SSBP: Stage-Specialized Block Pruning for Video Diffusion Models

Mengmeng Ge[*1], Takashi Isobe[*†1], Dong Zhou[1], Dong Li[1], and Emad Barsoum[1]

[1]Advanced Micro Devices, Inc.
{k15201363625,panisobe}@gmail.com

Fig. 1: Visual comparison of 4-step generation results: (a) original Wan2.1 model with DMD2 distillation, (b) global-importance-based block pruning with 30% acceleration, and (c) Stage-Specialized Block Pruning (SSBP) with 30% acceleration.

Abstract. Recent advances in video diffusion models have delivered remarkable generation quality, yet their generation cost remains high. This inefficiency is largely driven by the iterative denoising process, with computationally expensive per-step architectures. Step distillation enables few-step generation, but per-step latency remains high because a large-scale backbone is invoked at each denoising step. Several works seek to reduce per-step latency via block pruning using importance scores averaged across denoising steps. However, such averaged scores can obscure stage-critical blocks and bias the pruning process, removing blocks essential for certain stages while leaving redundancy in others. In this paper, we propose a novel pruning paradigm, Stage-Specialized Block Pruning, that partitions the denoising process into multiple stages and assigns a tailored pruned variant to each stage. Each variant consists of shared blocks across multiple stages and stage-specific blocks for its assigned stage. With parameter sharing, the overall parameter budget is smaller than the original model, while the per-step computational footprint is reduced for lower latency. To derive stage-specialized variants, we introduce

* Equal contribution.
† Corresponding author.

a hierarchical selection mechanism consisting of an Inter-Stage Variance Selector (IVS) and an Intra-Stage Quality Selector (IQS). IVS compares the variance of stage-wise block-importance distributions and prioritizes the stage with higher variance for pruning, indicating greater removable redundancy. IQS mitigates noisy importance estimation by generating multiple low-importance pruning candidates within the selected stage and choosing the one with the highest video generation quality. IVS and IQS are applied iteratively until reaching a target latency budget. Furthermore, we introduce a two-phase training pipeline to recover the performance of these variants and facilitate sequential composition for few-step generation. Extensive experiments on the representative video diffusion model Wan2.1-1.3B demonstrate that our method reduces per-step inference cost while maintaining competitive performance on VBench.

Keywords: Stage-Specialized Pruning · Stage-aware Training · Video Diffusion Models

1 Introduction

Video diffusion models [2, 8, 9, 30] have recently become a leading paradigm for high-quality video generation, enabling realistic appearance and coherent motion across diverse content. However, their high generation quality comes at the cost of expensive iterative denoising, where numerous sampling steps and heavy backbone computations lead to substantial latency. This makes efficient acceleration crucial for practical deployment.

A dominant line of work focuses on step distillation, reducing the number of diffusion steps from tens of steps (e.g., 50) to only a few (e.g., 4). However, step distillation alone remains insufficient to meet deployment efficiency requirements, since each remaining step still incurs substantial computation. Block pruning [14, 21, 23, 29] offers a complementary solution by removing redundant blocks to reduce per-step latency. It provides a straightforward mechanism to trade model capacity for speed. Prior studies [14, 29] show an approximately linear relationship between pruning ratio and achieved speedup, making pruning both controllable and flexible. Moreover, block-level pruning is orthogonal to methods that replace expensive layers within a block (e.g., transformer attention) with efficient alternatives. These methods can be combined to further accelerate end-to-end video generation.

Existing block-wise pruning methods for diffusion models can be broadly categorized into (1) importance-based pruning and (2) learnable mask-based pruning. Importance-based pruning relies on step-wise proxies of a block's contribution during inference, such as feature similarity between adjacent blocks or loss-sensitive criteria estimated from local curvature (e.g., Hessian-based pruning). These step-wise scores are then averaged over denoising steps to obtain a final per-block score. Blocks with low averaged scores are treated as redundant and are pruned accordingly, yielding a lightweight architecture. The pruned model is then fine-tuned to recover performance. However, block importance varies across

denoising stages, as different stages emphasize different aspects of generation, such as global structure, motion, and fine details. Therefore, pruning based on averaged importance can lead to a stage-imbalanced model allocation, removing blocks that are essential for certain portions of the denoising process while retaining redundancy elsewhere. In addition, many importance-guided pipelines lack explicit feedback on generation quality during pruning, which can further limit the recoverability of the pruned models (Fig. 1 (b)). Learnable mask-based pruning takes a different route by optimizing trainable binary masks over blocks under a target objective that combines latency constraints with quality-related losses. While conceptually appealing, these methods face several practical challenges. First, learning masks introduces an additional costly training phase and often requires extensive hyperparameter tuning. Second, designing a stable objective that balances latency and quality is non-trivial, and optimization can be unstable when training discrete masks, especially under tight latency budgets that force the masks to become much sparser, making convergence more difficult. Third, masks learned as a single global decision may still struggle to generalize across the denoising process.

Motivated by these limitations, we propose a novel pruning paradigm, Stage-Specialized Block Pruning (SSBP), that partitions the denoising process into multiple stages and assigns a tailored pruned variant to each stage (Fig. 1 (c)). To derive stage-specialized variants, we introduce a hierarchical selection mechanism consisting of an Inter-Stage Variance Selector (IVS) and an Intra-Stage Quality Selector (IQS). IVS decides where to prune by computing the variance of block-importance scores for each stage and prioritizing the stage with the largest variance. Higher variance implies a clearer separation between important and redundant blocks, indicating greater pruning headroom. After selecting a stage, IQS determines which blocks to prune within that stage. Because importance estimates in the low-score region can be noisy, naively removing the single lowest-scoring block may lead to irreversible quality degradation. Instead, IQS forms multiple pruning candidates from the low-importance set, evaluates the resulting outputs using a quality metric, and chooses the candidate with the best score. IVS and IQS are applied iteratively until the target latency budget is met, yielding a stage-adaptive capacity allocation. Each pruned variant includes blocks shared by multiple variants, as well as stage-specific blocks specialized for its own stage. With parameter sharing, the overall parameter budget is smaller than the original model, and the per-step footprint is reduced for lower latency. To recover performance and enable few-step generation, we first fine-tune each pruned variant on its assigned timestep range to restore its essential denoising capability. We then apply DMD2 [31] to the fine-tuned variants with inference-time simulation, enabling them to operate sequentially along the resulting few-step denoising trajectory. We stochastically truncate the later portion of the sampling chain during training, so that each stage-specialized variant receives a direct learning signal. During training, shared blocks are updated and synchronized across variants, while stage-specific blocks are updated independently within each variant.

Extensive experiments show that SSBP reduces the per-step latency of video diffusion model Wan2.1-1.3B [24]. With our training pipeline, the pruned variants recover performance and achieve competitive results on VBench [11] in the few-step setting. Our main contributions are summarized as follows:

- We introduce SSBP, a new pruning paradigm that derives stage-specialized variants for different portions of the denoising trajectory, reducing per-step latency for efficient video generation.
- We propose a hierarchical selection mechanism with an IVS and an IQS to iteratively prune blocks in a stage-adaptive manner, meeting a target latency budget.
- We introduce a two-phase training pipeline to recover performance for these stage-wise variants and enable them to operate sequentially for few-step sampling.
- Extensive experiments show that SSBP further accelerates state-of-the-art video diffusion models in the few-step setting by reducing per-step latency, while achieving competitive results on VBench.

2 Related work

Acceleration via step distillation. A primary speed bottleneck of diffusion models stems from their inherently iterative sampling process, which is particularly expensive for large-capacity backbones. Consistency Models [36], as a pioneering work, provide key insights into step distillation and demonstrate its feasibility under the probability flow ODE (PF-ODE) formulation. Latent consistency models (LCM) [20] further advances this paradigm by performing consistency distillation in the latent space. Following these works, many subsequent studies have explored variants of consistency distillation and achieved promising results. Beyond consistency distillation, score-based distillation has also attracted considerable attention. Instead of enforcing step-wise consistency, score distillation aims to match the student distribution to the teacher distribution when samples are generated from a noise prior. The well-known Distribution Matching Distillation (DMD) introduces a KL-divergence objective under the variational score distillation framework [32]. DMD2 [31] improves DMD with backward simulation to mitigate train-inference mismatch, and removes the regression loss to preserve the student's capacity. These distillation philosophies have also been extended to video diffusion models. VideoLCM [25] enables four-step video generation via LCM-style distillation. T2V-Turbo [16] and T2V-Turbo-V2 [17] refine the consistency objective to improve few-step video quality. rCM [37] further combines consistency distillation with score distillation to enhance generation diversity. The concurrent Phased DMD [5] alleviates the quality degradation of DMD2 by introducing phase-specific LoRA adapters across denoising stages to increase the model capacity. To further improve efficiency, some recent studies combine step distillation with complementary techniques such as efficient block replacements to lower per-step latency.

Acceleration via pruning. Existing pruning schemes for video diffusion models can be broadly categorized into importance-guided pruning [12–14, 26, 29, 33, 38] and learnable mask-based pruning [6, 7, 27, 36]. For importance-guided pruning, ICMD [26] computes block-wise FVD scores along the denoising trajectory to identify and remove redundant blocks. Some works [12, 13, 29] estimate block importance at each denoising step using Optimal Brain Surgeon (OBS) [10], and prune blocks based on the trajectory-averaged importance. These methods rely on averaged importance as a heuristic and may overlook stage-wise differences across diffusion timesteps. As a result, they may prune blocks that are critical for particular stages, while retaining blocks that are redundant in those stages. For learnable mask-based pruning, Diff-Pruning [7] adopts gradient-based techniques to reduce model size. TinyFusion [6] jointly learns pruning masks and a LoRA-parameterized weight update to simulate post-pruning fine-tuning. However, learning pruning masks is often sensitive to hyperparameters and optimization settings, and typically requires different training settings for different target pruning ratios. In addition, larger pruning ratios further increase optimization difficulty, leading to unstable convergence.

Acceleration via efficient block. Current video diffusion models are primarily built upon Diffusion Transformers (DiTs) [15, 19, 24, 30]. Despite their strong generative capability, the multi-head self-attention (MHA) and feed-forward network (FFN) in each Transformer block incur substantial computational cost. Synthesizing a single video can take several minutes even on high-end GPUs. A straightforward approach is to design lightweight diffusion architectures [28, 29]. However, this typically requires training from scratch, relying on large-scale datasets and substantial compute. For example, SnapGen-V [28] reports training with over 150K iterations on 256 GPUs. An alternative line of work improves efficiency by replacing quadratic-cost attention layers with more efficient alternatives while preserving the essential diffusion capability, such that degraded performance can be recovered through efficient training. VSA [35] leverages inherent attention sparsity and replaces full attention with sparse attention. SLA [34] replaces standard attention with a linear-complexity attention. These efficient-block-based methods are orthogonal to pruning-based acceleration.

3 Method

In this section, we first conduct two pilot studies (Section 3.1) to systematically analyze existing importance-based block-pruning paradigms and reveal their key limitations. Motivated by these observations, we propose a stage-aware pruning paradigm for stage-specialized diffusion models (Section 3.2). We then present a hierarchical selection mechanism that implements this paradigm via iterative stage-aware pruning (Section 3.3). Finally, we introduce a two-phase training pipeline for performance recovery and few-step generation (Section 3.4).

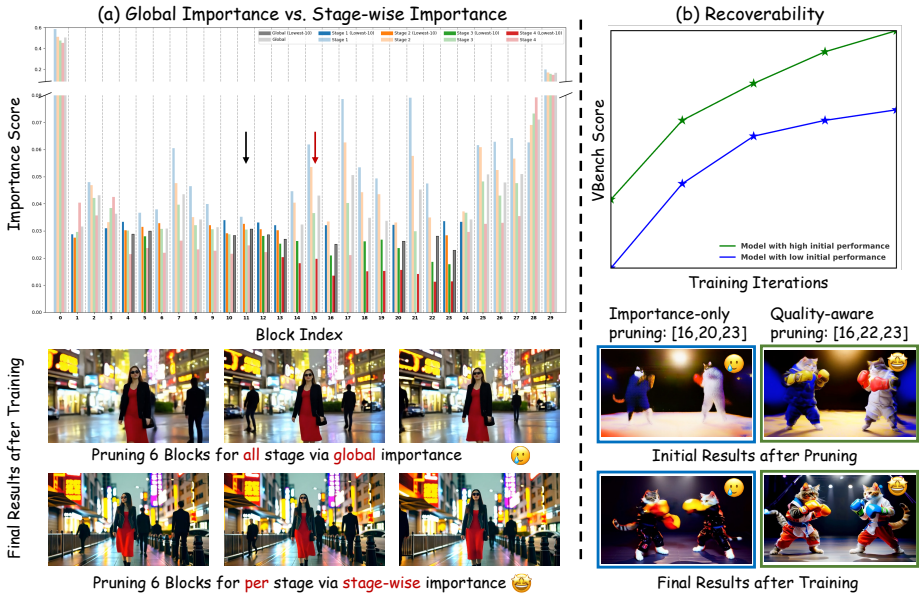
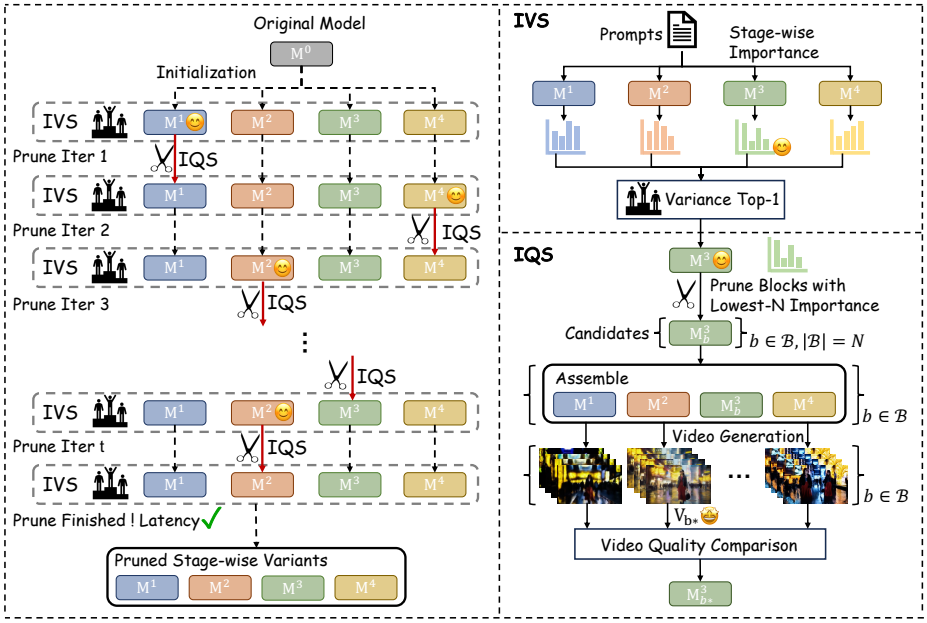


Fig. 2: Illustration of the two pilot studies that motivate the proposed pruning method. (a) Comparing stage-wise and global importance across 30 blocks reveals that global scoring can introduce bias, potentially leading to the unintended pruning of stage-critical blocks (e.g., black arrow, vital for Stages 1, 3, 4) or the retention of stage-redundant ones (e.g., red arrow, redundant for Stage 4). (b) Comparing the recoverability of two pruned models under the same number of training iterations reveals that a model with higher initial performance tends to exhibit better recoverability.

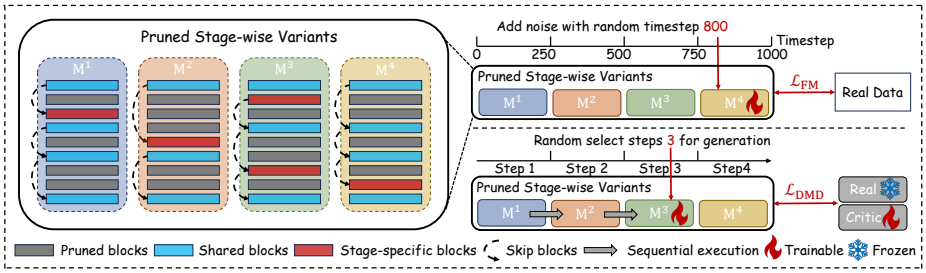
3.1 Pilot Study

To comprehensively analyze the limitations of the existing importance-based pruning paradigm, we conduct two pilot studies focusing on (1) the reliability of global importance and (2) the recoverability of pruned models.

Stage-wise block importance. To compute importance scores, existing paradigms typically rely on step-wise proxies of a block's contribution during inference, such as feature similarity between adjacent blocks at a given denoising step, or loss-sensitive criteria estimated from local curvature (e.g., Hessian-based pruning). A lower score suggests that a block contributes less at that specific step, whereas a higher score implies a larger contribution. These step-wise scores are then averaged over all timesteps to obtain a global importance score for each block, and blocks are pruned accordingly. However, the denoising process can be viewed as solving a PF-ODE along a discretized trajectory, where the model gradually evolves from pure noise to a realistic video sample. This coarse-to-fine evolution naturally consists of several stages that emphasize different aspects of generation (e.g., global structure, fine details, and motion). Therefore, block importance should vary across stages rather than remain constant. To illustrate this, we con-



(a) Stage-Specialized Block Pruning.



(b) Two-phase Training.

Fig. 3: Overview of the proposed stage-aware pipeline. (a) Stage-specialized block pruning. We conduct a hierarchical pruning scheme by iteratively applying IVS and IQS: IVS identifies the stage with the highest block-importance variance, indicating greater pruning potential. Subsequently, IQS evaluates candidate variants at that stage by comparing generation quality, selecting the optimal block to prune in that iteration. (b) Two-phase training for pruned variants. We first fine-tune each variant within its assigned timestep range. Then, we train them using DMD2 to encourage sequential composition for few-step generation. During training, shared blocks are updated and synchronized across variants, while stage-specific blocks are updated independently within each variant.

duct a pilot study on Wan2.1-1.3B [24], as shown in Fig. 2(a). We partition the 48 denoising steps into four stages (12 steps per stage) and compute the stage-wise average importance of each block over 100 text prompts. For comparison, we also compute the global importance scores by averaging over all 48 steps

using the same prompts. We observe that (1) some blocks remain consistently important across multiple, often all, stages, suggesting a core role throughout the denoising process; and (2) other blocks exhibit strong stage dependency, with importance varying substantially across stages and deviating markedly from the globally averaged scores. We further compare the post-training performance of models obtained by global-based pruning and stage-based pruning. For a fair comparison, the global-based baseline removes six blocks according to globally averaged scores, while the stage-based method removes six blocks within each stage according to stage-wise scores, so that the per-step latency is aligned. In addition, the stage-based variants adopt a block-sharing strategy across stages, making the total parameter count comparable to the global-based pruned model. After training, the stage-wise pruned model generates more fine-grained facial details compared to the global-score baseline.

Recoverability of the pruned model. Recoverability is crucial for pruned models, yet it remains underexplored. Existing paradigms typically prune models without quality-feedback guidance. As shown in the previous pilot study, block pruning based on biased importance scores can remove stage-critical blocks, thereby degrading the model's denoising capability at that stage. Intuitively, pruned models with different initial generation quality may exhibit significantly different recoverability after post-training. A pruned model with higher-quality initial generations preserves more diffusion priors from the original model and is therefore easier to recover to competitive performance. In contrast, a lower-quality pruned model may suffer substantial damage to these priors, requiring significantly more high-quality data and training budget for recovery, or even failing to recover under conventional fine-tuning. To illustrate this, we conduct a second pilot study using Wan2.1-1.3B [24], as shown in Fig. 2(b). We compare two models pruned to remove the same number of blocks but exhibiting different initial video generation quality. The first model is obtained by simply removing the three blocks with the lowest importance scores. The second model is chosen via quality feedback among multiple pruned candidates. Specifically, we select three blocks from the five lowest-scoring blocks, yielding ten candidate pruning sets, and choose the one that achieves the best generation quality. After fine-tuning both models with identical training settings and datasets, the model with higher initial quality consistently shows better recoverability.

3.2 Stage-Specialized Block Pruning

Motivated by the two pilot studies, our key insight is that generally important blocks should be retained across all denoising stages, while pruning should be tailored to each stage to remove redundant blocks and preserve stage-specific ones. To achieve this, we propose a novel pruning paradigm, Stage-Specialized Block Pruning (SSBP), which produces stage-specialized variants guided by stage-wise importance. Importantly, some variants reuse a subset of blocks shared across multiple stages, and the shared parameters are stored only once, so the total parameter footprint stays on par with the original model. For each stage, we compute the average per-block importance within that stage and prune accordingly,

yielding a set of stage-specialized variants tailored to different denoising stages. Each variant contains (1) shared blocks across multiple stages that are generally important, and (2) stage-specific blocks specialized for that stage. With parameter sharing, the stage-specialized variants incur a storage cost comparable to the original model, while each stage executes a smaller model footprint, reducing per-step latency. Formally, for a K-stage partition with stage-specialized variants $\{M^i\}_{i=1}^K$, we decompose each variant into a shared part and a stage-specific part:

$$M^i = M_{\text{shr}} \cup M^i_{\text{spec}}, \quad i = 1, \ldots, K. \tag{1}$$

Let $\Theta(\cdot)$ denote the set of model parameters. In SSBP, all parameters are selected from the original model (i.e., $\Theta(M_{\text{shr}}) \subseteq \Theta(M)$ and $\Theta(M^i_{\text{spec}}) \subseteq \Theta(M)$ for all $i \in \{1, \ldots, K\}$). Moreover, shared parameters are stored only once, and the stage-specific parameters are exclusive to each stage:

$$\Theta(M_{\text{shr}}) \cap \Theta(M^i_{\text{spec}}) = \emptyset, \ \forall i, \qquad \Theta(M^i_{\text{spec}}) \cap \Theta(M^j_{\text{spec}}) = \emptyset, \ \forall i \neq j. \tag{2}$$

Therefore, the total stored parameter set is:

$$\Theta(M_{\text{shr}}) \cup \Big(\bigcup_{i=1}^{K} \Theta(M^i_{\text{spec}}) \Big) \ \subseteq \ \Theta(M). \tag{3}$$

Which model to prune: the original or the few-step model? Our method prunes the original model and then performs post-training to recover performance and enable few-step generation for the pruned variants. A straightforward alternative is to prune an already step-distilled few-step model in a stage-wise manner, using the distilled step indices as the stage partition. However, recovering a pruned few-step model is substantially more challenging. Since few-step models operate on a small number of discrete denoising steps, they provide a much coarser approximation to the underlying continuous diffusion trajectory, making flow-matching-based recovery infeasible. In addition, distribution matching alone also struggles to restore the performance of pruned models. In practice, distribution matching assumes that both the teacher and the student retain sufficient diffusion capability, so the student can reliably seek out and align salient modes of the teacher distribution during training. After pruning, however, the student's generative prior is severely degraded, which limits its ability to reach these modes and hinders recovery.

3.3 Hierarchical Selection Mechanism for Iterative Stage-aware Pruning

Diffusion stages involve different generation difficulties and may exhibit varying levels of redundancy. A stage with higher redundancy should be pruned more aggressively, whereas a less redundant stage likely requires larger capacity and should be pruned less. In addition, stage-wise importance estimation is still computed in an averaged manner and may incur stage-wise bias, motivating an additional pruning-block selection mechanism. To address the two aforementioned issues, we propose a hierarchical selection mechanism that performs

Algorithm 1: Hierarchical Selection Mechanism for Iterative Stage-aware Pruning

Input: Original model M; stage partition $\mathcal{T} = \{T^i\}_{i=1}^K$; initial per-stage variants $\{M^i\}_{i=1}^K$; per-stage latency function $L(\cdot)$; target latency L_T; candidate count N.
Output: Pruned stage-specialized variants $\{M^i\}_{i=1}^K$.
while $\sum_{i=1}^K L(M^i) > L_T$ **do**
  `// Compute stage-wise block importance and stage variance`
  for $i \leftarrow 1$ **to** K **do**
    $\mathbf{s}_i \leftarrow \text{StageImportance}(M^i, T^i)$;
    $v_i \leftarrow \text{Var}(\mathbf{s}_i)$;
  `// IVS: select the stage to prune`
  $j \leftarrow \arg\max_{i \in \{1,\dots,K\}} v_i$;
  `// IQS: build` N `candidates by pruning the` N `lowest-importance blocks in stage` j
  $\mathcal{B}_j \leftarrow \text{LowestNBlocks}(\mathbf{s}_j, N)$;
  $\mathcal{C} \leftarrow \emptyset$;
  foreach $b \in \mathcal{B}_j$ **do**
    $M_b^j \leftarrow \text{PruneBlock}(M^j, b)$;
    `// Assemble a full denoising trajectory by replacing the stage-`j` variant with the candidate`
    $\hat{\mathcal{M}}_b \leftarrow \{M^1, \dots, M^{j-1}, M_b^j, M^{j+1}, \dots, M^K\}$;
    $V_b \leftarrow \text{GenerateVideo}(\hat{\mathcal{M}}_b)$;
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(b, M_b^j, V_b)\}$;
  `// Round-robin pairwise ranking with GPT-5.2 on visual fidelity, motion dynamics, and temporal consistency`
  $\mathcal{C} \leftarrow \text{RoundRobinRank}(\mathcal{C}, \text{GPT-5.2})$;
  $(b^\star, M_{b^\star}^j, V_{b^\star}) \leftarrow \mathcal{C}[1]$;
  `// Commit the best candidate for this iteration`
  $M^j \leftarrow M_{b^\star}^j$;
return $\{M^i\}_{i=1}^K$;

pruning iteratively with two coupled components: an Inter-Stage Variance Selector (IVS) and an Intra-Stage Quality Selector (IQS). IVS decides which stage should be pruned, and IQS decides which blocks to prune within that stage. The two selectors interact iteratively until the target latency budget is met, as shown in Fig. 3(a) and Algorithm 1. At each iteration, IVS compares stages by the variance of their block-importance distributions. A larger variance suggests a wider spread in block importance within that stage, indicating higher redundancy and thus greater pruning potential; therefore, IVS prioritizes the stage with the largest variance. Given the selected stage, IQS selects the pruning decision from N candidate pruned variants. Specifically, it prunes each of the N lowest-importance blocks to construct N candidate variants. For each candidate, it runs full denoising with the pruned stage variant and the other stage variants

to generate videos. It then conducts round-robin pairwise comparisons among the N candidates using GPT-5.2 to evaluate visual fidelity, motion dynamics, and temporal consistency. IQS assigns one win to the preferred candidate in each comparison and selects the candidate with the most wins as the pruning decision for this iteration. We set $N = 3$ to balance effectiveness and efficiency.

3.4 Performance Recovery and Few-step Generation

We design a two-phase training pipeline to recover generation quality and enable few-step generation, as shown in Fig. 3(b). In the first phase, we fine-tune each pruned variant using flow matching over the timestep range assigned to its stage. This isolated training does not encourage interactivity among variants. To facilitate the sequential composition of these variants for few-step sampling, we subsequently apply DMD2 [31]. Specifically, we assign each stage-specialized variant to one sampling step and simulate inference-time sampling by chaining the variants across the predefined timesteps to obtain a generated sample. It is then perturbed with noise and fed into both the teacher and the criterion model to predict their scores. The variants are updated based on the difference between these scores. To guarantee that each stage-specialized variant receives a direct learning signal, we stochastically truncate the later portion of the sampling chain during training. Throughout both training phases, shared blocks are updated and synchronized across the variants that include them, whereas stage-specific blocks are updated independently within their respective variants.

4 Experiments

4.1 Implementation

Datasets. We use three datasets in our two-phase training pipeline. In the first phase, we fine-tune each pruned variant on its assigned timestep range using OpenVidHD-0.4M [22] and MagicMotion [18]. OpenVidHD-0.4M contains a wide variety of 1080p video-text pairs spanning diverse scenes, from which we sample 30K pairs. MagicMotion comprises 23K videos curated from Pexels featuring dynamic foreground objects. Together, these two datasets help recover motion and fine-detail modeling and improve temporal consistency. In the subsequent phase, we use DMD2 to train the sequentially composed variants for few-step generation, leveraging text prompts from Vchitect T2V DataVers [4]. This dataset provides high-quality recaptioned text prompts that better satisfy the requirements of video-free DMD2 training.

Training details. We adopt Wan2.1-1.3B as the foundation model, specifically for text-to-video generation. In the fine-tuning phase, we uniformly partition the full denoising trajectory into four stages ($K = 4$) to match a predefined 4-step setting for few-step generation. We fine-tune each variant within its assigned timestep range using a flow-matching loss. These variants are trained for a total of 10K steps with a batch size of 64 and a learning rate of 2×10^{-5}. We then

Table 1: Quantitative comparison with representative text-to-video diffusion models on the VBench [11]. * denotes few-step models trained via DMD2 [31]. SSBP is configured with a 30% acceleration ratio.

Models	Total Score	Quality Score	Semantic Score	Temporal Flickering	Scene	Dynamic Degree	Object Class	Appear. Style	Aesthetic Quality
VideoCrafter2 [1]	80.44	82.20	73.42	98.41	55.29	42.50	92.55	25.13	67.22
LTX-Video [9]	80.00	82.30	70.79	99.34	51.07	54.35	83.45	21.47	59.81
CogVideoX-5B [30]	81.61	82.75	77.04	98.66	53.20	70.97	85.23	24.91	61.98
HunyuanVideo [15]	83.43	85.07	76.88	99.39	54.46	71.94	83.48	22.21	60.28
Wan2.1-1.3B [24]	83.31	85.23	75.65	99.55	41.96	65.19	88.81	21.81	65.46
DMD2-Wan2.1-1.3B*	84.21	85.98	77.09	98.69	45.26	72.50	90.52	21.57	67.90
SSBP-Wan2.1-1.3B	83.52	85.02	77.54	99.15	42.38	73.88	86.93	21.42	66.63

perform DMD2 training for 5K steps with a batch size of 16. In this phase, the original model serves as the teacher, while the fine-tuned variants are used as the student and criterion model. We use a learning rate of 1×10^{-5} for the student and 2×10^{-6} for the criterion model. The criterion model is updated every step, while the student is updated every 5 steps. Both training phases are conducted on 16 MI300 GPUs. The shared blocks are updated and synchronized across the variants that contain them, while stage-specific blocks are updated independently within their corresponding variants. We adopt the feature similarity between adjacent blocks as the block-importance criterion.

4.2 Comparison with State-of-the-Arts

Comparison with leading video diffusion models. We adopt Wan2.1-1.3B as our foundation model. We compare the SSBP-pruned models against leading text-to-video diffusion models. In addition, we distill both foundation models using DMD2 to obtain their few-step variants and include them as additional baselines. Quantitative results are shown in Tab. 1. Both pruned models achieve competitive VBench scores compared to the unpruned few-step variants, while offering lower per-step latency. The pruned models achieve comparable VBench performance to the original models using multiple denoising steps. We further show qualitative results in Fig. 4. The pruned models generate videos with dynamic motion, finer details for foreground objects, and temporal consistency throughout the video.

Comparison with leading pruning methods. We compare the proposed SSBP with two pruning strategies, including L-OBS and TinyFusion. The first is importance-based pruning method, while the latter is a learnable mask-based method. We compare them under 20% and 30% acceleration ratios, adopting Wan2.1-1.3B as the foundation model. For the learnable method, we carefully examine different hyperparameters to optimize the training configuration for learning the pruning masks. For a fair comparison, the models pruned by these methods are trained following the same pipeline as SSBP. The quantitative results are reported in Tab. 2. The model pruned by SSBP achieves higher VBench scores compared to other methods under both acceleration settings.

Fig. 4: Qualitative comparison of SSBP pruned model (30% acceleration) with the unpruned baselines DMD2-Wan2.1-1.3B.

Table 2: Quantitative comparison with state-of-the-art pruning methods.

Method	Acceleration 20%			Acceleration 30%		
	Total	Quality	Semantic	Total	Quality	Semantic
L-OBS [3]	79.62	81.65	71.48	78.53	80.84	69.31
TinyFusion [6]	81.23	83.39	72.58	80.46	82.70	71.43
SSBP	**84.02**	**85.54**	**77.92**	**83.52**	**85.02**	**77.54**

Table 3: Ablation on the number of stages K.

Stage	Acceleration 20%			Acceleration 30%		
	Total	Quality	Semantic	Total	Quality	Semantic
K=1	81.09	83.43	71.74	80.16	82.46	70.95
K=2	82.70	84.60	75.11	82.27	84.19	74.58
K=4	**84.02**	**85.54**	**77.92**	**83.52**	**85.02**	**77.54**

Table 4: Ablation on the effectiveness of the proposed stage-aware pruning (30% acceleration).

Method	Total	Quality	Semantic
One-stage	78.07	80.53	68.24
Stage-wise	80.91	83.01	72.50
Stage-wise + IVS	82.44	84.27	75.14
Stage-wise + IVS + IQS	**83.52**	**85.02**	**77.54**

Table 5: Ablation on the number of candidates N used in the IQS.

#Candidates	Total	Quality	Semantic	#GPT-Times
N=1	82.44	84.27	75.14	0×20
N=3	83.52	85.02	77.54	3×20
N=5	83.63	85.28	77.02	10×20
N=7	83.60	85.16	77.35	21×20

4.3 Ablation Study

Ablation on the number of stages. We set the default number of partitioned stages to $K = 4$, aligned with the few-step generation schedule (i.e., 4 steps). We further examine smaller numbers of stages, including $K = 1$ and $K = 2$. For $K = 1$, we disable stage partitioning and perform global pruning guided by IQS until the target acceleration ratio is achieved. For $K = 2$, we prune two stage-specialized variants, each responsible for two sampling steps. We conduct ablations under 20% and 30% acceleration ratios. As shown in Tab. 3, a finer stage partition (larger K) introduces more stage-specific variants, which in turn improves performance, demonstrating the effectiveness of stage specialization.

Ablation on the components of SSBP. To validate the mechanisms of SSBP, we progressively introduce (1) stage-wise partitioning, (2) IVS, and (3) IQS under the same acceleration ratio of 30%. For the one-stage baseline, we prune the model according to a globally averaged importance score to meet the target acceleration ratio. For stage-wise pruning, we apply the same pruning ratio as the baseline to each of the four variants, ensuring comparable per-step la-

tency. As shown in Tab. 4, stage-wise pruning outperforms one-stage pruning on VBench. This indicates that stage-wise averaging is tailored to each stage and mitigates the bias introduced by global averaging. With the hierarchical selection mechanism, each variant is pruned adaptively within its stage, better preserving the initial generation quality. Consequently, the resulting pruned model achieves better performance after training.

4.4 Ablation on the Number of Candidates N Used in the IQS.

We further provide comparison results with different numbers of candidates, as shown in Tab. 5. Increasing N allows IQS to explore a larger candidate set, but also introduces higher search cost. Specifically, IQS performs round-robin pairwise comparisons among all candidate variants, so the number of comparisons grows quadratically with N. For example, when $N = 8$, IQS performs $\binom{7}{2} = 21$ pairwise comparisons for each prompt, leading to a total of 21×20 GPT-5.2 calls per pruning iteration on a 20-prompt set. The experimental results also show that a larger N does not consistently lead to better performance. We conjecture that this is because stage-wise averaged importance already alleviates the estimation bias, so IQS does not require a large candidate set to further correct it. Therefore, we set $N = 3$ to balance search efficiency and the performance of the pruned model.

5 Conclusion

In this paper, we present Stage-Specialized Block Pruning (SSBP), a novel pruning paradigm for accelerating video diffusion models. Our key insight lies in adaptive pruning tailored for different denoising stages, enabling each variant to focus on its specialized capabilities. To achieve this, we propose a Hierarchical selection mechanism comprising IVS and IQS for iterative stage-wise pruning. IVS identifies the target stages for pruning, while IQS selects the redundant blocks within those stages. The pruned stage-wise variants comprise both shared and stage-specific blocks, maintaining fewer total parameters than the unpruned model. Furthermore, we introduce a two-phase training pipeline, first fine-tuning each variant within its assigned timestep range and then training via DMD2 to facilitate sequential composition for few-step generation. Extensive experiments show that our method accelerates representative video diffusion models, while maintaining competitive VBench performance. Our pruning paradigm is also compatible with other orthogonal acceleration methods, such as efficient block replacement.

6 Limitations and Future Work

SSBP achieves a favorable trade-off between efficiency and generation quality by introducing stage-specialized variants. However, the current pruning granularity

remains at the block level, which is relatively coarse. As the pruning ratio becomes more aggressive, block-level pruning may remove important blocks that are shared across multiple stages. Once these shared blocks are discarded, the resulting performance drop is difficult to recover through efficient training alone. A promising direction for future work is to explore intra-block pruning within the stage-specialized framework. Although prior works have explored finer-grained pruning, they remain stage-agnostic. We believe our stage-wise design offers a new perspective for future research. In addition, our method adopts a fixed and manually defined stage partition. While this design is simple and effective in practice, the optimal partition may vary across architectures, sampling schedules, and acceleration targets. Future work could therefore investigate adaptive stage partitioning strategies that automatically determine the number of stages and their boundaries.

References

1. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In: CVPR (2024)
2. Chen, J., Zhao, Y., Yu, J., Chu, R., Chen, J., Yang, S., Wang, X., Pan, Y., Zhou, D., Ling, H., et al.: Sana-video: Efficient video generation with block linear diffusion transformer. CoRR **abs/2509.24695** (2025)
3. Dong, X., Chen, S., Pan, S.: Learning to prune deep neural networks via layer-wise optimal brain surgeon. NIPS (2017)
4. Fan, W., Si, C., Song, J., Yang, Z., He, Y., Zhuo, L., Huang, Z., Dong, Z., He, J., Pan, D., et al.: Vchitect-2.0: Parallel transformer for scaling up video diffusion models. CoRR **abs/2501.08453** (2025)
5. Fan, X., Qiu, Z., Wu, Z., Wang, F., Lin, Z., Ren, T., Lin, D., Gong, R., Yang, L.: Phased dmd: Few-step distribution matching distillation via score matching within subintervals. In: CVPR (2026)
6. Fang, G., Li, K., Ma, X., Wang, X.: Tinyfusion: Diffusion transformers learned shallow. In: CVPR (2025)
7. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. In: NIPS (2023)
8. Ge, M., Isobe, T., Jia, X., Sun, Y., Yang, Z., Wang, W., Zhou, D., Li, D., Lu, H., Barsoum, E.: Ego-inbetween: Generating object state transitions in ego-centric videos. In: CVPR (2026)
9. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al.: Ltx-video: Realtime video latent diffusion. CoRR **abs/2501.00103** (2024)
10. Hassibi, B., Stork, D., Wolff, G.: Optimal brain surgeon: Extensions and performance comparisons. NIPS (1993)
11. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR. pp. 21807–21818 (2024)
12. Isobe, T., Cui, H., Zhou, D., Ge, M., Li, D., Barsoum, E.: Amd-hummingbird: Towards an efficient text-to-video model. CoRR **abs/2503.18559** (2025)

13. Karnewar, A., Korzhenkov, D., Lelekas, I., Karjauv, A., Fathima, N., Xiong, H., Vaidyanathan, V., Zeng, W., Esteves, R., Singhal, T., et al.: Neodragon: Mobile video generation using diffusion transformer. CoRR **abs/2511.06055** (2025)
14. Kim, B.K., Song, H.K., Castells, T., Choi, S.: Bk-sdm: Architecturally compressed stable diffusion for efficient text-to-image generation. In: ICML (2023)
15. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. CoRR **abs/2412.03603** (2024)
16. Li, J., Feng, W., Fu, T.J., Wang, X., Basu, S., Chen, W., Wang, W.Y.: T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. NIPS (2024)
17. Li, J., Long, Q., Zheng, J., Gao, X., Piramuthu, R., Chen, W., Wang, W.Y.: T2v-turbo-v2: Enhancing video generation model post-training through data, reward, and conditional guidance design. CoRR **abs/2410.05677** (2024)
18. Li, Q., Xing, Z., Wang, R., Zhang, H., Dai, Q., Wu, Z.: Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. CoRR **abs/2503.16421** (2025)
19. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al.: Open-sora plan: Open-source large video generation model. CoRR **abs/2412.00131** (2024)
20. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. CoRR **abs/2310.04378** (2023)
21. Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., Chen, W.: Shortgpt: Layers in large language models are more redundant than you expect. CoRR **abs/2403.03853** (2024)
22. Nan, K., Xie, R., Zhou, P., Fan, T., Yang, Z., Chen, Z., Li, X., Yang, J., Tai, Y.: Openvid-1m: A large-scale high-quality dataset for text-to-video generation. CoRR **abs/2407.02371** (2024)
23. Sreenivas, S.T., Muralidharan, S., Joshi, R., Chochowski, M., Mahabaleshwarkar, A.S., Shen, G., Zeng, J., Chen, Z., Suhara, Y., Diao, S., et al.: Llm pruning and distillation in practice: The minitron approach. CoRR **abs/2408.11796** (2024)
24. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. CoRR **abs/2503.20314** (2024)
25. Wang, X., Zhang, S., Zhang, H., Liu, Y., Zhang, Y., Gao, C., Sang, N.: Videolcm: Video latent consistency model. CoRR **abs/2312.09109** (2023)
26. Wu, Y., Chen, Z., Wang, H., Xu, D.: Individual content and motion dynamics preserved pruning for video diffusion models. CoRR **abs/2411.18375** (2024)
27. Wu, Y., Li, Y., Kag, A., Skorokhodov, I., Menapace, W., Ma, K., Sahni, A., Hu, J., Siarohin, A., Sagar, D., et al.: Taming diffusion transformer for efficient mobile video generation in seconds. CoRR **abs/2507.13343** (2025)
28. Wu, Y., Zhang, Z., Li, Y., Xu, Y., Kag, A., Sui, Y., Coskun, H., Ma, K., Lebedev, A., Hu, J., et al.: Snapgen-v: Generating a five-second video within five seconds on a mobile device. In: CVPR (2025)
29. Xie, E., Chen, J., Zhao, Y., Yu, J., Zhu, L., Wu, C., Lin, Y., Zhang, Z., Li, M., Chen, J., et al.: Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. CoRR **abs/2501.18427** (2025)
30. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. CoRR **abs/2408.06072** (2024)

31. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. NIPS (2024)
32. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: CVPR (2024)
33. Zhang, D., Li, S., Chen, C., Xie, Q., Lu, H.: Laptop-diff: Layer pruning and normalized distillation for compressing diffusion models. CoRR **abs/2404.11098** (2024)
34. Zhang, J., Wang, H., Jiang, K., Yang, S., Zheng, K., Xi, H., Wang, Z., Zhu, H., Zhao, M., Stoica, I., et al.: Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. CoRR **abs/2509.24006** (2025)
35. Zhang, P., Chen, Y., Huang, H., Lin, W., Liu, Z., Stoica, I., Xing, E., Zhang, H.: Vsa: Faster video diffusion with trainable sparse attention. CoRR **abs/2505.13389** (2025)
36. Zhang, Y., Jin, E., Dong, Y., Khakzar, A., Torr, P., Stegmaier, J., Kawaguchi, K.: Effortless efficiency: Low-cost pruning of diffusion models. CoRR **abs/2412.02852** (2024)
37. Zheng, K., Wang, Y., Ma, Q., Chen, H., Zhang, J., Balaji, Y., Chen, J., Liu, M.Y., Zhu, J., Zhang, Q.: Large scale diffusion distillation via score-regularized continuous-time consistency. CoRR **abs/2510.08431** (2025)
38. Zhu, J., Wang, H., Su, M., Wang, Z., Wang, H.: Obs-diff: Accurate pruning for diffusion models in one-shot. CoRR **abs/2510.06751** (2025)