ASDSV: Multimodal Generation Made Efficient with Approximate Speculative Diffusion and Speculative Verification

Kaijun Zhou, Xingyu Yan, Xingda Wei, Xijun Li, Jinyu Gu*
School of Computer Science, Shanghai Jiao Tong University

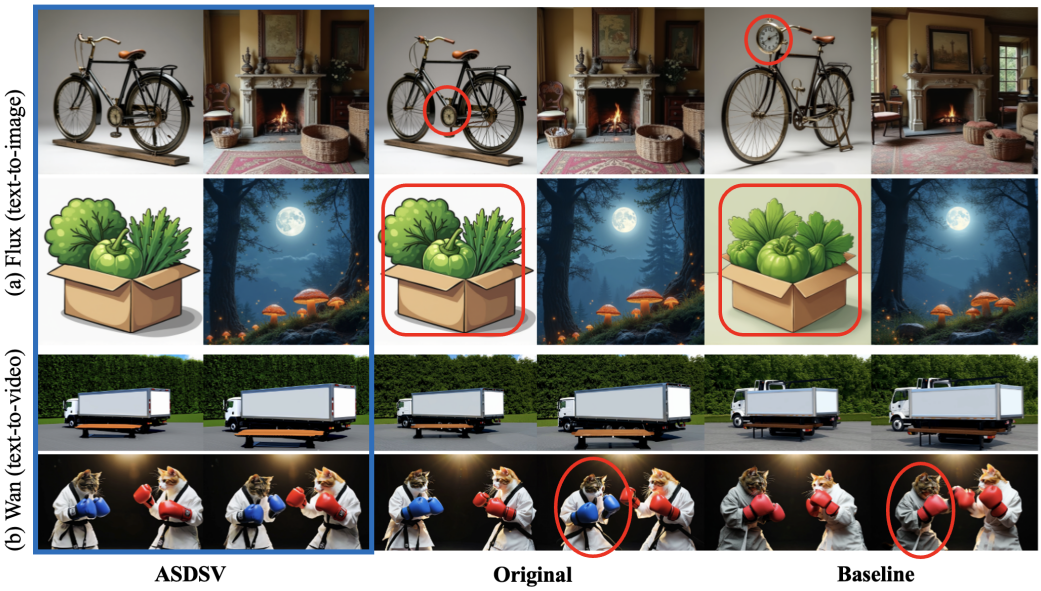


Figure 1: Visual comparisons of different methods on various generation tasks. From left to right: Approximate Speculative Diffusion with Speculative Verification (ASDSV), original model, Baseline (TeaCache). The red boxes highlight regions that differ significantly from the original image/video.

Abstract

Diffusion in transformer is central to advances in high-quality multimodal generation but suffer from high inference latency due to their iterative nature. Inspired by speculative decoding's success in accelerating large language models, we propose *Approximate Speculative Diffusion with Speculative Verification (ASDSV)*, a novel method to enhance the efficiency of diffusion models. Adapting speculative execution to diffusion processes presents unique challenges.

First, the substantial computational cost of verifying numerous speculative steps for continuous, high-dimensional outputs makes traditional full verification prohibitively expensive. Second, determining the optimal number of speculative steps K involves a trade-off between potential acceleration and verification success rates.

To address these, ASDSV introduces two key innovations: 1) A *speculative verification* technique, which leverages the observed temporal correlation between draft and target model outputs, efficiently validates K speculative steps by only checking the alignment of the initial and final states, significantly reducing verification

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

overhead. 2) A *multi-stage speculative strategy* that adjusts K according to the denoising phase—employing smaller K during volatile early stages and larger K during more stable later stages to optimize the balance between speed and quality. We apply ASDSV to state-of-the-art diffusion transformers, including Flux.1-dev for image generation and Wan2.1 for video generation. Extensive experiments demonstrate that ASDSV achieves up to 1.77×-3.01× speedup in model inference with a minimal 0.3%-0.4% drop in VBench score, showcasing its effectiveness in accelerating multimodal diffusion models without significant quality degradation. The code will be publicly available once the acceptance of the paper.

1 Introduction

Multimodal generation has achieved rapid progress in recent years, with diffusion models emerging as a dominant choice for producing high-quality, semantically coherent outputs across modalities such as images, videos, and text [7, 15]. As these models scale in capacity and complexity—exemplified by recent breakthroughs like Sora [3], Flux [18], and Wan2.1 [37]—their generative capabilities have reached unprecedented levels.

However, the inference latency of diffusion models remains a critical performance challenge due to the inherently iterative nature of the denoising process [1]. Each inference requires tens to hundreds of sequential denoising steps and each step is computationally expensive. This challenge is particularly pronounced in high-resolution image synthesis or long video generation, where the cumulative latency poses significant barriers to real-time applications and scalable deployment [41, 22].

To address this challenge, previous works have explored three primary directions: distillation, architectural modifications, and caching-based strategies. Distillation methods [29] compress pretrained models into faster variants but demand substantial computational resources for retraining and often degrade output quality. Architectural optimizations, such as attention compression and sparse attention mechanisms [43, 38], reduce per-step computation but also require retraining and are tightly coupled with model designs, limiting generalizability. Caching-based approaches [45, 42, 24] investigate the similarity between adjacent steps in the denoising process and propose skip-stepping strategies to reduce the number of denoising steps. Yet, this aggressive strategy may lead to a noticeable degradation in generation quality, as shown in Figure 1 (differences between Original and Baseline).

In language model inference, speculative decoding [19] has emerged as a powerful paradigm for latency reduction. It typically employs a lightweight draft model to predict K tokens sequentially, which are then verified in parallel by the target model in a single inference batch, achieving speedups without compromising the output results. Inspired by its success, we explore adapting speculative decoding to diffusion models. A key observation motivates this direction: multimodal generation models often have corresponding draft models readily available, either as lightweight variants from the same family (e.g., Wan2.1-1.3B for Wan2.1-14B [37], Stable-Diffusion 3.5 medium and large variants [9]) or through community-driven model compression efforts (e.g., Flux-lite for Flux.1-dev [18]).

However, adapting this strategy to diffusion models presents unique challenges. First, efficient verification is a significant hurdle. The continuous and high-dimensional nature of image and video data demands substantially more computational resources and memory for verification compared to discrete token sampling in language models. This can restrict batch sizes and has limited existing speculative approaches in diffusion [2] to low-resolution tasks such as CIFAR-10 [17]. To overcome this, we introduce a **approximate speculative verification** strategy. Instead of verifying all K intermediate outputs from the draft model, our method only checks the initial and final speculative outputs. If both align with the target model's outputs within a defined tolerance, all intermediate steps are accepted. The insight for this efficient verification is that the outputs from draft and target models tend to exhibit similar variations across adjacent denoising steps (detailed in Sec3.1).

Second, determining the optimal number of speculative steps, K, involves a critical trade-off. A larger K could theoretically yield greater acceleration, but it also increases the likelihood of the draft model's predictions diverging from the target model, leading to higher verification failure rates. To address this, we propose a **multi-stage speculative strategy** that adopts multiple K values based on the characteristic dynamics of diffusion models: early denoising steps exhibit rapid output changes, while later steps show gradual convergence [45, 10]. Consequently, our strategy employs a smaller

K during the volatile early stages to minimize error accumulation and a larger K in the more stable later stages to maximize performance.

In all, our proposed system, which we term *Approximate Speculative Diffusion with Speculative Verification (ASDSV)*, begins approximate speculative diffusion after letting the target model complete the first N denoising steps. The initial denoising steps exhibit large variations, making them unsuitable for speculative execution. Then, ASDSV enters an interleaved execution pattern: the draft model executes K steps, speculative verification is performed, and the target model executes a single step. This approach effectively reduces total computations while striving to preserve generation quality.

We apply ASDSV to a wide range of multimodal models, including two state-of-the-art diffusion transformers: Flux.1-dev [18] (image generation) and Wan2.1 [37] (video generation). Our extensive experiments demonstrate that ASDSV achieves up to $1.77\times$-$3.01\times$ speedup on model inference speed over the vanilla execution, while only incurring a minimal performance drop of 0.3%-0.4% in VBench score.

2 Related Work

2.1 Diffusion Model

Multimodal generation has achieved remarkable progress, with diffusion models continuously breaking state-of-the-art records [3, 12]. Diffusion models consist of an iterative process of denoising steps, where inference gradually removes noise from the initial data sample [15, 31]. By using attention mechanisms [34] and various optimization algorithm designs [25, 26, 23], diffusion in transformer (DiT) [30] models demonstrate excellent performance in text-to-image and text-to-video generation. State-of-the-art (SOTA) models have evolved from Pixart-Alpha (1.2B) [4] to Flux (12B) [18] in image generation; from Open-Sora (1.1B) [46] to Wan2.1 (14B) [37] and Step-Video (20B) [27] in video generation.

2.2 Diffusion Model Acceleration

Distillation and Architectural Optimization. Besides distillation methods [32, 6, 28] which requires substantial computational resources to re-train, many acceleration methods focus on optimizing the attention mechanism to reduce the computational cost. DistFastAtten [43] introduces spatial and temporal window attention with residual sharing, while Efficient-vDiT [8] and Sparse VideoGen [38] leverage the sparse properties of 3D full-attention in video generation to reduce the number of tokens involved in attention computation. Yet, such architectural optimizations either requires extra training or are tightly coupled with specific model designs.

Cache. Since the outputs of diffusion models after each denoising step exhibit strong similarity, cache-based methods have been proposed to reduce the number of denoising steps [42, 45, 24]. PAB [45] uses a step-skipping strategy (fixed-step) to skip some of the denoising steps. The blind step-skipping strategy may lead to performance degradation. TeaCache [24] designs a cache indicator to determine when to reuse the cached results (i.e., the output of the previous step) based on the similarity of the inputs of two adjacent steps, which avoids fixed-step skipping. However, directly reusing the cached outputs would still cause performance degradation, as shown in experiments.

Parallelization. Parallelization methods distributing computations across multiple GPUs, can also be effective for accelerating diffusion models (e.g., PipeFusion [11], DistriFusion [20], Classifier-Free Guidance Parallelism [10]) and orthogonal to our work.

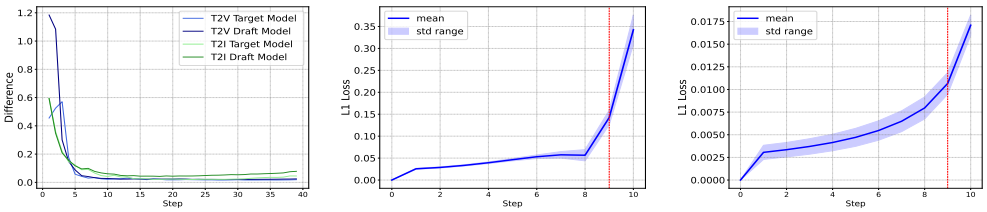
2.3 Speculative Decoding

To reduce the latency of autoregressive language model inference, Speculative Decoding introduces a Draft-then-Verify decoding paradigm [39, 40, 19]. It needs to train a small model (draft model) for the large language model (target model). For decoding, it first efficiently generates multiple tokens by the draft model and then verifies all these tokens in parallel (in one batch) using the target model. Thereby, the target model inferences once but generates multiple tokens, which is faster than the original autoregressive decoding.

One prior work, Speculative Diffusion [2], make the first attempt to extend Speculative Decoding to diffusion models. However, it is only applied to low-resolution image-generation tasks (32×32) [17], as the exsiting verification process is too computationally expensive to be efficiently done for high-resolution tasks. Notebly, the target model needs to generate all candidate steps in one batch for quickly verifying the draft steps, which is impractical for SOTA multimodal models [3, 37, 27].

3 Method

3.1 Approximate Speculative Diffusion with Speculative Verification



(a) (b) (c)

Figure 2: (a) shows strong temporal correlation between draft and target models in the diffusion process. (b) and (c) shows similar trends patterns of verification results: early steps show gradual followed by an abrupt divergence in the few last steps of the verification process.

Our design of ASDSV is motivated by two observations.

(1) The draft models for SOTA multimodal generation models are off-the-shelf, facillitating speculative diffusion. First, as multimodal generative models typically exhibit large parameter sizes and high computational demands, they are usually released with variants of varying sizes to accommodate diverse hardware capabilities. For examples, Wan2.1-1.3B for Wan2.1-14B and Stable-Diffusion 3.5's medium and large variants. Second, the community-developers also provide a rich pool of draft models through model compression techniques for resource-constrained environments.

(2) The output trends of draft and target models show temporal correlation. The *output trend* depicts that the differences between adjacent steps in the denoising process of a diffusion model, while difference between two steps is defined as L1 loss between the outputs of them.

$$\text{L1\_loss}(D_i, T_i) = \text{mean}(\|D_i - T_i\|)$$

D_i and T_i denote the predicted outputs from the draft and target models at step i, respectively. As shown in Figure 2(a), the output trends of draft and target models are close to each other. This refers to our key empirical observation: the strong temporal correlation between draft and target models in the diffusion process. Specifically, the magnitude of change of the draft model's output and the target model's output between consecutive steps is highly similar (i.e., $||D_i - D_{i-1}||_1 \approx ||T_i - T_{i-1}||_1$) after the first N initial steps. This observed characteristic holds for both text-to-image (T2I) and text-to-video (T2V) generation models.

For any adjacent K denoising steps for the draft and target models, we mark their outputs as D_N, D_{N+1}, ..., D_{N+K}, and T_N, T_{N+1}, ..., T_{N+K}, respectively. Since the output trends of draft and target models are close to each other, if the differences (L1_loss) between D_{N+1} and T_{N+1} and between D_{N+K} and T_{N+K} are both smaller than a threshold, we can speculatively conclude that all the differences between intermediate D_{N+i} and T_{N+i} are also within the threshold. Thereby, unlike traditional speculative decoding that verifies every output (T_{N+1}, ..., T_{N+K}) of the draft model, we propose a *speculative verification* strategy which only verifies the output of the draft model at the beginning and end of the sequence (T_{N+1} and T_{N+K}), which significantly reduces the verification cost for diffusion models. Here's a formalized expression for speculative verification:

$$D_{N+1} \leftarrow M_q(T_N), \quad D_{N+i} \leftarrow M_q(D_{N+i-1}), i = 2, \ldots, K$$

$$T_{N+1}, T_{N+K} \leftarrow M_p(D_{N+1}), M_p(D_{N+K}), \quad M_p \text{ and } M_q \text{ has strong temporal correlation}$$

$$\text{If L1\_loss}(D_{N+1}, T_{N+1}) \leq \delta \text{ and L1\_loss}(D_{N+K}, T_{N+K}) \leq \delta,$$

$$\text{then accept } D_{N+1}, \ldots, D_{N+K}, \text{ else reject and generate from } T_{N+2}$$

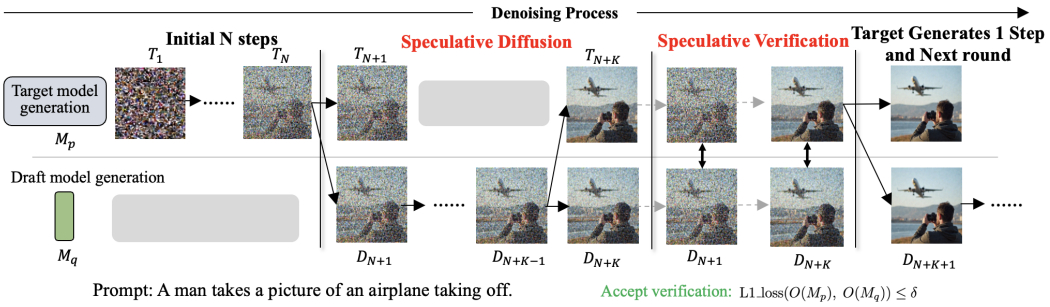


Figure 3: Overview of ASDSV compared with vanilla speculative diffusion. After Ntarget model steps, ASDSV sequentially samples K steps within draft model, and the target model verifies the noises by comparing the first and last step's noises. In contrast, vanilla speculative diffusion verifies the noise of each step.

Workflow. Figure 3 illustrates the workflow of a denoising process with ASDSV. The initial N steps are generated by the target model without any speculative steps, where the input for each step consists of the timestep embedding at the step and the output image of the previous step. After that, the repeated Speculative Diffusion and Speculative Verification process starts.

For each round of Speculative Diffusion, the draft model sequentially samples K steps, where the initial step's input image comes from the target model's output of the previous round; the target model only executes for the first step and the last step (used for verification), where the last step's input image comes from the draft model's output (D_{N+k}). The following Speculative Verification will calculate two differences between D_{N+1} and T_{N+1}, and between D_{N+k} and T_{N+k}. The verification succeeds if both differences are smaller than a threshold (a hyper-parameter, δ). If so, next round of Speculative Diffusion and Speculative Verification starts. Otherwise, the target model generates T_{N+2} and then a new round starts.

Batched Pipeline Optimization. As shown in Figure 3, the target model needs to generate the K step as the last step of the current round's verification process, and the $K+1$ step as the first step of subsequent round. If the verification succeeds, T_{N+K} is similar with D_{N+K}, so the target model can generate T_{N+K+1} based on D_{N+K}. This allows for pipelining the current round's verification process and the next round's generation process in parallel. Thereby, ASDSV batches the draft sampling results D_{N+K-1} and D_{N+K} as the input of the target model, which then generates T_{N+K} and T_{N+K+1} in parallel.

3.2 Multi-Stage Speculative Strategy

According to Figure 2(a), we can also observe that during the denoising process, the differences between adjacent steps are big in the early stages and small in the later stages, which is an intrinsic property of diffusion models [45, 24]. If we choose a large K value (speculative steps), the verification failure probability will be high in the early stages, leading to more wasted computation. In contrast, if we choose a small K value, the verification failure probability will remain low in different stages, but the acceleration ratio will be low. Therefore, unlike traditional speculative decoding approaches that employ a fixed number of speculative steps K throughout the entire decoding process, we propose a multi-stage speculative strategy which chooses different K values for different stages during the denoising process.

Specifically, ASDSV uses a three-stage strategy. Stage-0: The initial N steps is the warmup stage which is used to establish the foundation for high-quality generation. Stage-1: The subsequent warmup stage employs a smaller γ_1 as K value, to preserve a high success rate of verification during the early stage. Stage-2: The main generation stage employs a larger γ_2 as K value, to achieve a higher acceleration ratio. Such a strategy can achieve an optimal balance between generation speed and generation quality.

We choose the hyperparameters based on the step-metric results in both image and video scenarios. We illustrate with a red vertical line in Figure 2 (b) and (c), the verification error curve exhibits a sharp

increase at the 9-th step. This inflection point shows that the draft model's outputs are accumulating errors and starting to diverge significantly from the target model's. Therefore, it represents the appropriate moment to halt the drafting phase. So we set γ_2 as 9 for Stage-2 to achieve both high success rate of verification and high acceleration ratio. More details about other stages are provided in the Appendix A.3.

3.3 Walltime Improvement

We denote the draft model as M_q and the target model as M_p. Let γ_1 and γ_2 be the number of speculative steps in the Stage-1 and Stage-2, respectively. The running time of one step of M_q is T_{M_q} and that of M_p is T_{M_p}. Let c, the *cost coefficient*, be the ratio of T_{M_q} to T_{M_p}. The number of steps in Stage-0 is denoted as N. Assuming Stage-1 and Stage-2 respectively complete α_1 and α_2 speculative rounds and all verifications are successful, ASDSV can achieve a maximum speedup as follows.

$$N + \alpha_1\gamma_1 + \alpha_2\gamma_2 = \text{Total steps} \tag{1}$$

$$\text{Vanilla Diffusion: Total steps} \times T_{M_p} \tag{2}$$

$$\text{ASDSV: } NT_{M_p} + T_{M_q}(\text{Total steps} - N) + T_{M_p}(1 + \alpha_1 + \alpha_2) \tag{3}$$

Note that M_p generates the last step of the current round and the first step of the next round in parallel for each round, except for the first step of the first round. Therefore, in ASDSV, the cost for M_p is $NT_{M_p} + T_{M_p}(1 + \alpha_1 + \alpha_2)$. The theoretical upper bound of speedup is:

$$\frac{N + \alpha_1\gamma_1 + \alpha_2\gamma_2}{N + 1 + \alpha_1(c\gamma_1 + 1) + \alpha_2(c\gamma_2 + 1)} \tag{4}$$

We also provide the expected speedup of ASDSV and its proof in Appendix A.5.

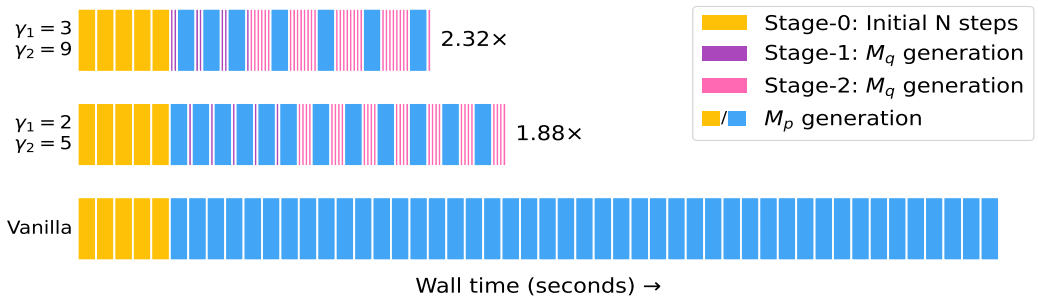


Figure 4: A simplified trace diagram to show ASDSV under different settings on Wan2.1 using 50 steps. The purple blocks and pink blocks represent steps where results are generated by the draft model in Stage-1 and Stage-2, respectively, N is set to 10% of total steps. The first row shows ASDSV with $\gamma_1 = 3$ and $\gamma_2 = 9$, the second row shows ASDSV with $\gamma_1 = 2$ and $\gamma_2 = 5$, and the third row shows vanilla diffusion.

Figure 4 presents the concrete theoretical upper bound of speedup of ASDSV under different settings. However, since there can be verification failures in the speculative steps, the theoretical upper bound is hard to achieve. Furthermore, in our experiments, c was 0.60 for text-to-image models and 0.19 for text-to-video models. In contrast, the cost coefficient c is much smaller in large language models (often less than 0.05 [19]). As this work focuses on making speculative diffusion effective in SOTA multimodal generation models, the exploration of optimal draft model is orthogonal.

4 Experiments

4.1 Settings

Models and Baselines. We evaluate ASDSV on two representative diffusion models: Flux.1-dev [18] for image generation tasks and Wan2.1 [37] for video generation tasks. We compare ASDSV with two state-of-the-art studies on accelerating diffusion inference: Teacache [24] (cache-based method)

for both image and video generation, and Sparse VideoGen [38] (architectural optimization) for video generation acceleration.

Metrics and Datasets. We evaluate ASDSV and other baselines on two aspects: generation quality and generation efficiency. For image generation with Flux.1-dev, we evaluate the generation quality using Frechet Inception Distance (FID) [14], Inception Score (IS) [33], Learned Perceptual Image Patch Similarity (LPIPS) [44], Peak Signal-to-Noise Ratio (PSNR) [35], and CLIP score [13]. Each comparison method generates 10k images using the COCO Captions 2014 dataset [5]. FID, LPIPS and PSNR are metrics for similarity assessment, i.e., how different the generated images are from the ground truth (the output of the original model). CLIP score measures the semantic alignment between the input prompt and the generated images and IS measures the diversity of the generated images.

For video generation with Wan2.1, like prior studies [24], we evaluate the generation quality using VBench Score [16] to evaluate the visual quality of the generated videos, LPIPS, PSNR and Structural Similarity Index Measure (SSIM) [36] to evaluate the similarity between the generated videos and the ground truth. Each comparison method generates 2k videos using the VBench dataset [16].

For generation efficiency evaluation across both models, we use Floating Point Operations (FLOPs) and inference latency as metrics.

Implementation Details. We implement two variants of our method: ASDSV-fast and ASDSV-slow. The ASDSV-slow variant performs the speculative verification and samller K for the Stage-1 speculative step to achieve better visual quality. The ASDSV-fast variant skips the verification entirely, assuming the draft samples pass the verification, and uses the same K for the Stage-1 and Stage-2 speculative steps to achieve higher efficiency, which is a more aggressive speculative diffusion strategy.

For both image and video generation, we set the total number of denoising steps to 50. For text-to-image generation, we use Flux-SVD[21] as the draft model. For ASDSV-slow, we set $\gamma_1 = 3$, $\gamma_2 = 9$, and warmup ratio to 15%. For ASDSV-fast, we use $\gamma_1 = \gamma_2 = 9$. We set initial steps N to 8% of total steps for both variants and a verification threshold (δ) of 0.02. For text-to-video generation, we use Wan2.1-1.3B as the draft model. For ASDSV-slow, we set $\gamma_1 = 2$, $\gamma_2 = 9$, and warmup ratio to 25%. For ASDSV-fast, we use $\gamma_1 = \gamma_2 = 9$. We set initial steps N to 10% of total steps for the fast variant and 15% for the slow variant with (δ) 0.2.

As for the baselines, we use the same settings as in the original papers: Teacache [24] threshold is 0.25 for slow and 0.6 for fast in text-to-image generation, and 0.14 for slow and 0.2 for fast in text-to-video generation. Sparse VideoGen [38] is used with sparsity 0.10 for fast and 0.20 for slow, without enabling custom kernel and FP8 quantization. We measure the latency per sample on a single NVIDIA A800 GPU using Pytorch 2.6.0 and CUDA 12.4.

4.2 Main Results

Comparison with Draft Models. Besides the above-mentioned baselines, we also compare ASDSV with using the different draft models for multimodal generation, which can be regarded as distillation-based methods. As provided in the Appendix A.1, the result is that ASDSV consistently outperforms all draft models, which also shows the effectiveness and broad applicability of ASDSV across diverse compression techniques.

Results of Evaluation Metrics. Table 1 and 2 present the quantitative evaluation of efficiency and visual quality using the evaluation metrics described in Section 4.1. Compared to both cache-based and architectural optimization methods, ASDSV delivers better visual quality and enhanced generation efficiency across different modalities, base models, resolutions and video length.

In Table 1 (text-to-image), ASDSV-slow achieves the best visual similarity metrics (FID, LPIPS, PSNR) and attaining nearly $1.3\times$ speedup. Although ASDSV-fast achieves the lowest FLOPs count, it doesn't attain the minimum latency. This discrepancy arises because the quantized draft model doesn't reach optimal acceleration on A800 GPUs. On hardware supporting efficient low-precision computation (e.g., 50-series GPUs), our approach would likely achieve ideal acceleration, fully realizing its theoretical efficiency advantages.

Table 1: Efficiency and visual quality comparison of ASDSV and other baselines on image generation. Flux-SVD is the draft model in ASDSV.

Method	Efficiency			Visual Quality				
	FLOPs (T) ↓	Latency (s) ↓	Speedup ↑	FID ↓	CLIP ↑	LPIPS ↓	IS ↑	PSNR ↑
FLUX.1-dev (512×512)								
FLUX.1-dev ($T = 50$)	1082.5	9.18	1×	-	31.14	-	26.32	-
TeaCache-fast	330.4	**4.25**	2.16×	8.16	**31.34**	0.30	25.66	28.89
TeaCache-slow	566.8	5.73	1.6×	4.33	31.20	0.14	26.7	31.67
ASDSV-fast	**206.3**	6.25	1.47×	4.11	31.15	0.11	26.66	32.95
ASDSV-slow	292.3	7.3	1.26×	**3.90**	31.15	**0.10**	26.52	**33.14**

Despite not achieving the highest IS and CLIP scores, ASDSV maintains scores closest to those of the original model, which aligns with our primary goal of achieving higher fidelity to the original model's output as the ground truth.

Table 2: Efficiency and visual quality comparison of ASDSV and other baselines on video generation.

Method	Efficiency			Visual Quality			
	FLOPs (P) ↓	Latency (s) ↓	Speedup ↑	VBench ↑	LPIPS ↓	PNSR ↑	SSIM ↑
Wan2.1 (81 frames, 832×480)							
Wan2.1 ($T = 50$)	168.1	924	1×	82.69	-	-	-
Sparse-fast	138.4	761	1.21×	78.97	0.75	8.6	0.27
Sparse-slow	153.1	789	1.17×	81.91	0.70	9.2	0.30
TeaCache-fast	84.2	462	2×	81.83	0.45	14.2	0.49
TeaCache-slow	114.4	628	1.47×	82.17	0.37	15.75	0.55
ASDSV-fast	**52.6**	**307**	**3.01×**	**82.36**	0.41	14.89	0.51
ASDSV-slow	74.4	521	1.77×	82.29	**0.33**	**16.58**	**0.58**

In Table 2 (text-to-video), ASDSV-fast achieves the highest 3.01× speedup with better VBench score compared to TeaCache-slow. Additionally, ASDSV-slow achieves the most faithful reproduction (0.3%-0.4% VBench score degradation), along with higher acceleration 1.77× than TeaCache-slow. These results demonstrate that ASDSV not only preserves the better enerative performance of the models but also achieves greater acceleration than existing methods especially in text-to-video generation.

Visualization. Figure 1 presents the images and the videos (the first frame and last frame of the video) generated by ASDSV compared with those by Teacache and the vanilla. Experimental results show that ASDSV not only produces higher visual quality, but also achieves better alignment with the ground truth (original images and videos). Specifically, our method better preserves the color fidelity, positioning and details of characters compared with Teacache. Additional visual comparisons can be found in the Appendix D.

4.3 Ablation Study

Verification Strategy Comparison. We configure ASDSV-medium with $\gamma_1 = 3$ and $\gamma_2 = 9$, using a warmup ratio of 20%. As shown in Figure 6, we design three verification schemes to compare the trade-off between efficiency and quality. Our method offers multiple configurable variants, allowing users to select the optimal configuration based on their specific requirements.

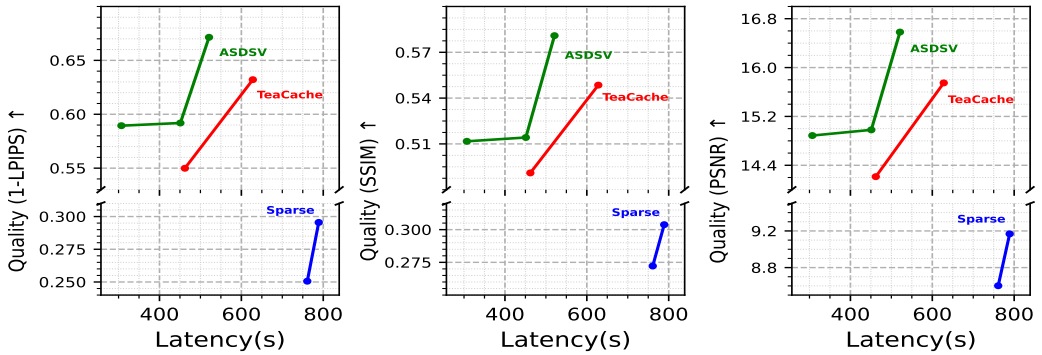
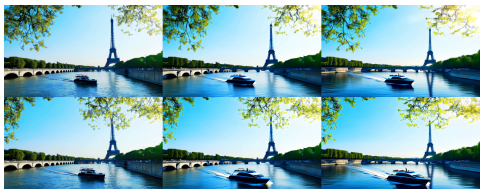


Figure 5: Quality-latency comparison of text-to-video diffusion models. The proposed ASDSV method demonstrates superior visual fidelity while maintaining lower inference latency compared to TeaCache[24] and Sparse VideoGen[38], evaluated under the Wan2.1 model.

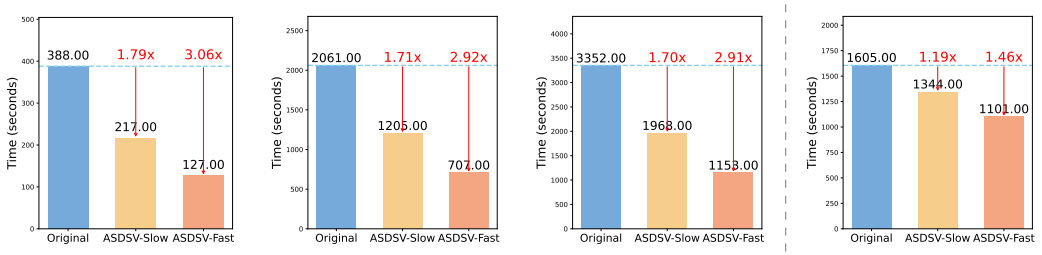


(a) prompt: A boat sailing leisurely along the Seine River with the Eiffel Tower in background, surrealism style.

(b) prompt: A couple in formal evening wear going home get caught in a heavy downpour with umbrellas, zoom in.

Figure 6: Visual comparisons of different verification strategies on text-to-video generation. From left to right: ASDSV-fast, ASDSV-medium, and ASDSV-slow. Each column presents the first and last frames of the generated video.

Performance at different Resolution and Length. Figure 7 illustrates that ASDSV maintains robust acceleration performance across varying video lengths and resolutions. This sustained efficiency highlights ASDSV's capacity to expedite sampling for extended and high-fidelity video content, thereby addressing practical and effective requirements in more complex applications.



(a) 480P, 41 frames (b) 480P, 141 frames (c) 720P, 81 frames (d) 1024×1024, 1 frame

Figure 7: Inference efficiency at different video lengths and resolutions. (a), (b), and (c) show different resolutions and video lengths using Wan2.1, while (d) corresponds to different resolutions using Flux.1-dev.

Sensitivity Analysis and Tuning Process for γ. Both γ_1, γ_2 depend on the temporal correlation between draft and target model on different stages, so ASDSV is sensitive to these parameters. Nevertheless, since the temporal correlation can be profiled offline, we can systematically search for the best candidates for these hyperparameters for each model offline. We generated 200 videos per setting for this specific study on Wan2.1 model, following are the detailed analysis and tuning methods of hyperparameters γ.

Analysis of γ_1. First, we use brute force to search γ_1 because the space is small: it can only be an integer in (1,5). Below is the search space of Wan2.1. We only present one model result for simplicity, other models are the same.

Table 3: Ablation study on the choice of speculative step γ_1 in Stage-1.

γ_1	LPIPS↓	PSNR↑	SSIM↑	Speedup
2	0.23	18.57	0.61	1.70x
3	0.30	16.80	0.59	1.71x
4	0.30	16.76	0.59	1.72x

Analysis of γ_2. Because the search space of γ_2 is larger than γ_1 (an integer selected from range (1,15)), we developed a two-step iterative-guided search process for γ_2 to balance speed-fidelity trade-off: (1) Profile Model Dynamics: We conduct a small-scale sampling run to profile the model's behavior, generating plots similar to Figure 2. This allows us to observe the output change dynamics and step-wise verification loss, which are crucial for setting the γ_2 and decide the verfication threshold. (2) Iteratively Tune γ_2 and δ: We start with a candidate γ_2 based on the theoretical speedup ceiling for our target. Next, using the verification loss curve (like Figure 2(b)(c)), we set a corresponding threshold δ above the observed loss at the γ_2-th step. We then incrementally increase γ_2 and adjust δ, evaluating the trade-off on a validation set until the desired speedup is met with minimal quality impact. The following results show typical search results on Wan2.1 model, others are the same:

Table 4: Ablation study on the choice of speculative step γ_2 and verification threshold δ in Stage-2.

γ_2	δ	LPIPS↓	PSNR↑	SSIM↑	Speedup
3	0.1	0.24	18.89	0.60	1.2x
9	0.2	0.23	18.57	0.61	1.7x
12	∞	0.25	18.47	0.60	3.1x

The baseline for these comparisons is the original model, corresponding to the setting $\gamma_1 = 0, \gamma_2 = 0$. The setting of $\delta = \inf$ represents an "always-accept" strategy that effectively bypasses the verification stage to achieve maximum acceleration, corresponding to our ASDSV-fast variant in Figure 5. We have summarized the tuning process described above into an automated method, which is detailed in Appendix B.

Quality-Efficiency Tradeoff. Figure 5 shows the trade-off between the quality and the efficiency of our method compared with other baselines. Notably, our method achieves the best scores not only on the reference-free VBench, but also on similarity metrics. This performance stems from the fact that ASDSV most closely aligns with the outputs of the original model, resulting in quality scores that are nearly indistinguishable from those of SOTA diffusion models.

5 Conclusion and Future Work

This paper introduces Approximate Speculative Diffusion with Speculative Verification (ASDSV), a novel speculative decoding method for multimodal generation. Different from traditional speculative decoding, ASDSV employs a speculative verification strategy specifically for diffusion models to minimize verification cost, by leveraging the temporal correlation between draft and target models. Meanwhile, we propose a multi-stage speculative strategy based on the dynamics of diffusion process to to balance the trade-off between generation speedup and performance. Experiments demonstrate that ASDSV achieves significant inference acceleration while maintaining high similarity and robust performance in both video and image generation models.

Limitations and Future Work. First, ASDSV performance depends on the cost coefficient between models, with limited gains when target models are already optimized or draft models lack efficiency. Second, our current verification failure handling discards all subsequent samples. Advanced recovery strategies could potentially enhance performance, such as employing binary search techniques to efficiently identify the last acceptable draft sample. Third, integration with parallelism strategies requires further research to maximize performance in distributed environments.

Acknowledgments

We sincerely thank the anonymous reviewers, whose reviews, feedback, and suggestions have significantly strengthened our work. This research was supported in part by New Generation Information Technology Program from Shanghai Committee of Science and Technology (NO.25511104100), National Natural Science Foundation of China (No. 62432010), and the Fundamental Research Funds for the Central Universities.

References

[1] Shubham Agarwal, Subrata Mitra, Sarthak Chakraborty, Srikrishna Karanam, Koyel Mukherjee, and Shiv Kumar Saini. Approximate caching for efficiently serving text-to-image diffusion models. In *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation*, NSDI'24, USA, 2024. USENIX Association. ISBN 978-1-939133-39-7.

[2] Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling, 2025.

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

[4] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. URL `https://arxiv.org/abs/2310.00426`.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015. URL `https://arxiv.org/abs/1504.00325`.

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

[7] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL `https://arxiv.org/abs/2105.05233`.

[8] Hangliang Ding, Dacheng Li, Runlong Su, Peiyuan Zhang, Zhijie Deng, Ion Stoica, and Hao Zhang. Efficient-vdit: Efficient video diffusion transformers with attention tile, 2025. URL `https://arxiv.org/abs/2502.06155`.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL `https://arxiv.org/abs/2403.03206`.

[10] Jiarui Fang, Jinzhe Pan, Xibo Sun, Aoyu Li, and Jiannan Wang. xdit: an inference engine for diffusion transformers (dits) with massive parallelism. *arXiv preprint arXiv:2411.01738*, 2024.

[11] Jiarui Fang, Jinzhe Pan, Jiannan Wang, Aoyu Li, and Xibo Sun. Pipefusion: Patch-level pipeline parallelism for diffusion transformers inference, 2024. URL `https://arxiv.org/abs/2405.14430`.

[12] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2024. URL `https://arxiv.org/abs/2307.04725`.

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. URL `https://arxiv.org/abs/2104.08718`.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[16] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models, 2023. URL https://arxiv.org/abs/2311.17982.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

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

[19] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

[20] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Ming-Yu Liu, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models, 2024. URL https://arxiv.org/abs/2402.19481.

[21] Muyang Li*, Yujun Lin*, Zhekai Zhang*, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[22] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 20662–20678. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/41bcc9d3bddd9c90e1f44b29e26d97ff-Paper-Conference.pdf.

[23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[24] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. *arXiv preprint arXiv:2411.19108*, 2024.

[25] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan LI, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5775–5787. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/260a14acce2a89dad36adc8eefe7c59e-Paper-Conference.pdf.

[26] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. URL https://arxiv.org/abs/2211.01095.

[27] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, Yu Zhou, Deshan Sun, Deyu Zhou, Jian Zhou, Kaijun Tan, Kang An, Mei Chen, Wei Ji, Qiling Wu, Wen Sun, Xin Han, Yanan Wei, Zheng Ge, Aojie Li, Bin Wang, Bizhu Huang, Bo Wang, Brian Li, Changxing Miao, Chen Xu, Chenfei Wu, Chenguang Yu, Dapeng Shi, Dingyuan Hu, Enle Liu, Gang Yu, Ge Yang, Guanzhe Huang, Gulin Yan, Haiyang Feng, Hao Nie, Haonan Jia, Hanpeng Hu, Hanqi Chen, Haolong Yan, Heng

Wang, Hongcheng Guo, Huilin Xiong, Huixin Xiong, Jiahao Gong, Jianchang Wu, Jiaoren Wu, Jie Wu, Jie Yang, Jiashuai Liu, Jiashuo Li, Jingyang Zhang, Junjing Guo, Junzhe Lin, Kaixiang Li, Lei Liu, Lei Xia, Liang Zhao, Liguo Tan, Liwen Huang, Liying Shi, Ming Li, Mingliang Li, Muhua Cheng, Na Wang, Qiaohui Chen, Qinglin He, Qiuyan Liang, Quan Sun, Ran Sun, Rui Wang, Shaoliang Pang, Shiliang Yang, Sitong Liu, Siqi Liu, Shuli Gao, Tiancheng Cao, Tianyu Wang, Weipeng Ming, Wenqing He, Xu Zhao, Xuelin Zhang, Xianfang Zeng, Xiaojia Liu, Xuan Yang, Yaqi Dai, Yanbo Yu, Yang Li, Yineng Deng, Yingming Wang, Yilei Wang, Yuanwei Lu, Yu Chen, Yu Luo, Yuchu Luo, Yuhe Yin, Yuheng Feng, Yuxiang Yang, Zecheng Tang, Zekai Zhang, Zidong Yang, Binxing Jiao, Jiansheng Chen, Jing Li, Shuchang Zhou, Xiangyu Zhang, Xinhao Zhang, Yibo Zhu, Heung-Yeung Shum, and Daxin Jiang. Step-video-t2v technical report: The practice, challenges, and future of video foundation model, 2025. URL https://arxiv.org/abs/2502.10248.

[28] Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching, 2024. URL https://arxiv.org/abs/2406.01733.

[29] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2023. URL https://arxiv.org/abs/2210.03142.

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

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[32] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

[33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[35] Zhou Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002. doi: 10.1109/97.995823.

[36] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[37] WanTeam, 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.

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

[39] Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation, 2023. URL https://arxiv.org/abs/2203.16487.

[40] Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding, 2024. URL https://arxiv.org/abs/2401.07851.

[41] Yuchen Xia, Divyam Sharma, Yichao Yuan, Souvik Kundu, and Nishil Talati. Modm: Efficient serving for image generation via mixture-of-diffusion models, 2025. URL https://arxiv.org/abs/2503.11972.

[42] Mengwei Xu, Mengze Zhu, Yunxin Liu, Felix Xiaozhu Lin, and Xuanzhe Liu. Deepcache: Principled cache for mobile deep vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, page 129–144, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359030. doi: 10.1145/3241539.3241563. URL https://doi.org/10.1145/3241539.3241563.

[43] Zhihang Yuan, Hanling Zhang, Lu Pu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. DiTFastattn: Attention compression for diffusion transformer models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=51HQpkQy3t.

[44] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

[45] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast, 2025. URL https://arxiv.org/abs/2408.12588.

[46] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

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: They are properly stated.

   Guidelines:

   - The answer NA 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 NA 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: They are properly stated in Section 5.

   Guidelines:

   - The answer NA 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: They are properly stated in Section 3.3.

Guidelines:

- The answer NA 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: They are included in 4.1.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: Codes provided in supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: They are properly stated in Section 4.1 and Appendix.

   Guidelines:

   - The answer NA 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: We do not report error bars.

   Guidelines:

   - The answer NA 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: They are properly stated in Section 4.1 implementation details.

   Guidelines:

   - The answer NA 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: It conforms in every aspect.

   Guidelines:

   - The answer NA 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: Discussed in Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We have no new models/datasets.

    Guidelines:

    - The answer NA 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 a best 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: Credits for all assets are well given.

    Guidelines:

    - The answer NA 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: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subject.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subject.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The LLM is only used for writing, editing. It is not used as any important, original, or non-standard components.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Additional Experiments

A.1 Results of Draft Model Comparison

In autoregressive language models, draft models (M_q) are typically selected from existing off-the-shelf smaller transformers with the same architecture as the target model (M_p). For example, T5-XXL (11B) as M_p and T5-large (800M) as M_q. Following this practice, we evaluate the performance of different draft models on text-to-image generation. Flux.1-dev is the state-of-the-art text-to-image model, and Flux-lite (distilled model) and Flux-SVD (quantized model) are two efficient draft variants of Flux.1-dev.

Table 5 illustrates that ASDSV consistently outperforms both draft models, which demonstrates the effectiveness and broad applicability of ASDSV across diverse compression techniques. The quantized model Flux-SVD shows the best overall performance and is chosen as the draft model for Flux.1-dev in our experiments. FID, LPIPS and PSNR are reference-based metrics for measuring the similarity between generated results and Flux.1-dev original outputs. CLIP score measures the semantic alignment between the input prompt and the generated images and IS measures the diversity of the generated images.

Table 5: Performance of FLUX.1-dev compared with different draft models. *ASDSV w/ Flux-lite* and *ASDSV w/ Flux-SVD* are ASDSV that use Flux.1-dev as the target model and Flux-lite and Flux-SVD as draft model, respectively.

Model	**FLUX.1-dev**				
Score	**FID** ↓	**CLIP** ↑	**LPIPS** ↓	**IS** ↑	**PSNR** ↑
Flux.1-dev	—	31.14	—	26.32	—
Flux-lite	9.51	31.07	0.40	25.59	29.73
Flux-SVD	5.52	**31.18**	0.19	26.26	30.53
ASDSV w/ Flux-lite	6.62	31.20	0.22	25.68	30.49
ASDSV w/ Flux-SVD	**4.97**	**31.24**	**0.11**	**26.36**	**32.69**

Additionally, we improved the draft model baseline by combined the Stage-0 into the draft model diffusion process. ASDSV-slow and ASDSV-fast are the two variants of ASDSV with different settings, detailed in Experiments Settings of the main paper. Specifically, we use the target model to generate the first $N = 5$ steps as the draft model's initial steps (the same as the ASDSV-fast), and then use the draft model to generate the remaining steps. The evaluation metric results are shown in Table 6. The visual results are shown in Figure 8. ASDSV outperforms the Improved-Draft baseline in both visual quality metric (VBench score) and similarity metrics (LPIPS, PSNR, and SSIM).

Table 6: Efficiency and visual quality comparison of ASDSV and improved draft model baseline on text-to-video generation.

Method	**Efficiency**			**Visual Quality**			
	FLOPs (P) ↓	**Latency (s)** ↓	**Speedup** ↑	**VBench** ↑	**LPIPS** ↓	**PNSR** ↑	**SSIM** ↑
Improved-Draft	28.0	256	3.67×	82.23	0.42	14.70	0.50
ASDSV-fast	52.6	307	3.01×	**82.36**	0.41	14.89	0.51
ASDSV-slow	74.4	521	1.77×	82.29	**0.33**	**16.58**	**0.58**

A.2 Multi-Seed Stability Analysis

Our method is insensitive to the random seed. This is because the seed in the diffusion process only affects the initial noise, whereas the efficacy of ASDSV depends on the temporal correlation between the draft and target models (as shown in Figure 2).

We conducted a new set of focused experiments on three additional seeds (3461, 772190542, 118270042274490399) for our main comparisons (Original vs. ASDSV). For this analysis, we

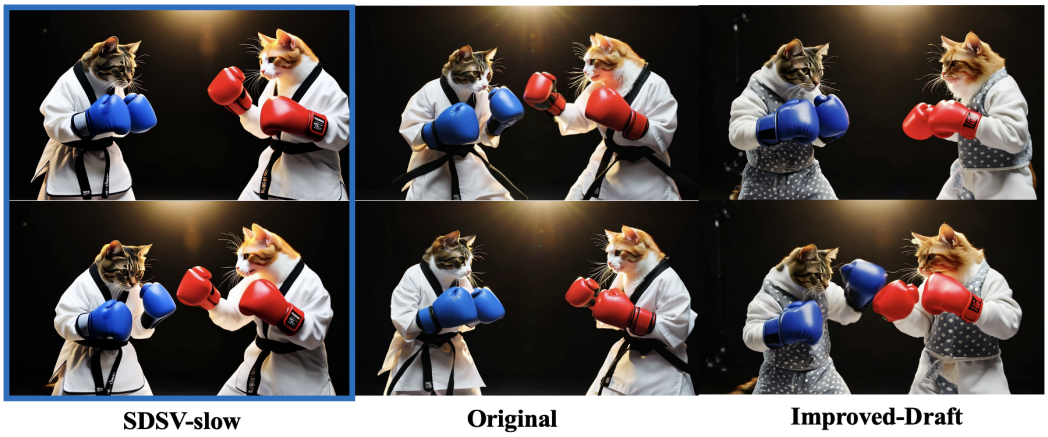


Figure 8: Visual comparisons of improved draft model baseline on text-to-video generation.

generated 2000 images per seed for the Text-to-Image task and 200 videos per seed for the Text-to-Video task. Performance was consistent across all seeds (ASDSV achieves better performance than other acceleration methods). The standard deviations over multiple seeds results on Text-to-Image(T2I) over are shown in Table 7.

Table 7: Standard deviations over multiple seeds for T2I generation.

Images Metric	Ratio (Original / ASDSV)
CLIP	$100.04\% \pm 0.10\%$
IS	$103.84\% \pm 3.7\%$

Since most of the Text-to-Video(T2V) metrics are reference-based, we cannot calculate a relative ratio (original/accelerated method). Therefore, we used a comparison between the accelerated methods (ASDSV vs. Teacache) as shown in Table 8, and ASDSV is about 1.5x faster than Teacache.

Table 8: Standard deviations over multiple seeds for T2V generation.

Videos Metric	Ratio (ASDSV / Teacache)
LPIPS↓	$88.02\% \pm 2.85\%$
PSNR↑	$106.10\% \pm 1.25\%$
SSIM↑	$104.95\% \pm 1.84\%$

A.3 Analysis of Multi-Stage Speculative Strategy

ASDSV employs a multi-stage speculative strategy to balance the trade-off between generation speedup and performance: first N initial steps (Stage-0) combined with the subsequent Stage-1 serve as the warmup period for the diffusion process, establishing the foundation for high-quality generation, followed by the main generation stage as Stage-2. A key characteristic of the warmup period is the relatively large variation between outputs of adjacent steps, unlike the highly similar outputs observed between consecutive steps in Stage-2. To ensure higher quality model outputs, modifications to the architecture are typically avoided during the warmup period.

We evaluated two warmup approaches: the 2 steps method and a more conservative fixed ratio of 20% of total diffusion steps. After completing their respective warmup phases, they employ the same speculative strategies as ASDSV in Stage-2. Our experimental results in Figure 9 reveal a clear visual difference. Configuration (a) with minimal warmup steps significantly reduces target model invocations, improving speed but compromising generation quality. In contrast, configuration (c) with larger warmup steps maintains higher quality output but with reduced acceleration. To balance this trade-off, we divided the warmup phase into two distinct components (Stage-0 and Stage-1): first allowing the target model to establish critical structural elements, then employing small speculative steps to accelerate the remaining warmup procedure. As demonstrated in configuration (d), which

uses 2 steps for Stage-0 followed by speculative steps with $K = 3$ for Stage-1, then employing the same $K = 9$ speculative steps for Stage-2 as configurations (a) and (c). This multi-stage approach achieves nearly $2.1\times$ acceleration over the vanilla diffusion process while preserving better visual quality compared to configuration (a).

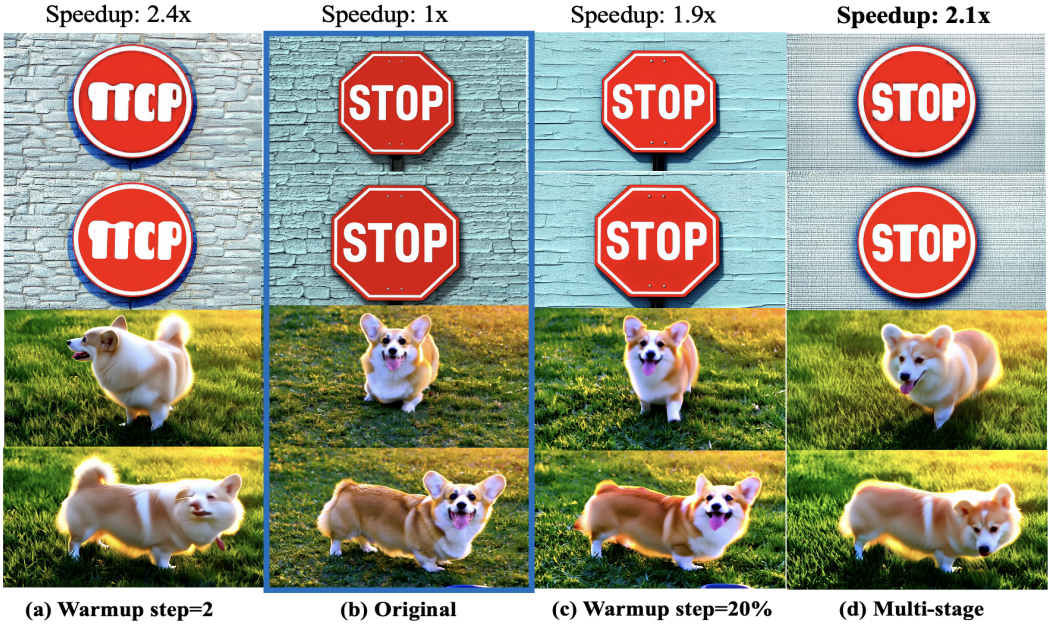


Figure 9: Comparison of different warmup and stage strategies: (a) minimal warmup with 2 steps followed by ASDSV Stage-2, (b) vanilla diffusion process, (c) conservative warmup with 20% of total steps (10 steps) followed by ASDSV Stage-2, (d) proposed multi-stage strategy with 2 steps for Stage-0 and 8 steps for Stage-1 (20% total diffusion steps as warmup), followed by the same ASDSV Stage-2.

A.4 Results of FID against Real Images

ASDSV aims to preserve the generation quality of original models, so that we only present the FID compared with the original generationed images in the paper. We now provide the FID scores against real images on COCO Captions 2014 in Table 9:

Table 9: FID scores against real images on COCO Captions 2014.

Method	FID (based ground truth)	FID (based original) ↓
original	35.63	–
ASDSV-fast	**35.25**	4.11
ASDSV-slow	35.31	**3.90**

The scores based ground truth here are very close, with a change of only 0.9%. Although the score for ASDSV-fast is higher (0.17% higher than ASDSV-slow), ASDSV-slow is actually closer to the original score, which also meets our expected goal.

A.5 Expected Speedup of ASDSV

Here is the theoretical formula for the expected speedup, which incorporates the verification failure rate $E(\beta)$. The expected speedup factor for each stage is as follows:

$$S_{expected} = \frac{(1 - E(\beta))^2\gamma + (1 - (1 - E(\beta))^2)}{\gamma c + 2} \tag{1}$$

Proof. Let the cost of running a single step of M_p by T. An unoptimized round of ASDSV costs $\gamma cT + 2T$. This cost covers running the draft model M_q γ times and running the target model M_p

twice (at the first and last step). Assuming the two verification checks are independent, the expected number of steps advanced per cycle is given by:

$$E[\text{steps}] = (1 - E(\beta))^2\gamma + (1 - (1 - E(\beta))^2) \cdot 1$$

This calculation assumes simple failure handling, where a failed verification results in advancing a single step. Thus, the effective cost per advanced step is the total cycle cost divided by the expected steps advanced:

$$\frac{\gamma cT + 2T}{(1 - E(\beta))^2\gamma + (1 - (1 - E(\beta))^2)}$$

Since the cost of generating a single step with the target model $M_p$is T, we get the desired result as equation (1).

As stated in Section 3.3, c is a system-dependent constant, determined by the specific hardware and software implementation. A smaller c (a faster draft model) reduces the drafting overhead. However, it may imply a simpler model that struggles to approximate the target model's output, thus increasing the verification failure rate $E(\beta)$. In our experiments, conservatively-chosen γ leads to few failure cases.

B Hyperparameter Auto-Tuning Process

As a follow-up to the sensitivity analysis in Section 4.3, this section details the principled, semi-automated process for determining the key hyperparameters for new models. We address each hyperparameter below:

For γ_1 and Initial Steps N (Generalization): We found γ_1 to be a less sensitive hyperparameter for the short Stage 1, generalizing well across models. A fixed, conservative value (typically 2 or 3) is a robust choice. The initial step count N is set to 5, analogous to the warm-up phase in other methods [10, 20].

For the Stage Boundary (Profile Model Dynamics): We determine the boundary between Stage 1 and Stage 2 via a small-scale profiling run. By generating a small number of samples (as in Figure 2(a)), we can observe the model's output dynamics. This empirical data is crucial for setting the stage boundary.

For γ_2 and δ (Iterative Tuning): We start with a candidate γ_2 based on the theoretical speedup ceiling for our target (Section 3.3). Next, using the verification loss curve obtained from the profiling stage (e.g., Figure 2(b)(c)), we set a corresponding threshold δ slightly above the observed loss at the γ_2-th step. Finally, we iteratively increase γ_2 and adjust δ, evaluating the trade-off on a validation set until the desired speedup is met with minimal quality impact.

C Social Impact

The acceleration of diffusion models provided by ASDSV reduces computational resources and latency, improving real-time applicability of state-of-the-art diffusion models while promoting environmental sustainability through reduced energy consumption. However, it is important to note that ASDSV focuses primarily on efficiency gains and does not address inherent challenges such as privacy, bias, and fairness in the underlying diffusion models.

D Additional Visualization

We provide comprehensive visual comparisons between ASDSV and baseline acceleration methods for both text-to-image and text-to-video generation, as shown in Figure 10 and Figure 11. Extensive results demonstrate the superior visual fidelity of ASDSV across different generation scenarios.

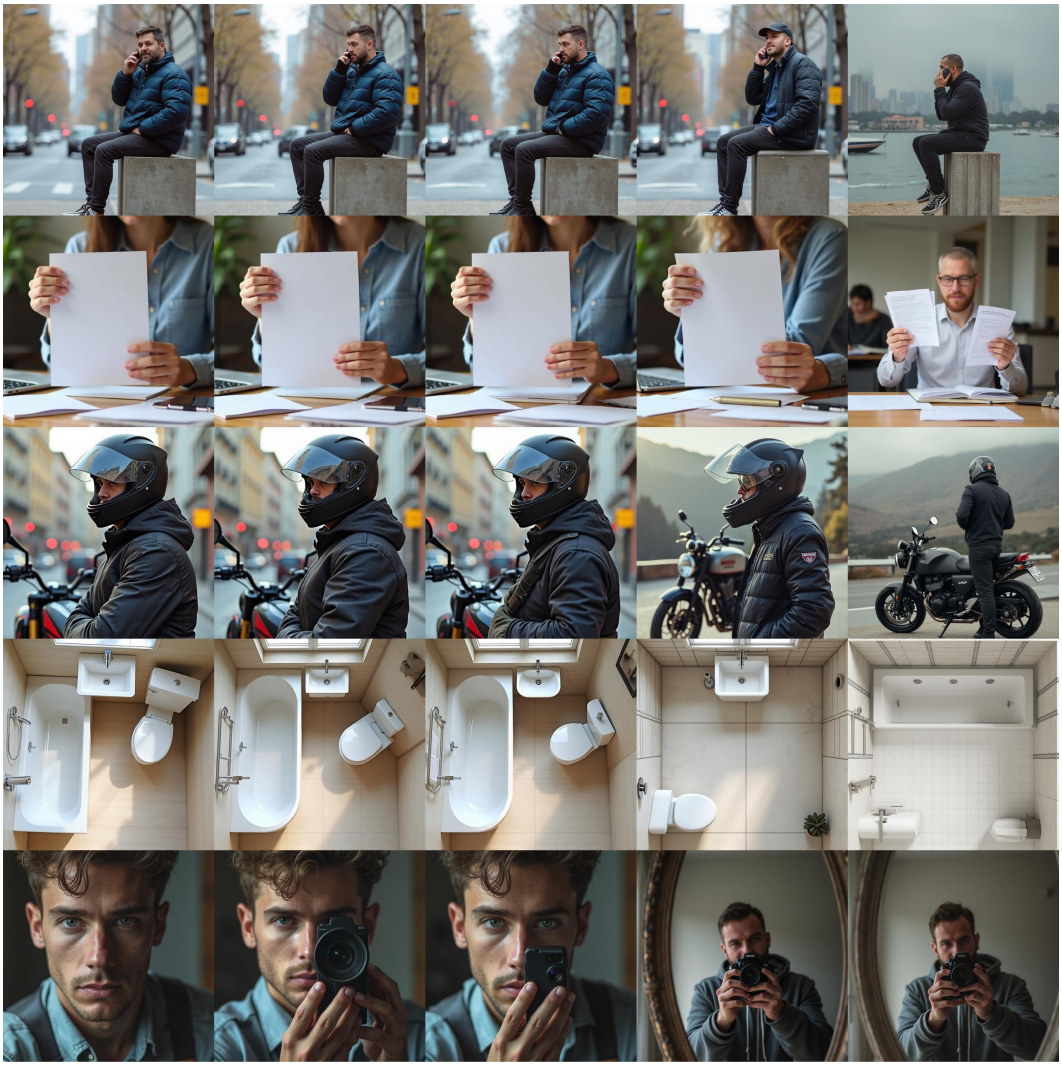


Figure 10: Comparison of different accelerating methods on text-to-image generation using Flux.1-dev at 512x512 resolution.

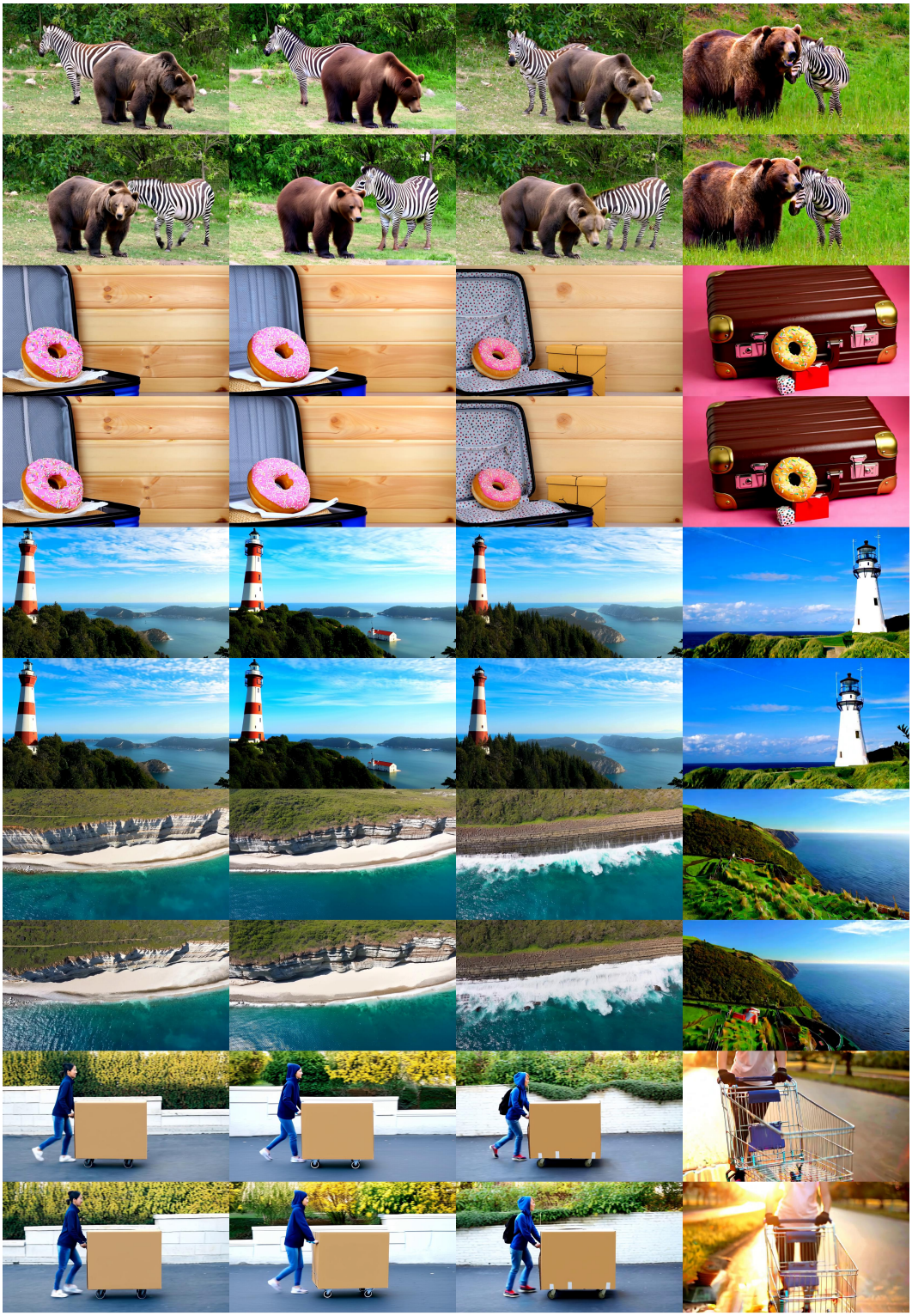


Figure 11: Comparison of different accelerating methods on text-to-video generation using Wan2.1 at 480P resolution with 81 frames.