Video-T1: Test-Time Scaling for Video Generation

Fangfu Liu[1]*, Hanyang Wang[1]*, Yimo Cai[1], Kaiyan Zhang[1], Xiaohang Zhan[2], Yueqi Duan[1]†
[1]Tsinghua University, [2]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 scaling 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, 60], 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 [62]. 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 [62]. 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, 48, 54].

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 and quantitative results in Figure 2). 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

Test-Time Scaling in LLMs. Recent advancements have demonstrated the effectiveness of test-time scaling (TTS) methods such as chain-of-thought prompting [34, 55], outcome reward models, and process reward models [25, 52, 64] 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, 44]. Moreover, strategies like self-consistency [5, 53], best-of-N [36, 46], Monte Carlo Tree Search [56, 67], and Reward-guided Search [6, 20] employ diverse generation techniques and sophisticated aggregation methods, often facilitated by process reward models. These approaches help in produc-

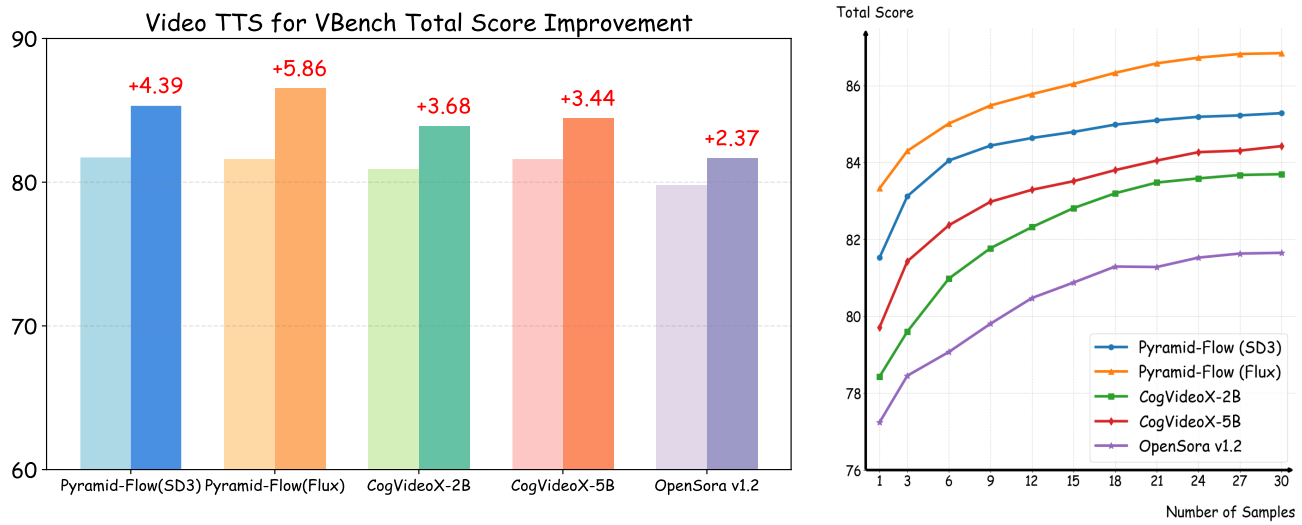

Figure 2. **Results of Test-Time Scaling for Video Generation.** As the number of samples in the search space increases by scaling test-time computation (TTS), the models' performance exhibits consistent improvement (In the bar chart, light colors correspond to the results without TTS, while dark colors represent the improvement after TTS.).

ing diverse and integrated outputs. Additionally, DeepSeek R1 [8] utilizes outcome-based reinforcement learning techniques, like group relative policy optimization [43], to enhance the reasoning capabilities of pre-trained models. 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.

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) [49, 57] 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].

Video Generation. Efficient and high-quality video generation has attracted increasing attention due to its wide applications in areas [26–28, 47]. 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, 60] 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, 59] 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, 45]. 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 es-

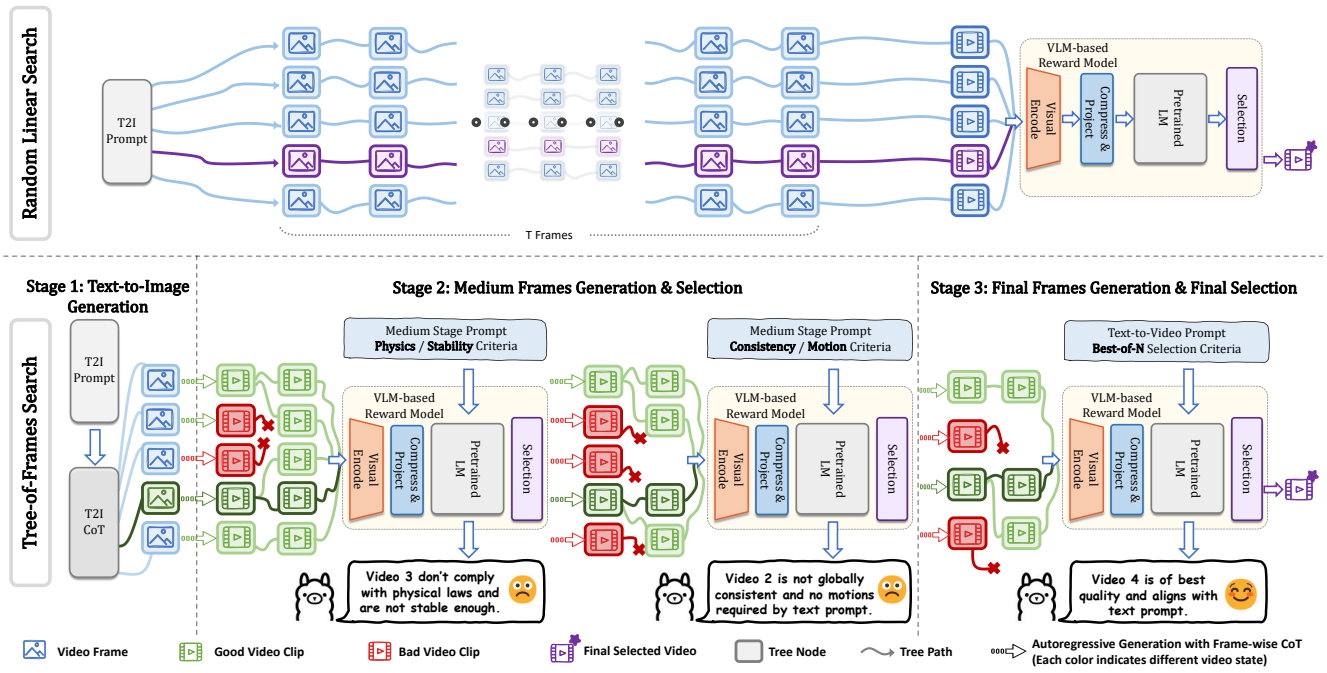

Figure 3. **Pipeline of Tes-Time Scaling for Video Generation.** *Top:* **Random Linear Search** for TTS video generation is to randomly sample Gaussian noises, prompt the video generator to generate sequential of video clips through step-by-step denoising in a linear manner, and select the highest score form the test verifiers. *Bottom:* **Tree of Frames (ToF) Search** for TTS video generation is to divide the video generation process into three stages: (a) the first stage performs image-level alignment that influences the later frames; (b) the second stage is to apply dynamic prompt in test verifiers $\mathcal{V}$ to focus on motion stability, physical plausibility to provide feedback that guides heuristic searching process; (c) the last stage assesses the overall quality of the video and select the video with highest alignment with text prompts.

sential 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:

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

  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:

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

  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:

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

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 3), as it performs step-by-step denoising in a linear manner along the noise dimension. In this search algorithm, each noise sample deterministically corresponds to a determined video output, and 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 increasing 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 2), which leverages the sequential generation capability of autoregressive models (unlike diffusion models that denoise the entire video sequence simultaneously), introducing inference-time reasoning along the temporal dimension. This approach enables a more flexible and scalable video generation process, structured into three distinct stages: 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 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 goal of three stages, we meticulously design three key techniques in ToF search algorithm: *image-level alignment*, *hierarchical prompting*, and *heuristic pruning*.

Algorithm 1 Random Linear Search

Require: Number of noise samples N, video frame length T, verifier $\mathcal{V}$, video generator $\mathcal{G}$ and decoder $\mathcal{D}$, Gaussian noise distribution $\mathcal{N}$

Ensure: Video $\hat{v}$ with the highest verifier score

Initialize empty set $\mathcal{C} \leftarrow \{\}$
for $i = 1$ to N **do**
  Sample initial noise $z^{(i)} \sim \mathcal{N}$
  Initialize $x_0^{(i)} \leftarrow z^{(i)}$
  for $t = 0$ to $T-1$ **do**
    $x_t^{(i)} \leftarrow \mathcal{G}(x_{t-1}^{(i)}, t)$
  end for
  Decode video $v^{(i)} \leftarrow \mathcal{D}(x_T^{(i)})$
  Compute score $s^{(i)} \leftarrow \mathcal{V}(v^{(i)})$
  Add $(v^{(i)}, s^{(i)})$ to $\mathcal{C}$
end for
Final verify $\hat{v} \leftarrow \arg\max_{(v,s)\in\mathcal{C}} s$
return $\hat{v}$

Image-level alignment. Different from LLMs, video generation involves both spatial and temporal dimensions. Along the spatial axis, video frames are generated through step-wise denoising employed in diffusion models. Inspired by the Chain-of-Thought (CoT) reasoning mechanism in image generation [9], we introduce a progressive evaluation strategy at the frame level to dynamically scale computation during the denoising process. Specifically, during the denoising of each frame, a potential test verifier evaluates whether the partially denoised image has reached sufficient clarity for reliable assessment. Early stage frames often remain too blurry for a meaningful evaluation, which could mislead the scoring of frame quality. Once the frame reaches a visually informative state, the model further assesses its potential to evolve into a high-quality final image. By performing early rejection of low-potential candi-

dates and allocating compute toward promising trajectories, image-level scaling ensures more efficient use of resources during inference.

Hierarchical prompting. From a spatial perspective, each video frame is generated as an independent image. However, different frames play distinct roles in shaping the video's narrative and temporal coherence. With the analysis above, we design a hierarchical prompting strategy in three different stages: (a) for the first frame, we extract the key prompts related to core semantics (*e.g.*, color, object count, relative positions) to prompt the verifiers to provide feedback that determines the consistency and correctness of subsequent frames; (b) for intermediate frames, we apply dynamic prompt in test verifiers $\mathcal{V}$ to focus on action description and motion continuity based on the context established by the first frame; (c) lastly, we prompt test verifiers to assess the overall quality of the final text-video alignment while mitigating the risk of accumulating temporal artifacts from excessive motion. To maintain smooth transitions across these distinct stages, we introduce *adaptive branching* by injecting additional initial noise samples when switching between stages, thereby 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 and pruned over time. Similar to random linear search, 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 2 for more details.

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

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

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 complexity of this process is:

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

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:

$$O(N + TN + 2T \log(N)) = O(TN). \tag{6}$$

This complexity is consistent with the random linear search. In our practical experiments, we perform branching operations only at specific prompt stages to ensure a diverse and stable transition between stages. Consequently, b_t remains 1 for most timesteps, and Eq. 5 can simplify to $O(N + T)$. Compared to the quadratic complexity of random linear search, our proposed ToF significantly reduces computational costs while maintaining high sample diversity. The logarithmic dependency on N ensures efficient scaling. Additionally, by dynamically adjusting the branching factor, we achieve a better trade-off between exploration in early timesteps and convergence in later stages. For detailed complexity analysis, please refer to supplementary materials.

3.4. Multi-Verifiers

Beyond test-time scaling in policy models, previous research [24, 35, 65] has demonstrated that applying test-time scaling to generative verfier models can significantly 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:

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

where $\mathcal{M}$ is the set of test verifiers, Rank_v indicates the score ranking assigned by verifier $v \in \mathcal{M}$ to the ith 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

Algorithm 2 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

Initialize empty priority queue Q
for $i = 1$ to N **do**
  Sample initial noise $z^{(i)} \sim \mathcal{N}$
  Inital root frame $f_0^{(i)} \leftarrow z^{(i)}, 0$
  Enqueue $(f_0^{(i)}, \text{score} = 0, \text{path} = \{f_0^{(i)}\})$ into Q
end for
for $t = 1$ to T **do**
  Initialize empty list $\mathcal{C} \leftarrow \{\}$
  for $j = 0$ to k_{t-1} **do**
    Dequeue node (f, s, p) from Q
    for $m = 1$ to b_t **do**
      Generate continuation $f_m \leftarrow \mathcal{G}(f, t)$
      Compute heuristic reward $h_m \leftarrow H(f_m, t)$
      Add $(f_m, s + h_m, p \cup \{f_m\})$ to $\mathcal{C}$
    end for
  end for
  Heap sort $\mathcal{C}$ by total score in descending order
  Clear Q
  for $n = 1$ to k_t **do**
    Enqueue the n-th top node from $\mathcal{C}$ into Q
  end for
end for
Final verify $(\hat{f}, \hat{s}, \hat{v}) \leftarrow \arg\max_{(f,s,v)\in\mathcal{C}} s$
return $\hat{v}$

popular open-sourced pre-trained video generation models, including three diffusion-based video models (OpenSora-v1.2 [66], CogVideoX-2B, and CogVideoX-5B [60]) 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 [58], VideoScore [10], and VideoLLaMA3 [63]) to assess generated video quality under two search algorithms. VisionReward [58] is designed to capture human preferences across multiple dimensions (dividing the evaluation into 29 weighted questions), while VideoScore [10] is initialized from LMM and trained on a dataset containing human-provided multi-aspect scores to automatically assess video quality. VideoLLaMA3 [63] 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 the three 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 the performance of text-to-video generation, we use VBench [14], which is a comprehensive benchmark incorporating 16 fine-grained dimensions that evaluate both motion quality and semantic alignment. For the computational cost, we extend the metric of the number of function evaluations (NFE) [33, 50, 61] from image generation to video generation by defining NFE as the product of the total number of denoising steps executed during the generation process and the temporal length of latent embeddings, which takes the temporal dimension of the video into inference computational costs.

4.2. Analysis of Experimental Results

TTS consistently yields stable performance gains across different video generation models. We conduct 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—a composite metric aggregating 16 distinct dimensions of quality (*e.g.*, motion smoothness, semantic alignment, aesthetic quality). Figure 4 demonstrates that as the inference computational budget increases, all video generation models exhibit improved performance across different verifiers, eventually approaching a convergence limit once a certain threshold is reached. This finding indicates that the TTS strategy can effectively guide the search process during test time and significantly enhance generation quality. Moreover, when comparing different verifiers applied to the same video model, we observe varying growth rates and extents in their performance curves. This divergence suggests 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 4 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 4 and Table 2 reveals that lightweight models (*e.g.*, NOVA) exhibit only marginal performance improvements with increased inference effort,

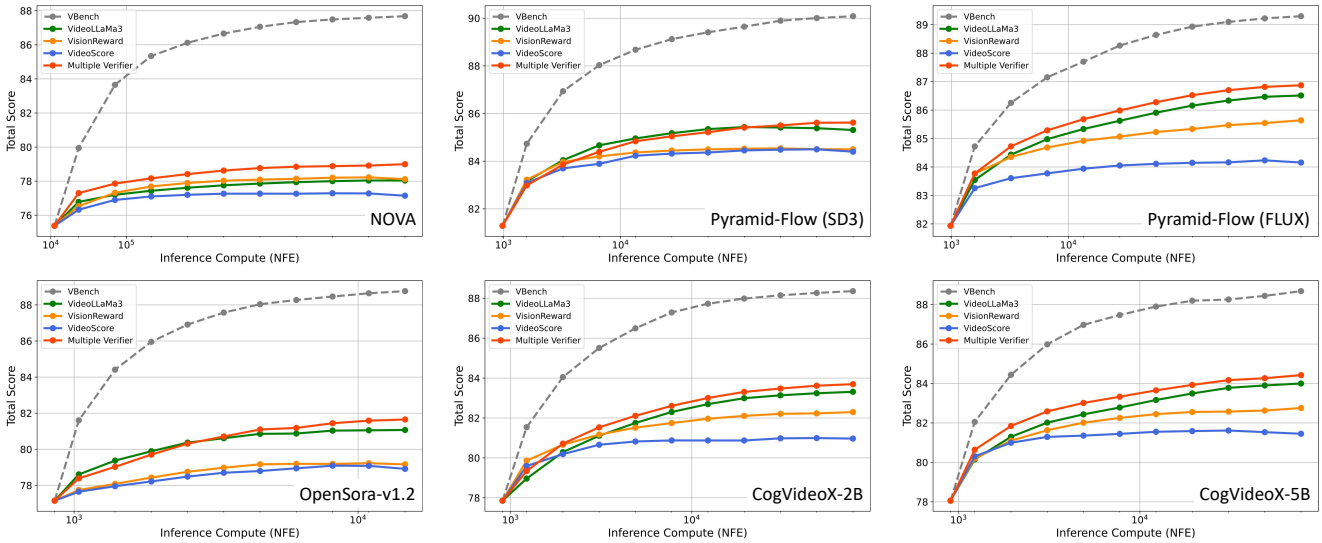

Figure 4. **Performance of random linear search on different video models and verifiers.** The top row displays results for autoregressive models, while the bottom row shows diffusion-based models. The initial points of the curves represent the random video sample results without TTS. The models are 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 baseline established by VBench, which serves as a ground-truth verifier.

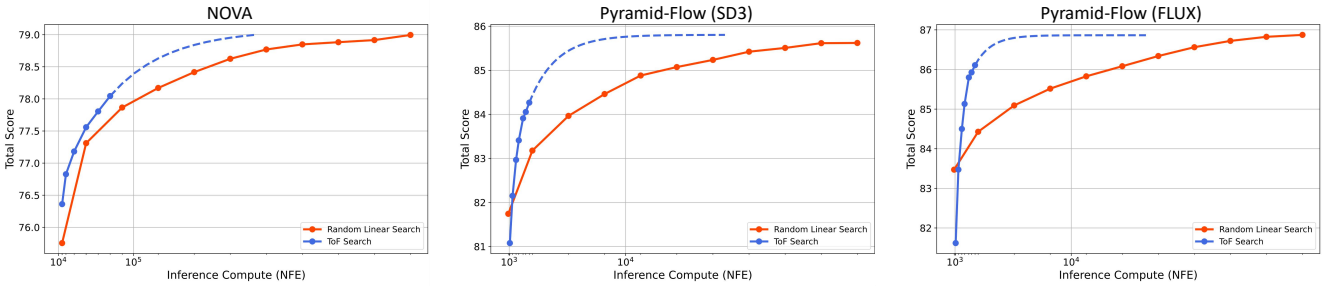

Figure 5. **Comparison between random linear search and ToF search.** The red curve represents random linear search. The blue curve represents ToF search, with the dashed line being the predicted curve from a geometric series decay approximation. Curve fitting reveals that similar subsequent trends tend to converge to an upper limit.

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, thereby yielding higher-quality video generation under increased computational budgets.

Table 1. Inference-time scaling cost comparison on GFLOPs.

Methods	Linear Search	ToF Search
Pyramid-Flow(FLUX)	5.22×10^7	1.62×10^7
Pyramid-Flow(SD3)	3.66×10^7	1.13×10^7
NOVA	4.02×10^6	1.41×10^6

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 5. 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 1. Moreover, larger and better models show higher efficiency, as evidenced by the faster rising speed of the curve.

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 methods across various dimensions (See Figure 6 and Table 2). As ToF and random linear search can achieve

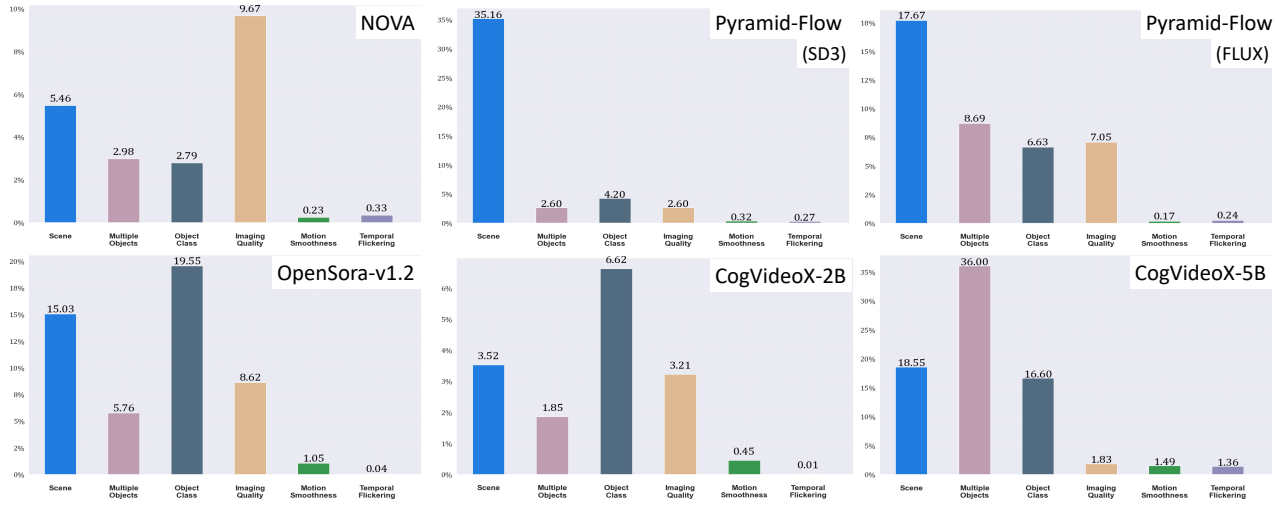

Figure 6. **Qualitative TTS performance improvement ratio on different complexities of prompts** across different video generation models across diverse benchmark dimensions of Vbench.

Table 2. **Quantitative Performance Comparison** on VBench across different video generation models.

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

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 trajectories across frames, which is challenging for current video generation models to achieve. Temporal Flickering, on the other hand, involves maintaining consistent appearance and intensity over time, which is difficult to precisely assess, especially when dealing with complex scenes and dynamic objects. (See Figure 6 and Table 2)

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 in part 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, 7

[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, 3, 13

[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, 5, 13

[10] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan 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. 7

[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, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 7

[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, 7, 14

[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 Burapacheep, 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, Vighnesh 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, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 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. 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, 7

[34] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, 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. 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 langauge 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] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3

[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. 2

[45] 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

[46] 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

[47] 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

[48] 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

[49] 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

[50] 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. 7

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 13

[52] 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

[53] 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

[54] 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

[55] 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

[56] Yuxi Xie, Anirudh Goyal, Wenyue 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

[57] 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

[58] 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. 7

[59] Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022. 3

[60] 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, 7, 14

[61] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 7

[62] 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

[63] 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. 7

[64] Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing 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

[65] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2025. 6

[66] 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. 7

[67] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023. 2

A. More Implementation Details

A.1. More Discussion of Image-level Alignment

In our generation process, inspired by recent work [9], each frame is generated with image-level TTS. Specifically, we employ a two-stage evaluation mechanism. First, a potential assessment reward model examines whether the partially generated frame exhibits sufficient visual clarity for meaningful evaluation and assigns a binary label. If the output is deemed unclear ('no'), the model skips further processing for that frame. Otherwise ('yes'), the frame advances to a secondary evaluation stage where its potential to yield a high-quality final image is assessed. Again, a binary decision is made: if the frame is unlikely to lead to an optimal outcome, the generation path is truncated immediately; if it passes, the synthesis continues to produce the final image.

A.2. More Discussion of Hierarchical Prompting

In our approach to hierarchical prompting, we employ DeepSeek-R1-8B [8], a large language model distilled from the LLaMA-8B [51] architecture. DeepSeek-R1-8B is used to decompose a given input prompt into three distinct hierarchical prompts, each tailored to represent a specific stage of the video sequence:

- Static scene description: The first prompt provides a detailed depiction of the static scene intended for the initial frame, establishing the starting visual context of the sequence.
- Action/motion directions: The second prompt outlines the actions or motion directions that guide the dynamic progression across the intermediate frames, ensuring a coherent evolution of the scene.
- Expected ending state: The third prompt delineates the expected ending state for the concluding frame, defining the desired outcome of the video sequence.

DeepSeek-R1-8B processes the input prompt and generates these three hierarchical prompts, which are then returned as an ordered list of length three. At each stage of the video sequence—initial, intermediate, and final—the test verifier evaluates the generated output with the corresponding hierarchical prompt. This stage-specific assessment ensures that the video content aligns with the intended descriptions, maintaining both accuracy and coherence throughout the sequence. Figure 9 gives an example of hierarchical prompting generation and test-time verification.

A.3. Detailed Complexity Analysis

In our method, the video generation process is modeled as the dynamic growth of a forest, where the trees are branched and pruned over time. Specifically, similar to the linear search, we begin by generating N initial frames, representing the roots of N trees in the forest. Each time step $t \in [0, T-1]$ corresponds to a level in the tree, and each frame is treated as a tree node. We consider the process in which each of the N trees grows by adding one level of nodes. At each time step, the k_{t-1} surviving parent nodes dynamically branch into b_t possible continuations. We then evaluate all $k_{t-1} \cdot b_t$ nodes using a heuristic reward function H, followed by pruning to retain only the top k_t branches. The time complexity of growing one level of the tree is:

$$O(k_{t-1}b_t + b_t \log(k_{t-1}b_t)). \tag{8}$$

Here, generating $k_{t-1}b_t$ nodes takes $O(k_{t-1}b_t)$ time, the evaluation cost per node is $O(1)$, and the total evaluation time is $O(k_{t-1}b_t)$. 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 complexity of this process is:

$$O(k_0 + \sum_{t=1}^{T-1} k_{t-1}b_t + b_t \log(k_{t-1}b_t)). \tag{9}$$

In practice, we set a branching limit b for dynamic branching, *i.e.*, $b_t \leq b$. In the heuristic pruning step, we use the heuristic reward function H to prune branches that fall below the average value. On average, each pruning step retains $k_t \approx \frac{k_{t-1}b_t}{2} = \frac{k_0 \prod_{i=1}^{t} b_i}{2^t} \leq \frac{k_0 b^t}{2^t}$ branches before k_t drops to 1. Therefore, we have:

$$\begin{aligned} &O(k_0 + \sum_{t=1}^{T-1} k_{t-1}b_t + b_t \log(k_{t-1}b_t)) \\ &= O(k_0 + \sum_{t=1}^{T-1} \frac{k_0 \prod_{i=1}^{t} b_i}{2^t} + b_t \log(\frac{k_0 \prod_{i=1}^{t} b_i}{2^t}) \end{aligned} \tag{10}$$

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

$$O(N + TN + 2T \log(N)) = O(TN). \tag{11}$$

This complexity is consistent with that of the linear search. However, in our actual experiments, we perform branching operations only at specific prompt stages to ensure a diverse and stable transition between stages. Consequently, b_t remains 1 for most timesteps, leading to the following update rule:

$$k_t = \begin{cases} \frac{k_{t-1}b_t}{2}, & 0 < t \leq \log(k_0) \\ 1, & t > \log(k_0). \end{cases} \tag{12}$$

Thus, Eq. 9 can simplify to:

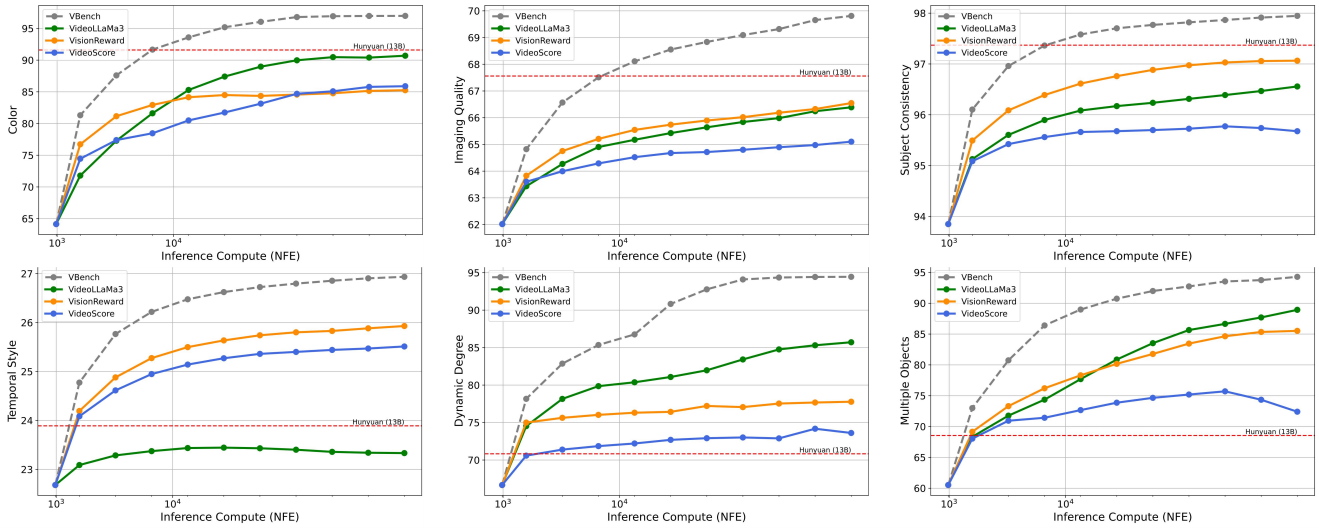

Figure 7. Using TTS, the small model (Pyramid-Flow) achieves scores that are close to, or even exceed, those of the 13B large model (HunyuanVideo) in many dimensions. The gray dashed horizontal line in the figures indicates HunyuanVideo's score in that dimension.

$$
\begin{aligned}
& O(k_0 + \sum_{t=1}^{T-1} k_{t-1} b_t + b_t \log(k_{t-1} b_t)) \\
&= O(k_0 + \sum_{t=\log(k_0)+1}^{T-1} b_t + \sum_{t=1}^{\log(k_0)} \frac{k_0 \prod_{i=1}^{t} b_i}{2^t} + b_t \log(\frac{k_0 \prod_{i=1}^{t} b_i}{2^t})) \\
&= O(k_0 + T - \log(k_0) + k_0 + \log^2(k_0) - \frac{\log 2}{2} log^2(k_0)) \\
&= O(N + T).
\end{aligned} \tag{13}
$$

Compared to the quadratic complexity of linear search, our approach converge at a geometric rate, ultimately achieving linear complexity. It significantly reduces computational costs while maintaining high sample diversity. The logarithmic dependency on N ensures efficient scaling, making our method more suitable for high-dimensional video generation. Additionally, by dynamically adjusting the branching factor, we achieve a better trade-off between exploration in early timesteps and convergence in later stages.

B. More Experiments

B.1. More Quantitative Results

We performed multiple random linear search experiments across various video generation models with different verifiers. Figures 10-12 present more quantitative results on VBench across different dimensions. These indicate that TTS consistently delivers stable performance improvements across dimensions. Moreover, the evaluation accuracy of different verifiers varies across dimensions, justifying our use of multiple verifiers.

B.2. More Qualitative Results

We provide additional visual results for Pyramid-Flow [16] and CogVideoX-5B [60] in Figures 13-18. Each example displays different video outputs as NFE increases with the same prompt input. The results show that video quality and text alignment improve with more TTS samples.

B.3. Comparison with Large Models

Figure 7 compares the outputs of the small model Pyramid-Flow (2B) with TTS and the large model HunyuanVideo (13B) single. As scaling increases, the small model's performance approaches or even surpasses that of the large model in many dimensions.

B.4. Failure Cases

Figure 8 shows failure cases in Pyramid-Flow experiments where TTS does not significantly improve video quality. Increasing inference computation fails to generate reasonable details (*e.g.*, hand movements), indicating that model performance limitations constrain TTS method improvements.

Figure 8. Failure cases on prompt "A person is clapping".

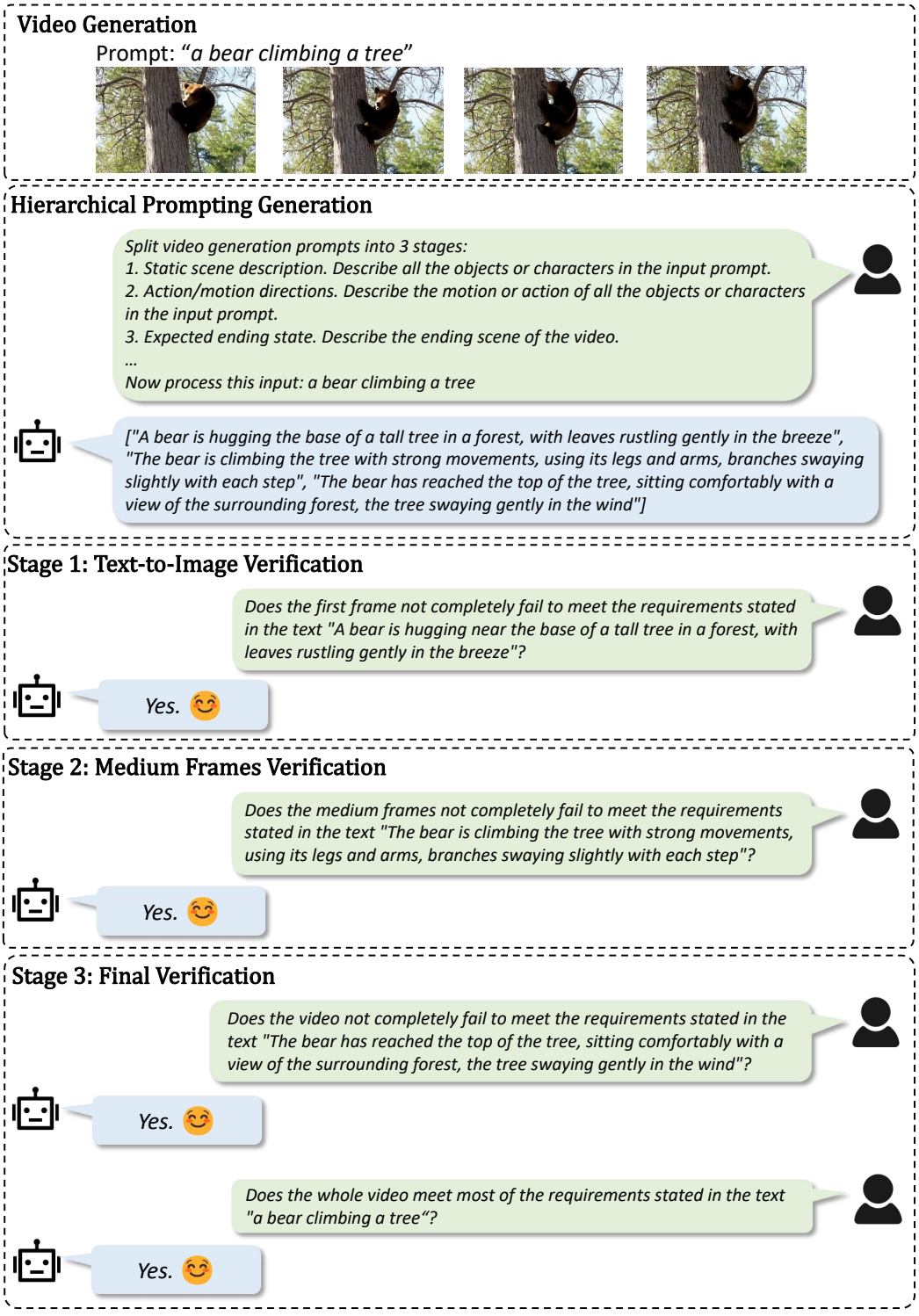

Figure 9. Verifications during TTS process.

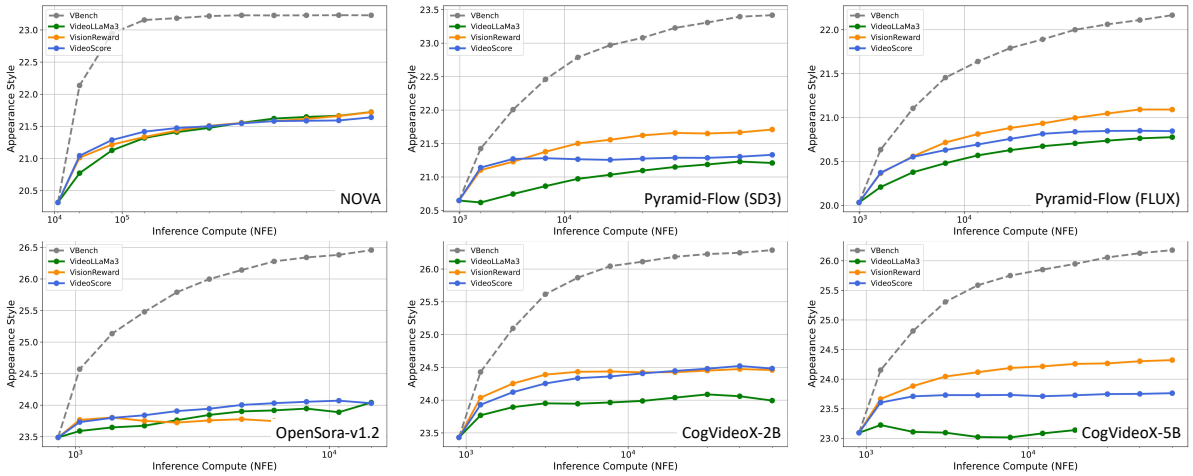

Figure 10. TTS performance on Appearance Style across diverse verifiers.

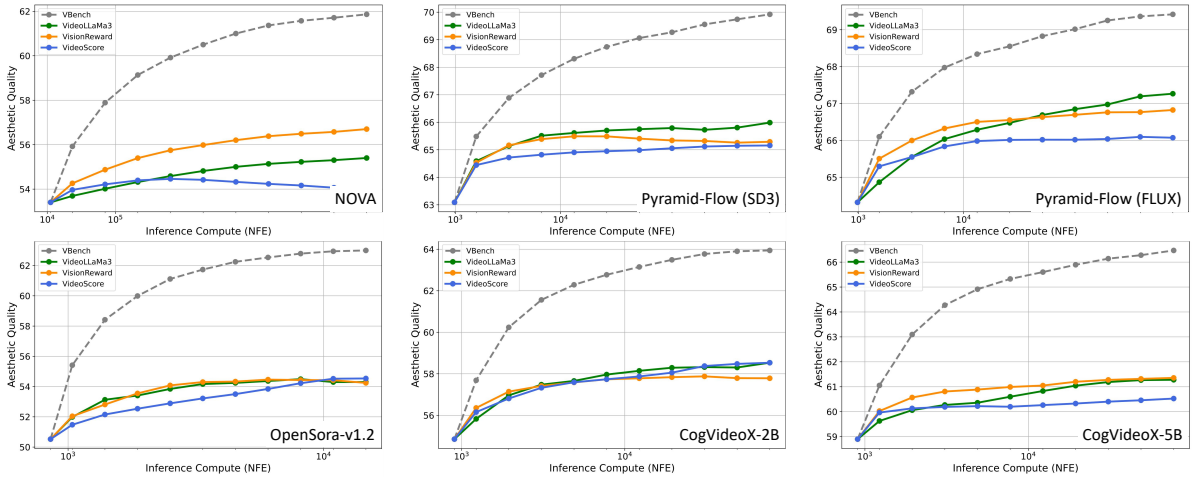

Figure 11. TTS performance on Aesthetic Quality across diverse verifiers.

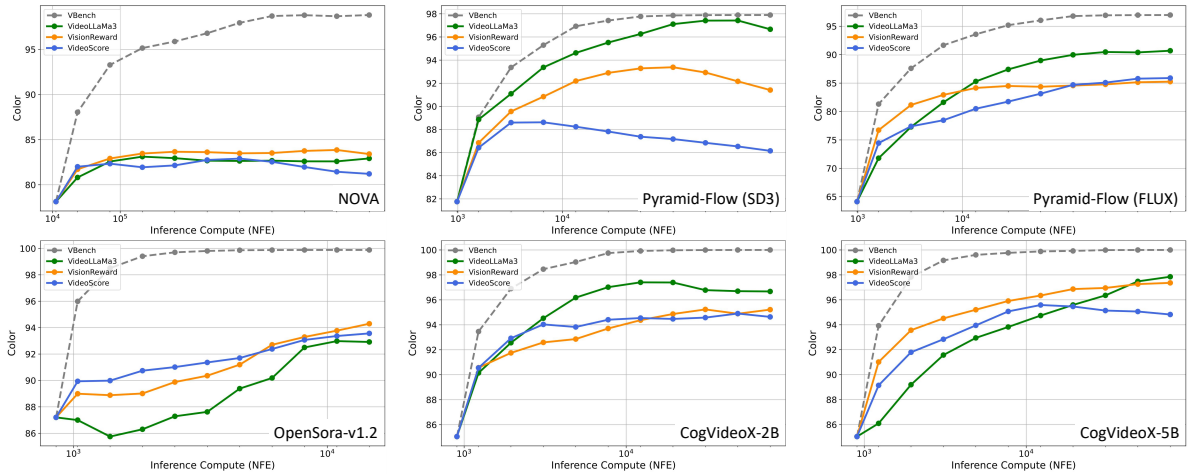

Figure 12. TTS performance on Color across diverse verifiers.

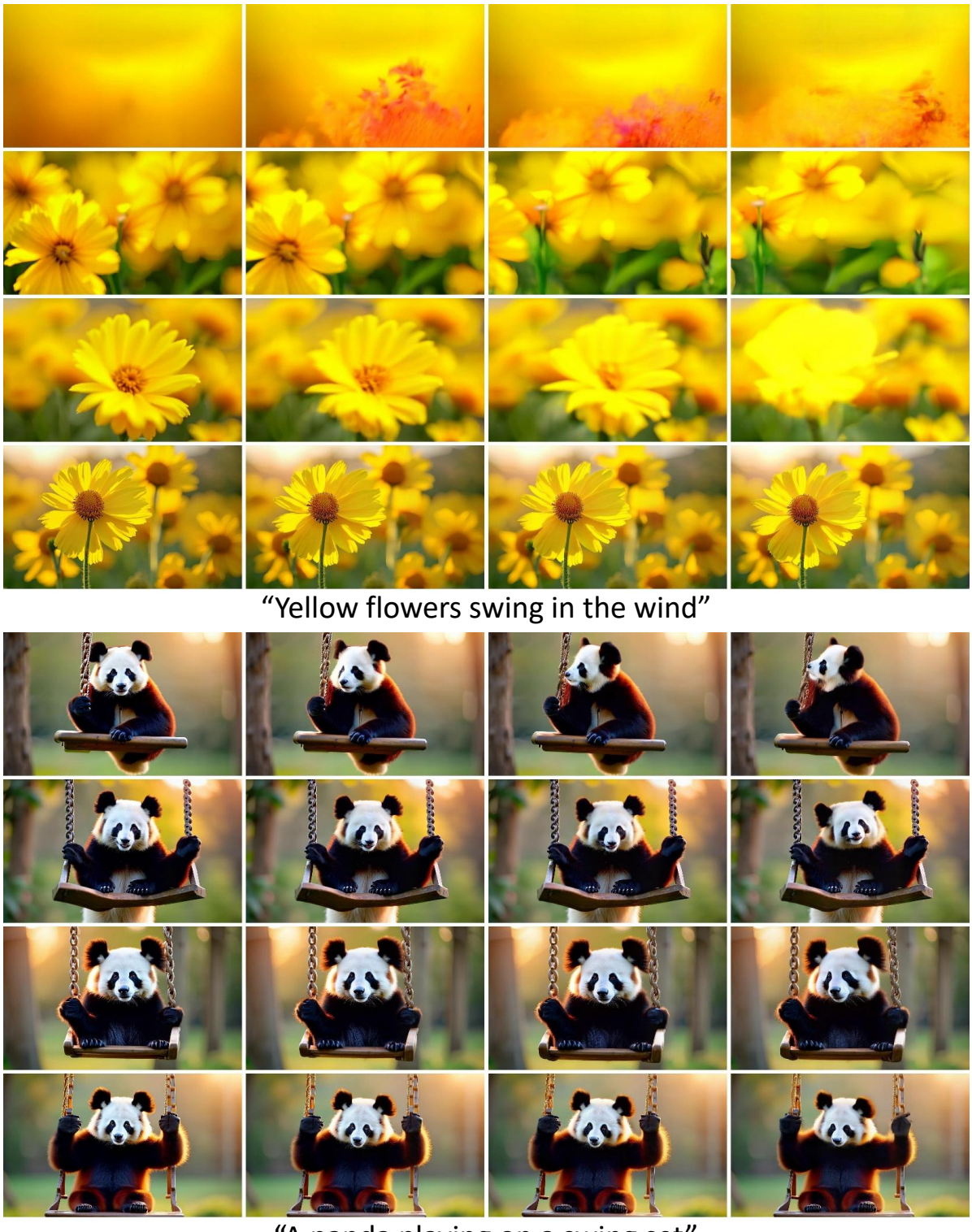

Figure 13. More visual results during TTS process on Pyramid-Flow. From left to right, each row of frames are extracted from a video sequence. From top to bottom, each row represents the output video results of TTS with an increasing number of samples.

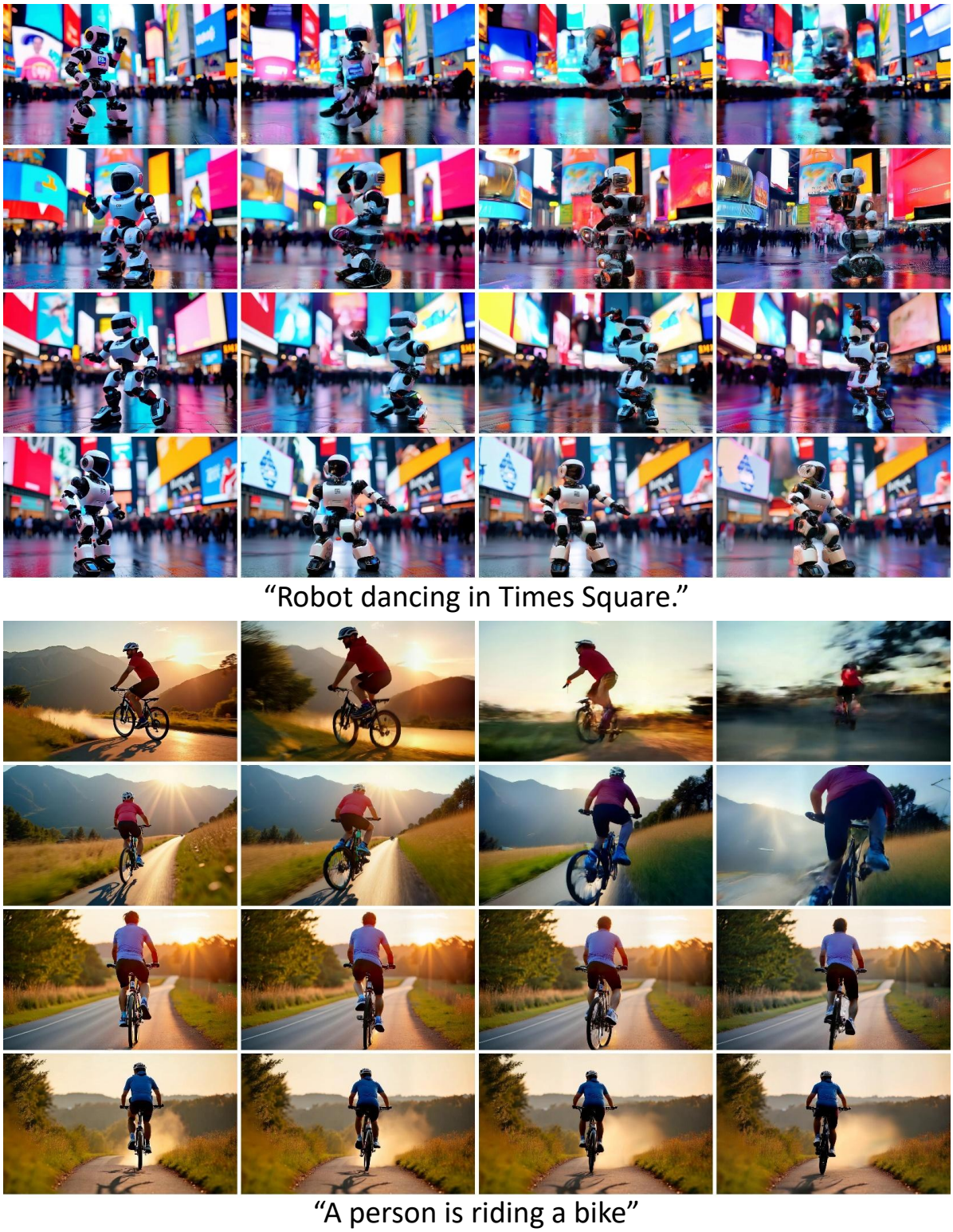

Figure 14. More visual results during TTS process on Pyramid-Flow. The TTS method can effectively alleviate common issues in video generation, such as those related to human motion and complex movements.

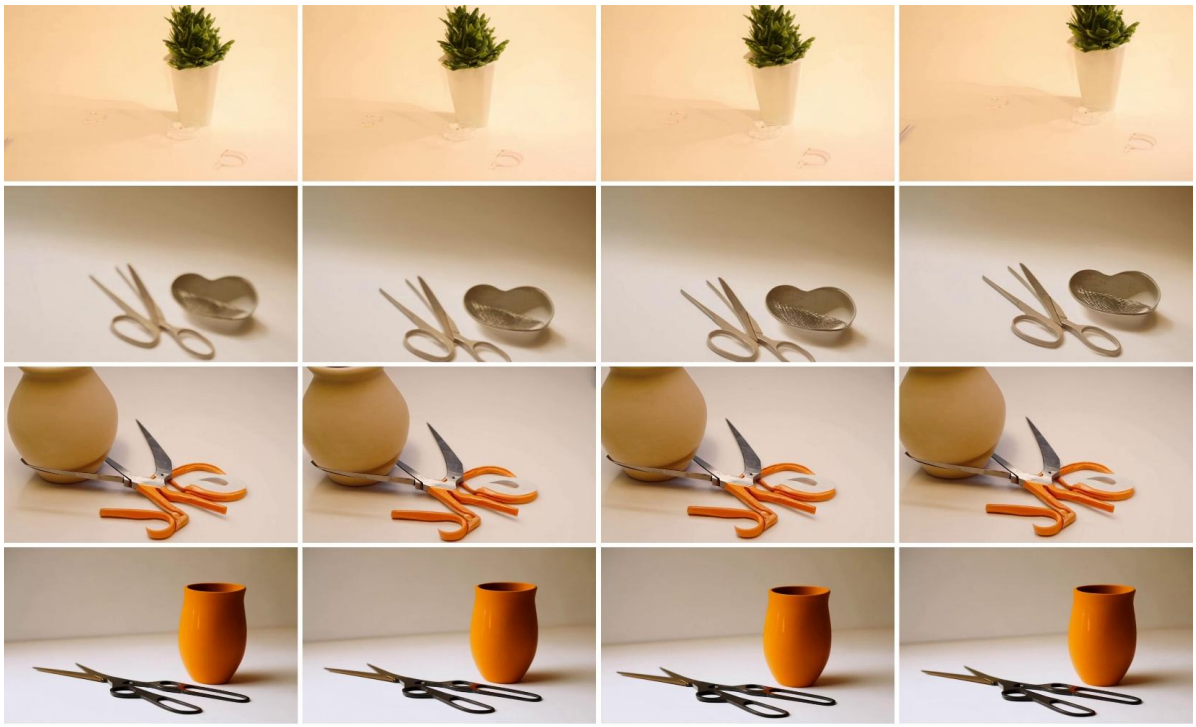

Figure 15. More visual results during TTS process on CogVideoX-5B.

Figure 16. More visual results during TTS process on CogVideoX-5B.

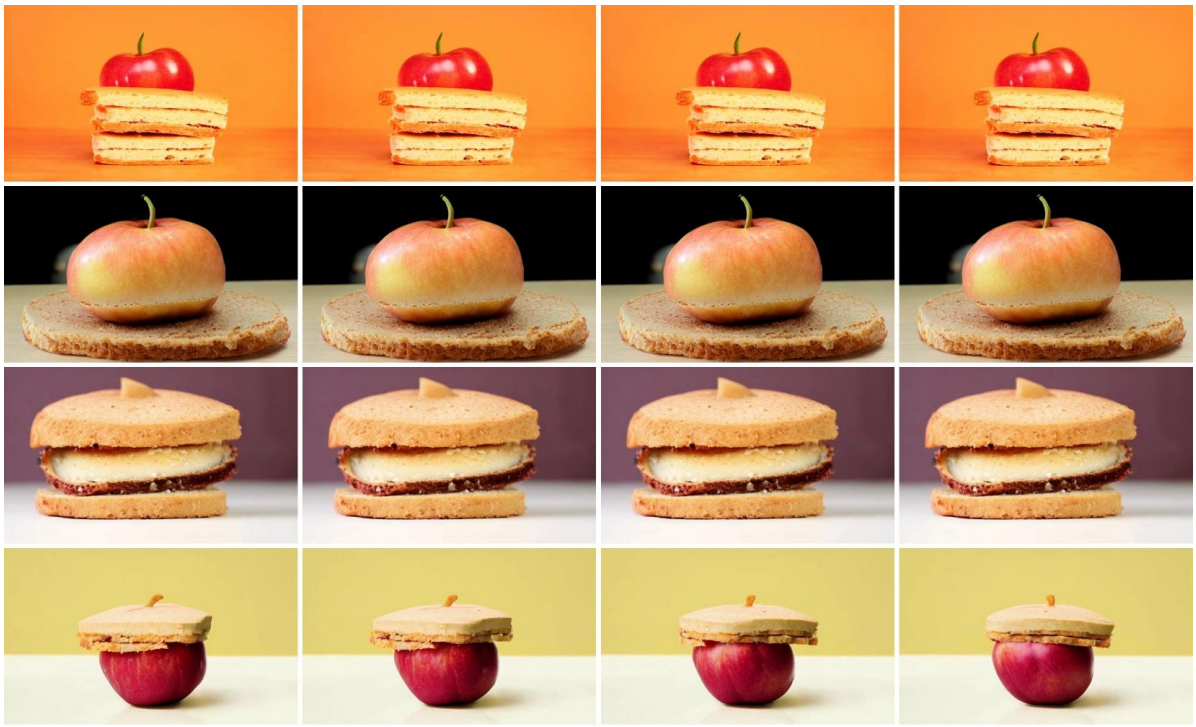

Figure 17. More visual results during TTS process on CogVideoX-2B. The TTS method can help to enhance multi-object spatial perception.

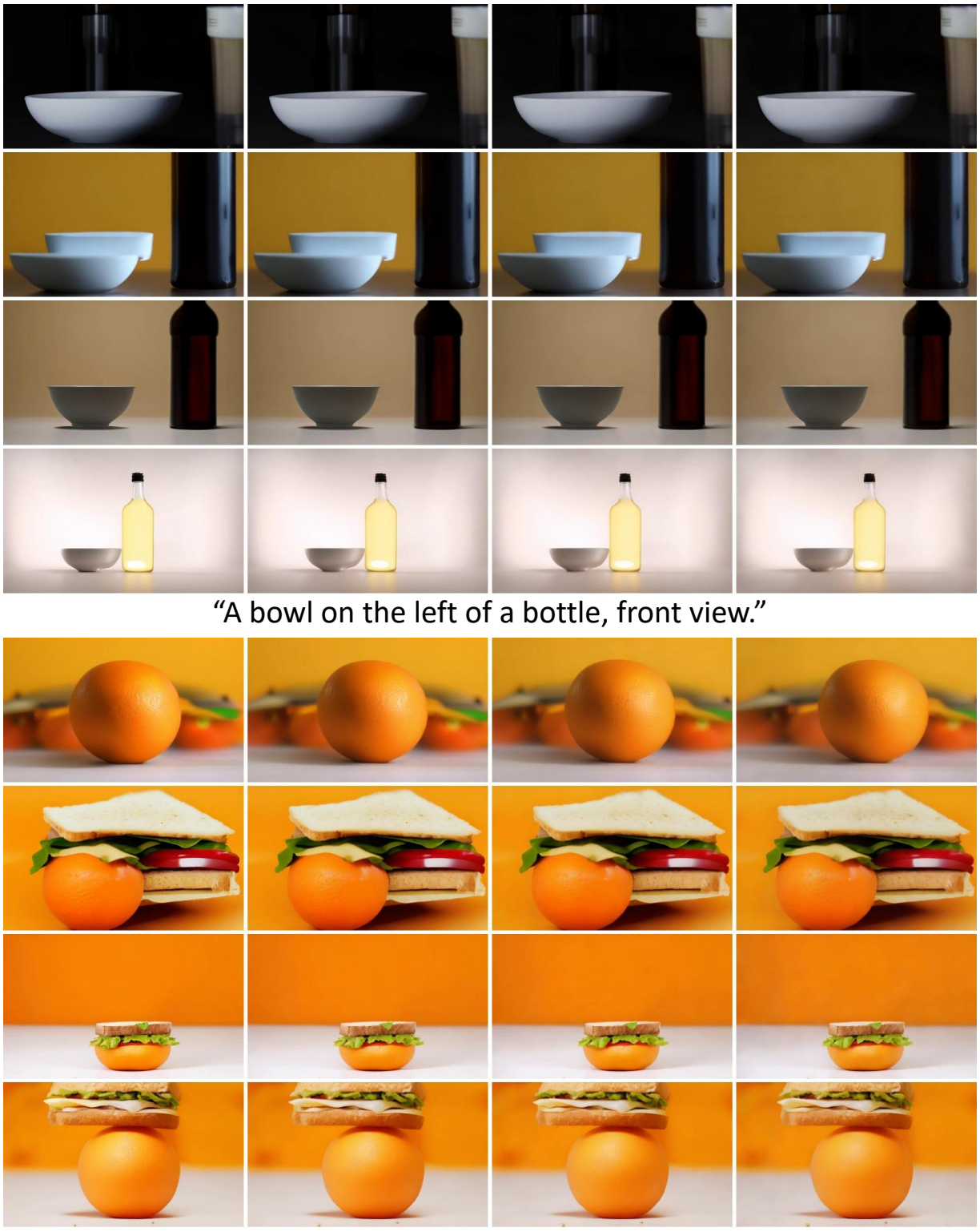

Figure 18. More visual results during TTS process on NOVA. The TTS method improves the alignment between the spatial relationships of objects in videos and the corresponding text prompt.