert pool and K_g=6 selected from the generic expert pool.

Each routed expert uses an intermediate dimension of 1536, and the Shared Expert uses an intermediate dimension of 3072. Therefore, the active intermediate dimension per token is

$$K \times 1536 + 3072 = 8 \times 1536 + 3072 = 15360,$$

which closely matches the dense FFN dimension of 13824 in Wan 2.2 with hidden dimension 5120. This design keeps the activated computation comparable to the dense baseline while increasing the total sparse capacity. In our implementation, the resulting MoE model contains 27B total parameters and activates 14B parameters per forward pass.

To preserve the generative prior of the pre-trained dense model, all experts are initialized from the original dense FFN neurons. For each replaced FFN, neurons are ranked by their ℓ_2 norms. The top-ranked neurons are copied to the Shared Expert, which is intended to retain high-utility components that are broadly useful across tokens. The remaining neurons form a residual pool. We apply K-Means clustering to this pool to initialize the Semantic Experts, providing a structured starting point for later semantic specialization under prototype-guided regularization. The Generic Experts are initialized by sampling from the residual neuron pool with small Gaussian perturbations, which preserves the statistics of the dense FFN while introducing mild diversity among experts. Router biases are initialized to zero.

A.2 Ablation Study of MoE Layers and Experts Configuration

Expert configuration. We ablate the ratio of Semantic to Generic experts while keeping the total expert count fixed at 100. As shown in the top of the Table 3, reducing the number of semantic experts below 20 leads to a consistent drop across all dimensions, indicating that a sufficient semantic expert capacity is necessary to disentangle concept-level representations from low-level visual patterns. Conversely, progressively increasing the semantic-to-generic ratio (40S+60G, 50S+50G, 60S+40G, 80S+20G) also degrades performance, demonstrating that generic experts play a critical role in learning general-purpose visual representations shared across diverse prompts, that means their reduction deprives the model of the capacity to handle common low-level patterns. Our 20S+80G configuration strikes the optimal balance, achieving the highest scores on Creativity, Controllability, and Human Fidelity while maintaining competitive performance on Commonsense and Physics.

MoE layer placement. We further evaluate the effect of where MoE layers are inserted by comparing placements at layers 0–20, 20–40, and 0–40 against our default 15–35 setting, where in all cases 10 MoE layers are evenly distributed across the specified range. As shown in the bottom of the Table 3, our middle-layer placement consistently outperforms alternatives on semantics-sensitive dimensions such as Controllability, Human Fidelity and Physics, suggesting that intermediate transformer layers are more critical sites for semantic specialization than early or late layers, which are better suited to low-level feature extraction and high-level synthesis, respectively.

A.3 Comparison with Commercial Models

We further benchmark our method against closed-source commercial models, including Sora, Kling, Vidu-Q1, Seedance, and Veo3, on VBench-2.0 in Table 4. Commercial models benefit from significantly larger training data, proprietary post-training pipelines, and dedicated aesthetic fine-tuning at a scale inaccessible to academic research, making direct comparison inherently asymmetric. Nevertheless, our method remains competitive on Controllability and Physics, surpassing commercial baselines except Veo3 or Vidu-Q1 on the former, and matching or exceeding most on the latter. This

Table 3: Ablation study on different experts configurations and MoE layer placement.

Experts Config.	Layer Placement	Creativity↑	Common Sense ↑	Control-lability ↑	Human Fidelity↑	Physics↑
10S+90G	Layer 15-35	58.24%	62.88%	42.82%	79.43%	68.32%
40S+60G	Layer 15-35	57.27%	64.33%	45.02%	80.72%	71.08%
50S+50G	Layer 15-35	57.01%	63.85%	45.11%	80.66%	70.77%
60S+40G	Layer 15-35	56.58%	63.77%	43.33%	80.64%	69.61%
80S+20G	Layer 15-35	55.54%	61.96%	40.29%	78.67%	66.67%
20S+80G	Layer 0-20	56.88%	64.01%	45.22%	83.81%	68.38%
20S+80G	Layer 20-40	57.74%	64.27%	45.03%	83.79%	68.89%
20S+80G	Layer 0-40	58.04%	65.18%	45.53%	84.22%	69.33%
20S+80G(Ours)	Layer 15-35	58.46%	64.89%	45.72%	84.47%	69.41%

Table 4: Comparison with commercial models on VBench 2.0 benchmark.

Model name	Creativity↑	Common Sense ↑	Control-lability ↑	Human Fidelity↑	Physics↑
Sora-480p	60.57%	64.32%	22.09%	87.72%	57.18%
Kling-1.6	48.58%	65.45%	33.05%	83.56%	64.35%
Vidu-Q1	56.54%	65.98%	38.13%	81.24%	71.63%
Seedance1.0-pro	53.04%	64.31%	39.84%	77.06%	64.81%
Veo3	60.85%	69.48%	47.04%	86.88%	69.35%
Ours	58.46%	64.89%	45.72%	84.47%	69.41%

suggests that our semantic routing mechanism effectively enhances the model's ability to faithfully follow fine-grained text instructions and respect physical plausibility, capabilities that stem directly from structured semantic specialization rather than data scale.

A.4 User Study

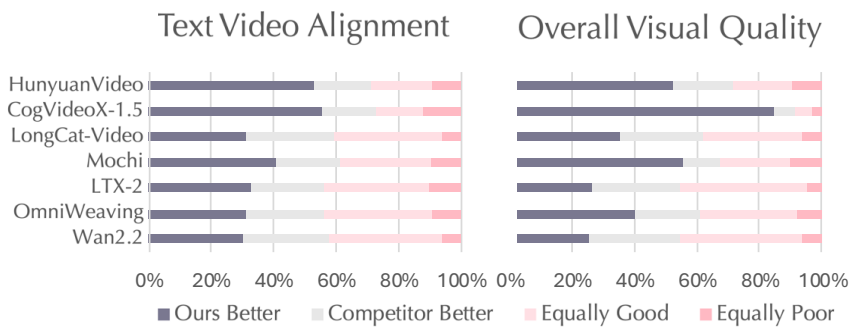


Figure 7: GSB user study results.

We conduct a user study to complement the automated metrics with human perceptual judgment. We randomly sample 90 video prompts from VBench2.0 [43] and generate outputs from SplitMoE and all 7 baselines, yielding 630 pairwise comparisons. A total of 65 participants evaluated each pair under a Good/Same/Bad (GSB) protocol across two dimensions: text-video alignment and overall visual quality. As shown in Fig. 7, SplitMoE consistently receives higher preference rates against all baselines on text-video alignment, confirming that the semantic routing mechanism effectively enhances prompt controllability. While Wan 2.2 achieves a marginal advantage in visual quality attributable to its dedicated aesthetic fine-tuning on curated cinematic data [35], SplitMoE surpasses it on text-video alignment. Notably, SplitMoE outperforms LongCat-Video on visual quality despite requiring no post-training, underscoring the efficiency of our upcycling strategy.

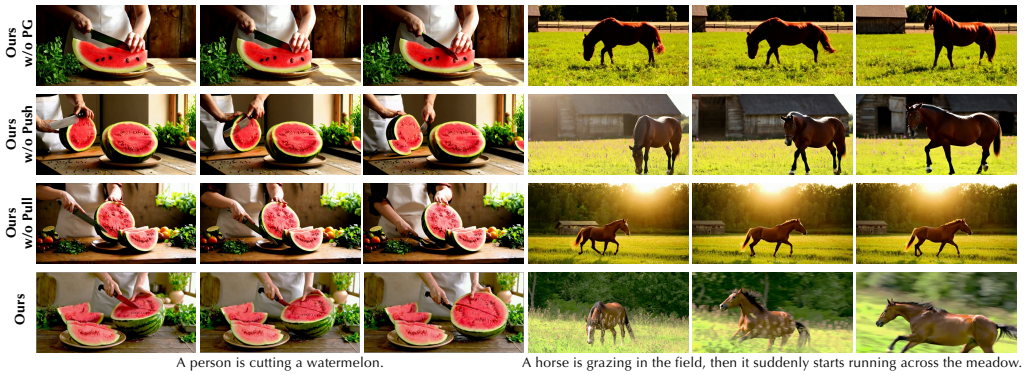


Figure 8: Qualitative results of ablation study.

Qualitative ablation on regularization components. Fig. 8 provides qualitative comparisons of removing each regularization term. Without the prototype-guided regularization (w/o PG), the model fails to capture dynamic semantics: in the horse prompt, the generated subject remains stationary rather than transitioning to a running motion as described, suggesting that semantic experts collapse into generic representations without prototype supervision. Removing the push loss (w/o Push) leads to cross-frame semantic inconsistencies, most visibly in the watermelon sequence where object appearance and scene context drift noticeably between frames, indicating that expert diversity is insufficient to maintain stable semantic assignments over time. Removing the pull loss (w/o Pull) similarly degrades temporal coherence, producing abrupt subject-level changes across frames. Only the full model faithfully renders the action transition in the horse sequence and maintains consistent object semantics throughout the watermelon cutting scene, demonstrating that all three components are necessary for coherent spatio-temporal generation.

A.5 Visual Statistical Analysis of MoE Routing

To quantitatively evaluate whether the proposed MoE routing mechanism develops meaningful semantic organization, we conduct a visual statistical analysis on routing logs collected from 250 generated videos. This analysis aims to reveal how different expert types, *i.e.*, Semantic Experts and Generic Experts, behave when processing regions with explicit physical and semantic meanings.

Semantic token annotation. Given the generated videos, we first obtain pixel-level semantic labels for each frame using a pretrained SegFormer model [39]. Specifically, we use the ADE20K-finetuned checkpoint `nvidia/segformer-b0-finetuned-ade-512-512`. The resulting segmentation masks are downsampled to match the spatial resolution of the DiT token grid [24]. This produces a direct correspondence between each visual token and its semantic category, allowing us to associate token-level MoE routing decisions with semantic regions. We then analyze the top-K expert assignments recorded from the MoE layers [28]. In our configuration, each token selects $K_s = 2$ experts from 20 Semantic Experts and $K_g = 6$ experts from 80 Generic Experts.

Intra-sample routing dominance. We first measure whether tokens from the same semantic region are routed consistently within each video sample. For a semantic category c in sample s, let $\mathcal{T}_{s,c}$ denote the set of tokens labeled as category c, and let a_i be the selected expert assignment of token i. We define the dominant expert for this region as the expert receiving the largest number of assignments:

$$e^{\star}_{s,c} = \arg\max_{e} \sum_{i \in \mathcal{T}_{s,c}} \mathbb{I}(a_i = e). \tag{9}$$

The intra-sample routing dominance is then computed as

$$D_{s,c} = \frac{1}{|\mathcal{T}_{s,c}|} \sum_{i \in \mathcal{T}_{s,c}} \mathbb{I}(a_i = e^{\star}_{s,c}). \tag{10}$$

A higher value indicates that a coherent semantic region, such as sky, face, or body, is mainly handled by a consistent expert group, suggesting stronger spatial routing coherence and less fragmentation.

Semantic distinctiveness. We further evaluate whether different semantic categories collapse onto the same dominant experts. For each MoE layer, if an expert e is selected as the dominant expert for category c, we count how many semantic categories also use the same expert as their dominant expert. Let $\mathcal{C}_e$ denote the set of semantic categories whose dominant expert is e. The semantic distinctiveness of category c is defined as

$$S_c = \frac{1}{|\mathcal{C}_{e_c^\star}|}, \tag{11}$$

where $e_c^\star$ is the dominant expert for category c after aggregating statistics over samples. A higher distinctiveness score means that the expert is shared by fewer semantic categories, indicating stronger semantic specialization and reduced expert homogenization.

Discussion. Compared with a standard MoE baseline that expands the FFN layers of the Wan backbone into MoE layers, our SplitMoE achieves higher intra-sample routing dominance while also improving semantic distinctiveness. This indicates that the proposed semantic expert pool does not merely concentrate tokens into a few popular experts. Instead, it routes tokens from the same semantic region to consistent expert groups while assigning different semantic regions to more distinguishable experts. These results support the effectiveness of our semantic-generic expert partitioning: Semantic Experts capture high-level physical and semantic concepts, whereas Generic Experts retain flexible capacity for residual appearance variations, textures, and background details.

A.6 Limitation, Future Work and Societal Impacts

Limitations. Although SplitMoE improves the efficiency-quality trade-off of video diffusion transformers, several limitations remain. First, the semantic routing branch relies on frozen visual features and learnable prototypes to provide a stable semantic topology. While this design avoids manual labels and improves routing coherence, the quality of the learned expert specialization may still depend on the coverage and bias of the frozen visual encoder. Second, the proposed architecture introduces additional routing operations, prototype regularization, and expert dispatching. Although the measured runtime overhead is small under our implementation, further kernel-level optimization and distributed routing design are needed for more efficient large-scale deployment.

Future work. Future work will investigate several directions. We plan to extend SplitMoE to broader generation settings, including image-to-video generation, video editing, and controllable multi-condition generation. Another promising direction is to develop adaptive expert partitioning, where the number of semantic and generic experts can be dynamically determined according to the data distribution and denoising stage. In addition, improving the interpretability of routing decisions may provide a more direct understanding of how different experts contribute to object structure, motion consistency, and local appearance.

Societal impacts. This work advances text-to-video generation by improving the routing structure and parameter efficiency of sparse video diffusion models. The proposed method may lower the computational cost of scaling video generation systems and can benefit applications such as creative content production, visual prototyping, education, simulation, and accessibility tools. By improving temporal coherence and structural consistency, the method may also reduce the amount of manual correction required in professional video creation workflows. At the same time, more capable video generation models may amplify existing risks associated with synthetic media. Therefore, we encourage the use of video generation models in a safe, fair, and transparent manner.

A.7 More of our results

We present additional qualitative results in Fig. 9, covering a diverse range of video generation scenarios, including fine-grained human actions, human–human interaction, human–animal interaction, animal motion, and dynamic activities in complex indoor and outdoor scenes. These examples demonstrate that our method can generate temporally coherent videos with consistent subjects, plausible motion transitions, and faithful alignment with text prompts across different visual domains.



A person is sitting at the table, then they suddenly start drawing on a notepad.



An old man wearing a cloak gestured with his hands, and magic appeared at his fingertips.



The girl was grooming the dog sitting on the ground.



Two men playing chess in the park.



A man and a woman embraced next to the train.



The man stroked the cat sitting on the windowsill.



A red fox walks in the snow.



The boy is skateboarding in the spacious indoor parking lot.



Figure 9: More text to video results from the proposed SplitMoE.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims regarding the SplitMoE architecture and its ability to mitigate spatiotemporal fragmentation are clearly stated in the Abstract and explicitly listed at the end of Section 1 (Introduction).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated discussion regarding the limitations of our proposed SplitMoE architecture and directions for future work is provided in Appendix A.6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All mathematical formulations governing our SplitMoE architecture, including the Contrastive Routing Loss and the physical dynamics of the prototype-guided mechanism, are rigorously derived and clearly presented in Section 3, with all necessary underlying assumptions explicitly stated.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The baseline model, training steps, and expert configuration are described in Section 4.1, with additional architectural details in Appendix A.1.

Guidelines:

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

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

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

   Answer: [No]

   Justification: We do not provide a complete public package of data, source code, and instructions sufficient to reproduce all main experimental results. The method and experimental setup are described in Sections 3 and 4.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The evaluation benchmarks and model configurations are described in Sections 4.1 and 4.2, with further architectural details in Appendix A.1.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the extreme computational cost associated with training and evaluating large-scale video diffusion models (as detailed in Section 4.1), we did not perform multiple full-training runs across different random seeds to generate error bars. Instead, we report the absolute performance on standardized evaluation benchmarks (e.g., VBench-2.0 and T2V-CompBench) and user studies, following the established conventions in the video generation community.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 4.1 explicitly states the compute resources used, specifying that the model was trained for 80k steps using bf16 precision on 64 NVIDIA H100 GPUs.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted conforms in every respect with the NeurIPS Code of Ethics, as it focuses on architectural improvements for generative models without introducing novel ethical violations.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The societal impacts are discussed in Appendix A.6. While our work focuses on foundational efficiency improvements, we acknowledge that high-fidelity video generation inherently carries dual-use risks, such as the creation of deepfakes or misinformation, which require community-wide mitigation efforts.

Guidelines:

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

11. **Safeguards**

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

Answer: [Yes]

Justification: Our work involves fine-tuning an existing open-source video generation model (Wan 2.2). Any public release of our model weights will strictly adhere to the usage restrictions and safeguard protocols established by the original foundation model.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The creators of the pre-trained foundation model (Wan 2.2) and the evaluation benchmark (VBench-2.0 and T2V-CompBench) are properly credited through formal citations. All utilized assets are open-source, and we strictly adhere to their original licensing terms (e.g., Apache License 2.0) as specified in their official repositories.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: No new datasets, source code, or model checkpoints are released as part of this paper.

Guidelines:

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

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

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

Answer: [Yes]

Justification: The user study was conducted in accordance with standard ethical guidelines for human subjects research. Participants were informed of the study's nature, and the protocol was reviewed to ensure minimal risk with relevant ethical compliance details.

Guidelines:

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

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

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

    Answer: [Yes]

    Justification: The user study described in Appendix. A.4 involved minimal risk (simply evaluating the visual quality of generated videos) and did not collect sensitive personal data. It was conducted in strict adherence to the NeurIPS Code of Ethics and standard ethical protocols, ensuring that all participation was entirely voluntary.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core methodology of our SplitMoE architecture is purely based on Diffusion Transformers and Mixture-of-Experts. It does not involve LLMs as an important, original, or non-standard component in the research methodology.

    Guidelines:

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