

## Video-T1: Test-Time Scaling for Video Generation

Fangfu Liu<sup>1\*</sup>, Hanyang Wang<sup>1\*</sup>, Yimo Cai<sup>1</sup>, Kaiyan Zhang<sup>1</sup>, Xiaohang Zhan<sup>2</sup>, Yueqi Duan<sup>1†</sup>  
<sup>1</sup>Tsinghua University, <sup>2</sup>Tencent



Figure 1. **Video-T1**: We present the generative effects and performance improvements of video generation under Test-Time Scaling (TTS) settings. The videos generated with TTS are of higher quality and more consistent with the prompt than those generated without TTS.

### Abstract

With the scale capability of increasing training data, model size, and computational cost, video generation has achieved impressive results in digital creation, enabling users to express creativity across various domains. Recently, researchers in Large Language Models (LLMs) have expanded the scaling to test-time, which can significantly improve LLM performance by using more inference-time computation. Instead of scaling up video foundation models through expensive training costs, we explore the power of Test-Time Scaling (TTS) in video generation, aiming to answer the question: if a video generation model is allowed to use non-trivial amount of inference-time compute, how much can it improve generation quality given a challenging text prompt. In this work, we reinterpret the test-time scal-

ing of video generation as a searching problem to sample better trajectories from Gaussian noise space to the target video distribution. Specifically, we build the search space with test-time verifiers to provide feedback and heuristic algorithms to guide searching process. Given a text prompt, we first explore an intuitive linear search strategy by increasing noise candidates at inference time. As full-step denoising all frames simultaneously requires heavy test-time computation costs, we further design a more efficient TTS method for video generation called Tree-of-Frames (ToF) that adaptively expands and prunes video branches in an autoregressive manner. Extensive experiments on text-conditioned video generation benchmarks demonstrate that increasing test-time compute consistently leads to significant improvements in the quality of videos. Project Page: <https://liuff19.github.io/Video-T1>.

\*Equal contribution. † The corresponding author.

## 1. Introduction

The field of generative modeling has witnessed remarkable progress in recent years [1, 39, 42, 58], with applications spanning from image and text generation to more complex tasks, such as video synthesis. Among these, video generation [22, 23] stands out due to its potential to revolutionize digital content creation, enabling the automatic production of high-quality videos from simple textual descriptions [60]. This capability has profound implications for various industries [22, 23, 30] (e.g., entertainment, education, and advertisements). The pivotal factor of the exponential growth in video generation lies in the scaling-up capability by training with an expanding volume of data, more computational sources, and larger model sizes [22, 38]. This scaling behavior during the training process, commonly referred to as *Scaling Laws* [12, 19, 38, 41], plays a crucial guiding role in the advancement of generative models with progressively higher capabilities.

Despite these advancements, generating high-quality videos remains challenging due to the need for maintaining temporal coherence and capturing complex dynamics across frames [60]. While scaling video generation methods in the training process [22, 32] has yielded significant improvements, it is inherently limited by high costs and resource demands, making it challenging to scale further. Recently, researchers in LLMs have expanded the study of scaling to the test-time [29] (e.g., DeepSeek-R1 [8] and OpenAI o1 [15]) and demonstrated that Test-time Scaling (TTS) can significantly improve the performance of LLMs with more contextually appropriate responses by allocating additional computation at inference time [8, 15, 47, 52].

In this paper, we propose to investigate Test-Time Scaling (TTS) for video generation. Specifically, we aim to answer the question: *If a video generation model is permitted to use the larger amount of inference-time computation, how much can it improve the generation quality for challenging text prompts?* We seek to explore the potential of TTS to enhance video generation without the need for expensive retraining or model enlargement. To understand the benefits of scaling up test-time computation in video diffusion, we propose a general framework for TTS video generation, called **Video-T1**, which reinterprets the TTS of video generation as a searching problem within the space of possible video trajectories originating from Gaussian noise. The key insight is to scale the search space at test time with increased computation so that we can find a broader range of potential solutions to generate higher-quality and text-aligned videos. In our search framework, we introduce test-time verifiers to assess the quality of intermediate results and heuristic algorithms to navigate the search space efficiently. Initially, we conduct a straightforward random linear search strategy by sampling  $N$  noise candidates in parallel and selecting the one that scores the highest per a

test-time verifier. However, recognizing the computational intensity of this approach, particularly when denoising all video frames simultaneously, we introduce a more efficient framework called Tree-of-Frames (ToF). ToF operates in an autoregressive manner under a tree structure, which leverages the feedback from verifiers and adaptively expands and prunes branches of video frames to balance computational cost and generation quality. Through extensive experiment on text-conditioned video generation benchmark, our findings reveal that increasing test-time compute leads to substantial improvements in the quality and human-preference alignment of samples generated by video generation models. Longer term, this offers a significant promise on how to leverage inference-time computation to achieve superior results in computer vision. (See qualitative results gallery in Figure 1). Our contributions are summarized as follows:

- We propose a fundamental framework *Video-T1* for test-time scaling for video generation, which reinterprets this process as a search problem to sample better video trajectories. We show that scaling the search space of video generation can boost video performance across different dimensions of the benchmark.
- We carefully build the search space in test-time scaling by test-time verifiers to provide feedback and heuristic algorithms (i.e., a straightforward random linear search and ToF search for more efficient test-time scaling) to guide the search process.
- Extensive experiments demonstrate that scaling the search space of video generation can boost the performance of various video generation models across different dimensions of the benchmark, and our proposed ToF search can significantly reduce scaling cost when achieving high-quality results.

## 2. Related Work

### 2.1. Test-Time Scaling in LLMs

Recent advancements have demonstrated the effectiveness of test-time scaling (TTS) methods such as chain-of-thought prompting [34, 53], outcome reward models, and process reward models [25, 50, 62] in enhancing the reasoning capabilities of large language models (LLMs) during inference stages. Notable examples include implementations in OpenAI o1 [15] and DeepSeek-R1 [8]. These methods promote the generation of intermediate reasoning steps, resulting in more precise responses. These researches suggests that reallocating computational resources from pre-training [18] to test-time can enhance performance more efficiently [29, 43]. Moreover, strategies like self-consistency [5, 51], best-of- $N$  [36, 45], Monte Carlo Tree Search [54, 65], and Reward-guided Search [6, 20] employ diverse generation techniques and sophisticated aggregation methods, often facilitated by process reward models. These

approaches help in producing diverse and integrated outputs. The combination of parallel and sequential generation techniques in these models represents a nuanced approach to generating contextually appropriate outputs, thereby establishing new operational standards for LLMs in complex problem-solving scenarios.

## 2.2. Test-Time Scaling in Computer Vision

In both the visual understanding and visual generation fields, researchers have investigated various test-time scaling methods to further push the performance boundaries. With the success of test-time scaling methods in LLMs, several recent vision language models (VLMs) [48, 55] utilized step-by-step reasoning capability enhanced by test-time scaling methods and surpassed larger models in visual question-answering tasks. Recent investigations on image diffusion models have demonstrated that image diffusion models' generation quality could be further enhanced with test-time scaling methods [9]. With verifiers providing judgments and algorithms selecting better candidates, image diffusion models consistently improve their performance across generation tasks by scaling up inference time [33].

## 2.3. Video Generation

Efficient and high-quality video generation has attracted increasing attention due to its wide applications in areas [26–28, 46]. With the success of diffusion models [11, 40] in text-to-image generation, several studies have extended them to text-to-video (T2V) tasks, achieving promising results. One line of work [2, 4, 13, 31, 58] improves video quality by scaling up diffusion transformer (DiT) [37] pre-training, leading to high visual fidelity and smoother motion. These models have reached near-production-level performance but require extensive computational resources, especially for long videos [16]. Another line of work [3, 7, 16, 17, 21, 57] combines diffusion models with autoregressive mechanisms to better handle long and complex videos. For example, NOVA [7] generates videos by predicting frames sequentially over time while sampling tokens in random spatial order, unifying various generation tasks into a single framework. Pyramid-Flow [16] redefines the generation process as a multi-scale trajectory over compressed representations, using spatial and temporal pyramids to reduce training costs while maintaining quality. The autoregressive approaches show strong potential to generate longer, coherent, and high-quality videos with improved efficiency, making them a promising direction for future research.

## 3. Method

### 3.1. How to Scale Video Generation at Test Time

In the realm of LLMs, researchers have explored the benefits of scaling up test-time computation to boost model performance. Several key factors have been identified that shape the effectiveness of test-time scaling strategies in LLMs, such as the choice of policy models, process reward models (PRMs), and varying levels of problem difficulty [29, 44]. Similarly, Test-Time Scaling (TTS) in video generation hinges on key components like different video generation models, multimodal evaluation models, and the complexity of prompts across diverse benchmark dimensions. However, unlike LLMs, video generation poses specific challenges. First, videos inherently exhibit strong temporal continuity, meaning that while they consist of discrete frames, ensuring smooth transitions between frames is essential for perceptually coherent results. Second, state-of-the-art video generation models are primarily based on diffusion models, which employ a multi-step denoising process that complicates the direct scaling of computational resources. These factors introduce additional complexities: test-time scaling in video generation must simultaneously address both spatial (frame-level) quality and temporal consistency while also considering the heavy iterative diffusion denoising process.

To address these challenges, we propose to reinterpret video TTS as a path-search problem to sample better trajectories from pure Gaussian noise space to the target video distribution. The key insight is to scale the search space at test time with increased computation so that we can explore a broader range of potential solutions to generate higher-quality and text-aligned videos. Taking a closer look at this scheme, a video can be represented as a sequence of discrete frames. Considering the temporal nature of the frame sequence, it can be modeled as a chain-like architecture, where the video generation resembles the growth of a degenerate tree (*i.e.*, a tree where each non-leaf node has exactly one child – rooted in the Gaussian noise space of the video domain). In this way, we formalize the generation of a high-quality video as a searching problem: starting from an initial root node, we seek a path through  $T$  steps that reaches a leaf node, maximizing the quality along the generated sequence. To build such a search space, we define several key components:

- **Video Generator  $\mathcal{G}$ :** Video generation models, which generate videos from given text prompts by the multi-step denoising process. Formally, we define:

$$\mathcal{G} : c \rightarrow \mathbb{R}^{H \times W \times C \times T}, \quad (1)$$

where  $c$  represents the input text condition, and the output is a generated video with  $T$  frames.



- **Test Verifiers  $\mathcal{V}$ :** Multimodal evaluation models that assess the quality of generated videos and assign a final score to provide feedback in the generation process. This can be expressed as:

$$\mathcal{V} : \mathbb{R}^{H \times W \times C \times T} \times c \rightarrow \mathbb{R}, \quad (2)$$

where the function takes both the generated video and the input condition to produce a scalar quality score.

- **Heuristic Search Algorithms  $f$ :** The optimization methods that leverage feedback from the verifier to guide the search trajectory, ultimately finding better video sequences. We define this as:

$$f : \mathcal{G} \times \mathcal{V} \times (\mathbb{R}^{H \times W \times C})^N \times c \rightarrow \mathbb{R}^{H \times W \times C \times T}, \quad (3)$$

where  $(\mathbb{R}^{H \times W \times C})^N$  represents the set of  $N$  initial noise samples (*i.e.*, root nodes in the search forest), and  $\mathbb{R}^{H \times W \times C \times T}$  denotes the final selected video sequence (*i.e.*, a path from a root node to a leaf node at depth  $T$ ).

### 3.2. Random Linear Search

A straightforward approach for TTS video generation is to randomly sample Gaussian noises, prompt  $\mathcal{G}$  to generate complete video sequences by performing the full denoising process for each sample, and perform the Best-of- $N$  selection to obtain the one with the highest score from the test verifiers  $\mathcal{V}$ . We refer to this method as **random linear search** (the top of Figure 2), as it performs step-by-step denoising in a linear manner. In this search algorithm, the only scaling factor for test-time scaling is the number of noise samples  $N$ , leading to a computational cost that increases linearly with the number of samples.

From a more structural perspective, random linear search can be interpreted as a forest consisting of  $N$  degenerate trees, where each tree represents an independent sequence of  $T$  denoising steps. The search task then reduces to selecting a better length- $T$  path among them. The total number of nodes in the forest is  $TN$ , leading to a generation time complexity of  $O(TN)$ . Since each video evaluation requires a constant-time assessment of its quality, the evaluation cost per sample is  $O(1)$ , resulting in an overall quadratic time and space complexity  $O(TN)$ .

While random linear search provides a simple baseline, its linear structure introduces two inherent limitations: 1) **Simplicity of linear structure.** Although the final path selects a single branch, the tight bounds of this approach require exhaustive traversal of the entire space, lacking efficient optimization mechanisms. 2) **Isolation of independent structure.** Without any feedback or interaction mechanisms between trees, it introduces additional randomness, making it slower for test-time scaling.

### 3.3. Tree-of-Frames Search

Random linear search is essentially adopted by a Best-of- $N$  strategy that scales test-time computation through in-

creasing the number of initial noise samples  $N$ . However, this approach requires a fixed time complexity of  $O(TN)$  as analyzed above, which becomes increasingly inefficient as either  $N$  scales up or the video length  $T$  grows, making it impractical for long video generation or high-quality sampling at larger scales. To address this limitation and achieve a better balance between video quality and test-time computational efficiency, we propose **Tree-of-Frames (ToF) Search** (Algorithm 1), which leverages the sequential generation capability of autoregressive models, introducing inference-time reasoning along the temporal dimension. Given a text prompt as input, (a) the first stage is to generate the initial frame with text alignment on various dimensions (*e.g.*, spatial relation, appearance style, color), which strongly impacts later frames due to the continuity of video frames; (b) the second stage focuses on generating intermediate frames which should consider the key factors like subject consistency, motion stability, even physics plausibility to guarantee a smooth video flow; (c) the final stage is dedicated to assessing the overall video quality and alignment with text prompts. According to the three-stage goal, we meticulously design three key techniques in the ToF search algorithm: *image-level alignment*, *hierarchical prompting*, and *heuristic pruning*.

**Image-level alignment.** Unlike LLMs, video generation spans spatial and temporal dimensions. Spatially, video frames undergo step-wise denoising in diffusion models. Inspired by Chain-of-Thought (CoT) reasoning in image generation [9], we propose a progressive frame-level evaluation to dynamically scale denoising computation. A test verifier assesses partially denoised images for clarity. Early, blurry frames are skipped to avoid misleading evaluations. Once visually informative, the model evaluates a frame’s potential for high-quality completion. By rejecting low-potential candidates early and focusing compute on promising trajectories, image-level scaling enhances inference efficiency.

**Hierarchical prompting.** We propose a hierarchical prompting strategy across three stages to enhance video generation from a spatial and temporal perspective. (a) **First Frame:** Key semantic prompts guide verifiers to establish consistency for subsequent frames. (b) **Intermediate Frames:** Dynamic prompts in test verifiers  $\mathcal{V}$  focus on action description and motion continuity, building on the first frame’s context. (c) **Final Frames:** Verifiers assess overall text-video alignment, mitigating temporal artifact accumulation. To ensure smooth transitions, we introduce adaptive branching by injecting additional initial noise samples when switching stages, improving temporal coherence and diversity.

**Heuristic pruning.** Throughout the generation process, we model the video as the dynamic growth of a forest, where trees represent possible generation paths and are expanded

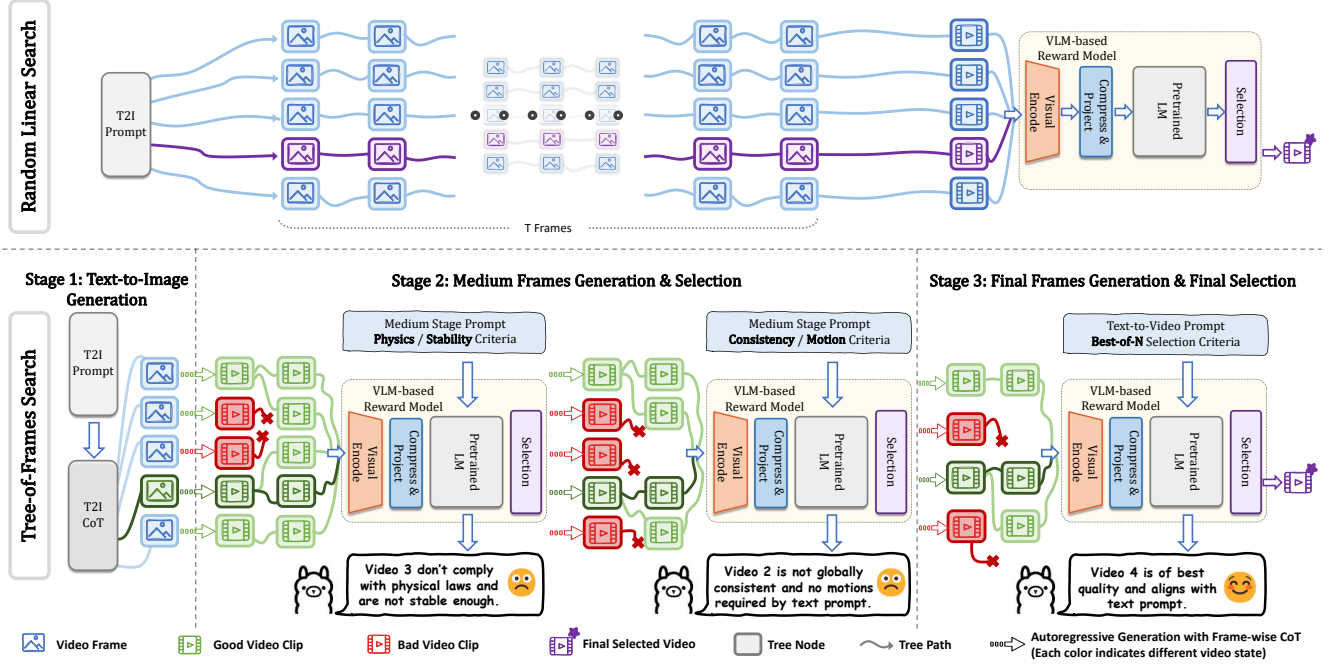


Figure 2. **Pipeline of Test-Time Scaling for Video Generation.** *Top: Random Linear Search* for TTS video generation is to randomly sample Gaussian noises, prompt the video generator to generate multiple videos in a linear manner, and select the best video using test verifiers. *Bottom: Tree of Frames (ToF) Search* for TTS video generation divides the process into three stages: (a) image-level alignment for later frames; (b) dynamic prompt in test verifiers  $\mathcal{V}$  for motion stability and physical plausibility, guiding heuristic search; (c) overall video quality assessment and selection based on text prompt alignment.

and pruned over time. We start by generating  $N$  initial frames, corresponding to the roots of  $N$  trees. Each time step  $t \in [0, T - 1]$  corresponds to a layer in the tree, with each frame acting as a node. At each time step, every surviving parent node  $k_{t-1}$  dynamically branches into  $b_t$  candidate continuations. All  $k_{t-1} \cdot b_t$  nodes are evaluated using a heuristic reward score  $H$  by test verifiers  $\mathcal{V}$ , after which only the top  $k_t$  nodes are retained for further growth. The heuristic score balances local frame quality with global consistency to prioritize the most promising paths. By iteratively applying adaptive branching and heuristic pruning, ToF search efficiently explores the search space while maintaining manageable compute costs. See Algorithm 1 for more details.

**Complexity analysis.** The time complexity of growing one level of the tree is:

$$O(k_{t-1}b_t + b_t \log(k_{t-1}b_t)). \quad (4)$$

Here, generating  $k_{t-1}b_t$  nodes takes  $O(k_{t-1}b_t)$  time, and heap sorting for pruning costs  $O(b_t \log(k_{t-1}b_t))$ . By iteratively applying dynamic branching and heuristic pruning, the deepest leaf nodes in the forest correspond to the final frames of the video, with the path to those nodes representing the optimal video sequence. The overall time complex-

ity of this process is:

$$O(k_0 + \sum_{t=1}^{T-1} k_{t-1}b_t + b_t \log(k_{t-1}b_t)). \quad (5)$$

In practice, we set  $k_0 = N$  and a branching limit  $b_i \leq b = 2$ . In the worst-case scenario, assuming  $b_i = b = 2$  for all  $i$ , the resulting time complexity is:

$$O(N + TN + 2T \log(N)) = O(TN). \quad (6)$$

This complexity aligns with random linear search. In practice, branching occurs only at specific prompt stages, ensuring diverse and stable transitions. Consequently,  $b_t$  is mostly 1, simplifying Eq. 5 to  $O(N + T)$ . Compared to random linear search’s quadratic complexity, our proposed ToF significantly reduces computational costs while maintaining high sample diversity. The logarithmic dependency on  $N$  ensures efficient scaling. Dynamic branching further balances early exploration and later convergence. For detailed analysis, see supplementary materials.

### 3.4. Multi-Verifiers

Beyond test-time scaling in policy models, previous research [24, 35, 63] has demonstrated that applying test-time scaling to generative verifier models can significantly

---

**Algorithm 1** Tree-of-Frames (ToF) Search

---

**Require:** Initial number of roots  $N$ , maximum tree depth  $T$ , branching factors  $\{b_t\}_{t=1}^T$ , pruning sizes  $\{k_t\}_{t=0}^T$ , heuristic score  $H$  by test verifier  $\mathcal{V}$ , video generator  $\mathcal{G}$  with image-level scaling, noise distribution  $\mathcal{N}$

**Ensure:** Video path  $\hat{v}$  with the highest heuristic score

```
1: Initialize empty priority queue  $Q$ 
2: for  $i = 1$  to  $N$  do
3:   Sample initial noise  $z^{(i)} \sim \mathcal{N}$ 
4:   Initial root frame  $f_0^{(i)} \leftarrow z^{(i)}, 0$ 
5:   Enqueue  $(f_0^{(i)}, \text{score} = 0, \text{path} = \{f_0^{(i)}\})$  into  $Q$ 
6: end for
7: for  $t = 1$  to  $T$  do
8:   Initialize empty list  $\mathcal{C} \leftarrow \{\}$ 
9:   for  $j = 0$  to  $k_{t-1}$  do
10:    Dequeue node  $(f, s, p)$  from  $Q$ 
11:    for  $m = 1$  to  $b_t$  do
12:      Generate continuation  $f_m \leftarrow \mathcal{G}(f, t)$ 
13:      Compute heuristic reward  $h_m \leftarrow H(f_m, t)$ 
14:      Add  $(f_m, s + h_m, p \cup \{f_m\})$  to  $\mathcal{C}$ 
15:    end for
16:  end for
17:  Heap sort  $\mathcal{C}$  by total score in descending order
18:  Clear  $Q$ 
19:  for  $n = 1$  to  $k_t$  do
20:    Enqueue the  $n$ -th top node from  $\mathcal{C}$  into  $Q$ 
21:  end for
22: end for
23: Final verify  $(\hat{f}, \hat{s}, \hat{v}) \leftarrow \arg \max_{(f, s, v) \in \mathcal{C}} s$ 
24: return  $\hat{v}$ 
```

---

enhance performance. This improvement can be achieved through methods such as majority voting with a single verifier model [35] or by ensembling multiple verifiers [24]. To further boost the performance of test-time scaling in video generation, we employ a mixture of different verifiers to mitigate biases and select the best videos from the candidates:

$$\begin{aligned} \hat{i} &= \arg \max_{0 \leq i < n} (H(f^{(i)})) \\ &= \arg \max_{0 \leq i < n} \left( \frac{1}{|\mathcal{M}|} \sum_{v \in \mathcal{M}} c_v \text{Rank}_v(f^{(i)}) \right), \end{aligned} \quad (7)$$

where  $\mathcal{M}$  is the set of test verifiers,  $\text{Rank}_v$  indicates the score ranking assigned by verifier  $v \in \mathcal{M}$  to the  $i$ th candidate video  $f^{(i)}$ ,  $c_v$  denotes the weight associated with verifier  $v$ ,  $n$  is the total number of sampled candidates, and  $\hat{i}$  is the index of the candidate with the highest score. This approach ensures the robustness of test-time scaling and yields better performance gains.

## 4. Experiment

### 4.1. Experiment Setup

**Video Generation Models.** We evaluate our TTS strategy (*i.e.*, random linear search and ToF search) using six popular open-sourced pre-trained video generation models, including three diffusion-based models (OpenSora-v1.2 [64], CogVideoX-2B, and CogVideoX-5B [58]) and three autoregressive models (NOVA [7], Pyramid-Flow (SD3), and Pyramid-Flow (FLUX) [16]). These models span a parameter range from 0.6B to 5B.

**Test Verifiers.** To obtain reasonable feedback and provide the heuristic score  $H$  in different stages, we leverage three multi-modal reward models specific to video generation (*i.e.*, VisionReward [56], VideoScore [10], and VideoLLaMA3 [61]) to assess generated video quality under two search algorithms. VisionReward [56] is designed to capture human preferences across multiple dimensions (29 weighted questions for evaluation), while VideoScore [10] is initialized from LMM and trained on a dataset containing human-provided multi-aspect scores to assess video quality. VideoLLaMA3 [61] is a multimodal foundation model that exhibits state-of-the-art image and video understanding. For comparison, we use metrics VBench [14] as a ground-truth "verifier" to demonstrate the upper bound achievable by test verifiers.

**Details of Search Algorithms.** We conduct experiments on two search algorithms assessed on VBench [14]. For the random linear search, experiments are conducted on 6 video generation models using various verifiers, where the initial sample noise level is incremented from 1 to 30 for each trial. In the case of ToF search, the method is applied to 3 autoregressive models using the best-performing multiple verifier where the initial sample noise is varied from 1 to 7.

**Metrics.** To quantify text-to-video generation performance, we utilize VBench [14], a comprehensive benchmark evaluating motion quality and semantic alignment across 16 dimensions. For computational cost, we extend the number of function evaluations (NFE)[33, 49, 59] from image to video generation. NFE is defined as the product of total denoising steps and the temporal length of latent embeddings, accounting for the video's temporal dimension in inference costs.

### 4.2. Analysis of Experimental Results

**TTS consistently yields stable performance gains across different video generation models.** We conducted a series of random linear search experiments across multiple video generation models using different verifiers. In these experiments, the final video outputs were evaluated with the VBench [14] total score. Figure 3 demonstrates that as the inference computational budget increases, all video generation models exhibit improved performance across differ-

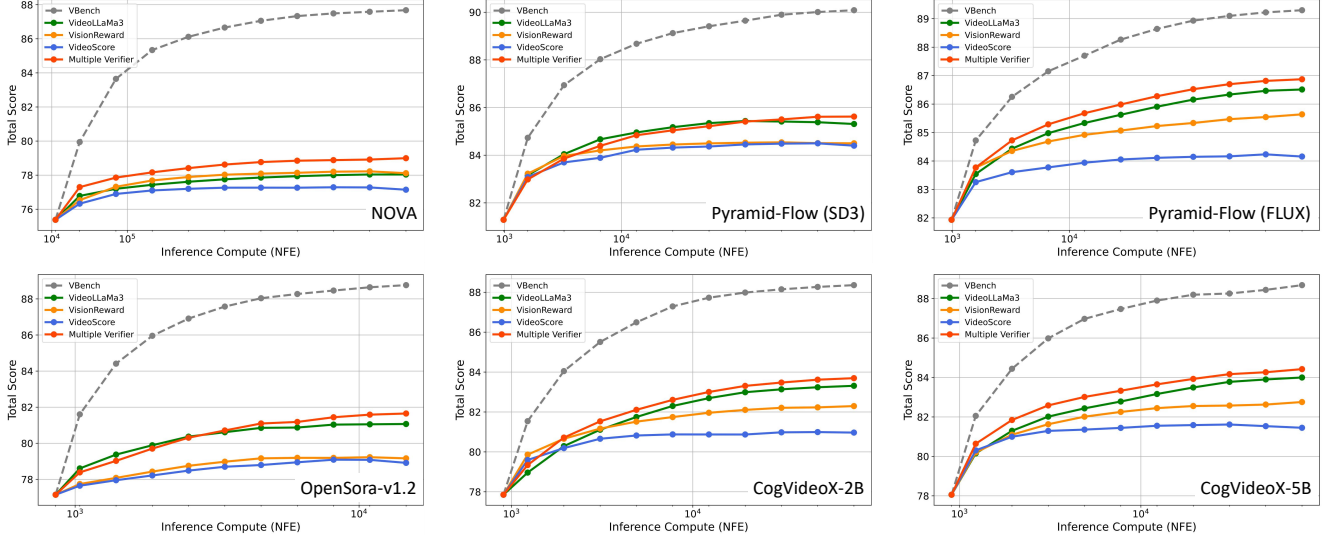


Figure 3. **Performance of random linear search on different video models and verifiers.** The initial points of the curves represent the random sample results without TTS. The models are arranged in order of increasing parameter count from left to right; different colored curves represent the performance trends under various verifiers, and the gray dashed line corresponds to the ground-truth verifier VBench.

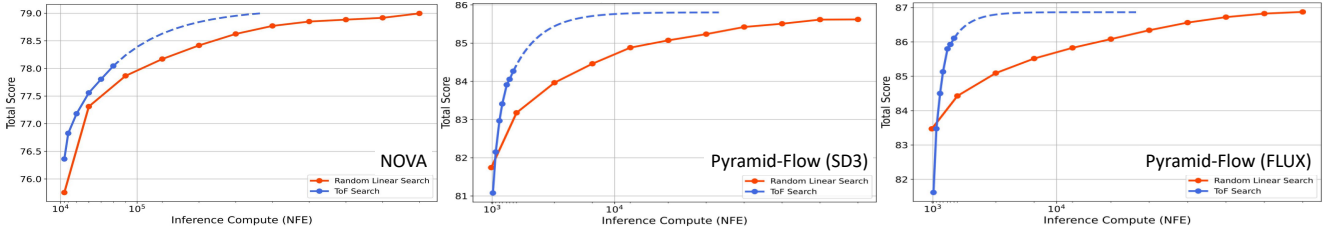


Figure 4. **Comparison between random linear search and ToF search.** The red curve represents random linear search. The blue curve represents ToF search, with the dashed line being the predicted curve from a geometric series decay approximation. Curve fitting reveals that similar subsequent trends tend to converge to an upper limit.

ent verifiers, eventually approaching a convergence limit. This finding indicates that the TTS strategy can effectively guide the search process and significantly enhance generation quality. Moreover, when comparing different verifiers applied to the same video model, we observe varying growth rates in their performance curves, suggesting that each verifier emphasizes different evaluation aspects.

**Multiple verifiers can further boost the curve of TTS.** Beyond the test-time scaling in video generation models, we ensemble the multiple verifiers in Figure 3 that can further boost the performance of test-time scaling in video generation. Such a mixture of different verifiers can also mitigate biases and select the best video from the candidates.

**Advanced foundation models offer significant potential for improvement with TTS.** Additionally, comparative analysis across video models in Figure 3 and Table 1 reveals that lightweight models (*e.g.*, NOVA) exhibit only marginal performance improvements, whereas larger models (*e.g.*, CogVideoX-5B) benefit from a substantially wider search

space and thus achieve more significant enhancements. This observation underscores the potential of larger models to leverage the TTS strategy more effectively, thus producing higher-quality video generation under increased computational budgets.

**ToF Search is more efficient and superior to the random linear search.** We implement the ToF search in three autoregressive models and conduct a comparison experiment with the random linear search and ToF search in Figure 4. We observe that the ToF search achieves comparable performance at a much lower computational cost, highlighting its high efficiency. To minimize the significant differences in computational costs among models of different sizes, We also show quantitative results of GFLOPs in Table 2.

**Performance across most dimensions can be greatly improved with TTS.** The complexity of prompts across diverse benchmark dimensions is a key component in video TTS. We conduct experiments to quantitatively evaluate the performance improvement of different models using TTS



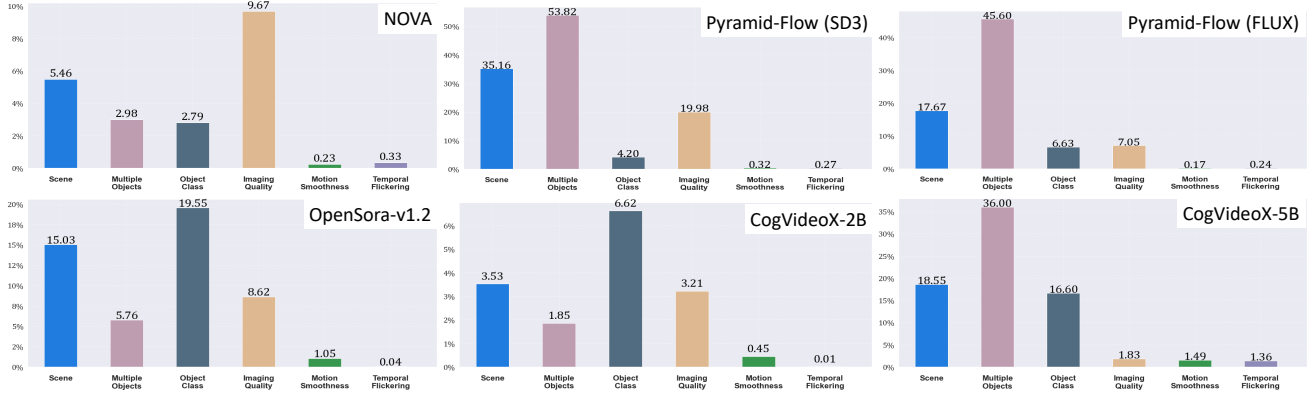


Figure 5. **Qualitative TTS performance improvement ratio on different complexities of prompts** across different video generation models across diverse benchmark dimensions of Vbench.

Table 1. **Quantitative Performance Comparison** on VBench across different video generation models.

	Total Score	Quality Score	Semantic Score	Object Class	Scene	Multiple Objects
<i>Diffusion-based Models</i>						
<b>CogVideoX-5B</b>	81.61	82.75	77.04	85.23	53.20	62.11
+ TTS	84.42 <sup>+3.44%</sup>	84.32 <sup>+1.90%</sup>	<b>84.83</b> <sup>+10.1%</sup>	99.38 <sup>+16.6%</sup>	<b>63.07</b> <sup>+18.6%</sup>	84.47 <sup>+36.0%</sup>
<b>CogVideoX-2B</b>	80.91	82.18	75.83	83.37	51.14	62.63
+ TTS	83.89 <sup>+3.68%</sup>	85.27 <sup>+3.76%</sup>	78.39 <sup>+3.38%</sup>	88.89 <sup>+6.62%</sup>	52.94 <sup>+3.52%</sup>	63.79 <sup>+1.85%</sup>
<b>OpenSora-v1.2</b>	79.76	81.35	73.39	82.22	42.44	63.34
+ TTS	81.65 <sup>+2.37%</sup>	81.90 <sup>+0.68%</sup>	80.63 <sup>+9.87%</sup>	98.29 <sup>+19.5%</sup>	48.82 <sup>+15.0%</sup>	66.99 <sup>+5.76%</sup>
<i>Autoregressive Models</i>						
<b>Pyramid-Flow (SD3)</b>	81.72	84.74	69.62	86.67	43.20	50.71
+ TTS	85.31 <sup>+4.39%</sup>	86.84 <sup>+2.48%</sup>	79.21 <sup>+13.8%</sup>	90.31 <sup>+4.20%</sup>	58.39 <sup>+35.2%</sup>	78.00 <sup>+53.8%</sup>
<b>Pyramid-Flow (FLUX)</b>	81.61	84.11	71.61	93.49	47.65	61.08
+ TTS	<b>86.51</b> <sup>+5.86%</sup>	<b>87.50</b> <sup>+3.26%</sup>	82.56 <sup>+18.6%</sup>	<b>99.69</b> <sup>+6.63%</sup>	56.07 <sup>+17.7%</sup>	<b>88.93</b> <sup>+45.6%</sup>
<b>NOVA</b>	78.56	83.79	57.63	91.36	45.22	67.87
+ TTS	79.80 <sup>+1.58%</sup>	84.99 <sup>+1.43%</sup>	59.03 <sup>+2.43%</sup>	93.91 <sup>+2.79%</sup>	47.69 <sup>+5.46%</sup>	69.89 <sup>+2.98%</sup>

Table 2. Inference-time scaling cost comparison on GFLOPs.

Methods	Linear Search	ToF Search
<b>Pyramid-Flow(FLUX)</b>	$5.22 \times 10^7$	$1.62 \times 10^7$
<b>Pyramid-Flow(SD3)</b>	$3.66 \times 10^7$	$1.13 \times 10^7$
<b>NOVA</b>	$4.02 \times 10^6$	$1.41 \times 10^6$

methods across various dimensions (See Figure 5 and Table 1). As ToF and random linear search can achieve a similar convergence score during test-time scaling, we choose the better score for (+TTS). We find that for common prompt sets (e.g., Scene, Object) and easily assessable categories (e.g., Imaging Quality), TTS methods achieve significant improvements across different models.

**A few dimensions heavily rely on the capabilities of foundation models, making improvements challenging for TTS.** However, for some hard-to-evaluate latent properties (e.g., Motion Smoothness, Temporal Flickering), the improvement is less pronounced. This is likely because Motion Smoothness requires precise control of motion trajec-

tories across frames, which is challenging for current video generation models. Temporal Flickering, on the other hand, involves maintaining consistent appearance and intensity over time, which is difficult to precisely assess with recent VLM based verifiers. (See Figure 5)

## 5. Conclusion

In conclusion, this study presents a novel framework for test-time scaling in video generation, redefining it as a search problem for optimal video trajectories. We build the search space in TTS by test-time verifiers and to provide feedback and employ heuristic algorithms like random linear search and the more efficient ToF search algorithm. Extensive experiments demonstrate that scaling the search space can boost the video performance across various video generation models, and our proposed ToF search can significantly reduce scaling cost when achieving high-quality video outputs. This framework opens new avenues for research into efficient test-time optimization strategies in video generation.



**Acknowledgments:** This work was supported by the National Natural Science Foundation of China under Grant 62206147, and in part by 2024 Tencent AI Lab Rhino-Bird Focused Research Program.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2
- [2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3
- [3] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2025. 3
- [4] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, pages 7310–7320, 2024. 3
- [5] Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*, 2023. 2
- [6] Haikang Deng and Colin Raffel. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. *arXiv preprint arXiv:2310.09520*, 2023. 2
- [7] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024. 3, 6
- [8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2
- [9] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with cot? let’s verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025. 3, 4
- [10] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyang Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2105–2123, 2024. 6
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 3
- [12] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2
- [13] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 3
- [14] Ziqi Huang, Yanan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 6
- [15] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 2
- [16] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 3, 6
- [17] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-lavit: unified video-language pre-training with decoupled visual-motional tokenization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 22185–22209, 2024. 3
- [18] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2
- [19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2
- [20] Maxim Khanov, Jirayu Burapachee, and Yixuan Li. Args: Alignment as reward-guided search. *arXiv preprint arXiv:2402.01694*, 2024. 2
- [21] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vignesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: a large language model for zero-shot video generation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 25105–25124, 2024. 3
- [22] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuo Zhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2

- [23] Chengxuan Li, Di Huang, Zeyu Lu, Yang Xiao, Qingqi Pei, and Lei Bai. A survey on long video generation: Challenges, methods, and prospects. *arXiv preprint arXiv:2403.16407*, 2024. 2
- [24] Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. Multi-agent verification: Scaling test-time compute with multiple verifiers, 2025. 5, 6
- [25] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023. 2
- [26] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. *arXiv preprint arXiv:2406.05338*, 2024. 3
- [27] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024.
- [28] Fangfu Liu, Hanyang Wang, Shunyu Yao, Shengjun Zhang, Jie Zhou, and Yueqi Duan. Physics3d: Learning physical properties of 3d gaussians via video diffusion. *arXiv preprint arXiv:2406.04338*, 2024. 3
- [29] Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*, 2025. 2, 3
- [30] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8599–8608, 2024. 2
- [31] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024. 3
- [32] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025. 2
- [33] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps, 2025. 3, 6
- [34] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023. 2
- [35] Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models, 2024. 5, 6
- [36] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 2
- [37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 3
- [38] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2
- [39] 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. 2
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 3
- [41] Yangjun Ruan, Chris J Maddison, and Tatsunori B Hashimoto. Observational scaling laws and the predictability of language model performance. *Advances in Neural Information Processing Systems*, 37:15841–15892, 2025. 2
- [42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2
- [43] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. 2
- [44] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. 3
- [45] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020. 2
- [46] Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion, 2024. 3
- [47] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 2
- [48] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan. Llamav-o1: Rethinking step-by-step visual reasoning in llms, 2025. 3
- [49] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image

- generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 6
- [50] Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023. 2
- [51] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 2
- [52] Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025. 2
- [53] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2
- [54] Yuxi Xie, Anirudh Goyal, Wenye Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024. 2
- [55] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025. 3
- [56] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. *arXiv preprint arXiv:2412.21059*, 2024. 6
- [57] Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022. 3
- [58] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2, 3, 6
- [59] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vignesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 6
- [60] Ailing Zeng, Yuhang Yang, Weidong Chen, and Wei Liu. The dawn of video generation: Preliminary explorations with sora-like models. *arXiv preprint arXiv:2410.05227*, 2024. 2
- [61] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 6
- [62] Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Bqing Qi, and Bowen Zhou. Openprm: Building open-domain process-based reward models with preference trees. In *The Thirteenth International Conference on Learning Representations*, 2025. 2
- [63] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2025. 5
- [64] 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, 2024. 6
- [65] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Hao-han Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023. 2