

## 319 Appendix

### 320 Training-Free Efficient Video Generation via Dynamic Token Carving

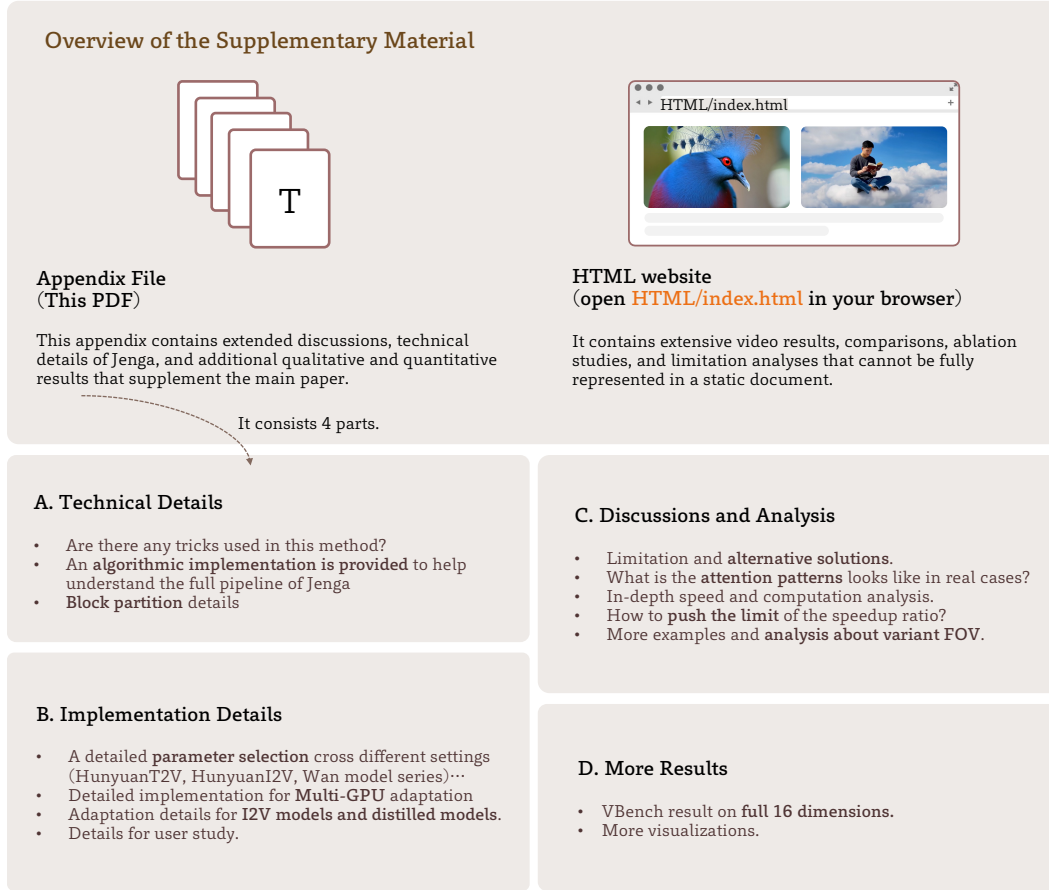


Figure 8: **Overview of the Supplementary.** We hope all readers enjoy this work in detail. We summarize common possible questions and important technical points here to arrange the supplementary. We strongly recommend that all readers open the <HTML/index.html> file in your browser for video result visualizations.

## 321 A Algorithmic Implementation

322 For a more comprehensive understanding of the method component of Jenga, we provide pseudo-code  
 323 algorithmic workflows in Algorithm 1 (Progressive Resolution), Algorithm 2 (Attention Carving  
 324 pipeline), and Algorithm 3 (building block mask B).

### 325 A.1 Details in Pipeline and ProRes

326 In the Progressive Resolution algorithm, we highlight three key technical details that were not fully  
 327 elaborated in the main text.

- 328 • *Frequency re-ordering.* Prior to each attention layer, input latent patches undergo positional  
 329 embedding operations such as RoPE [62], which typically establish frequency maps based on the  
 330 standard `thw` ordering. Since we employ  $\mathcal{G}$  to re-order the latents, we similarly apply  $f_{\text{blk}} = \mathcal{G}(f)$   
 331 to re-order the frequency components  $f$  across different dimensions, ensuring alignment with the  
 332 latent ordering. As this operation is performed only once per stage, its computational overhead is  
 333 negligible.
- 334 • *Ordering back before unpatchify.* Since the block selection in AttenCarve occurs after patchi-  
 335 fication, and both patchify and unpatchify operations need to be performed in the `thw` space,

---

**Algorithm 1** Progressive Resolution Framework for Jenga Video Generation

---

**Require:** Text prompt  $c$ , stage number  $S$ , resolutions  $R_1, \dots, R_S$ , block size  $m$ , block selection rates  $k_1, \dots, k_S$ , cutoff probability  $p$ , text-amplifier  $\rho$ , timestep lists  $T_1, \dots, T_S$   
**Ensure:** Diffusion model  $M_\theta$ , flow-matching scheduler

- 1: Text tokens:  $x_c = \text{LM}(c)$
- 2: **for**  $s = 1$  to  $S$  **do**
- 3:   Initial noise  $\tilde{\epsilon} \sim \mathcal{N}(0, I) \in \mathbb{R}^{R_s}$ ,  $x_T \leftarrow \tilde{\epsilon}$  **if**  $s = 1$
- 4:   Compute block reordering  $\mathcal{G}, \mathcal{G}^{-1}$  and adjacency masks  $\mathbf{B}_{\text{adja}}$
- 5:   Remap positional frequencies:  $f_{\text{blk}} \leftarrow \text{getFreq}(R_s, \mathcal{G})$
- 6:   **for**  $t$  in  $T_s$  **do**
- 7:     Reorder tokens:  $z_t \leftarrow \mathcal{G}(\text{patchfiy}(x_t))$
- 8:     Apply sparse attention:  $z_t \leftarrow M_\theta(z_t, x_c, k_s, \rho, f_{\text{blk}}, \mathbf{B}_{\text{adja}})$
- 9:     Restore order:  $\epsilon_t \leftarrow \text{unpatchfiy}(\mathcal{G}^{-1}(z_t))$
- 10:    Denoise step:  $x_{t-1} \leftarrow \text{scheduler}(x_t, \tilde{\epsilon}_t, t)$
- 11:   **end for**
- 12:   **if**  $s > 1$  **then**
- 13:     Predict clean latent:  $\hat{x}_0^s \leftarrow x_t - \sigma_t \epsilon_t$
- 14:     Resolution transition:  $x_{t-1} \leftarrow (1 - \sigma_t) \times \mathcal{U}(\hat{x}_0^s) + \sigma_t \tilde{\epsilon}$
- 15:     Reset text amplifier:  $\rho \leftarrow 0$  **for**  $s > 1$
- 16:     Increase sampling shift:  $\alpha \leftarrow \alpha + 2$
- 17:   **end if**
- 18: **end for**
- 19: **return** Final prediction  $x_0$

---

---

**Algorithm 2** Block-Sparse Attention with Conditional Enhancement

---

**Require:** Query  $Q$ , Key  $K$ , Value  $V$ , top- $k$ , block size  $m$ , text blocks  $M_c$ , probability threshold  $p$ , adjacency mask  $\mathbf{B}_{\text{adja}}$   
**Ensure:** Attention output

- 1: Get visual blocks  $M_v \leftarrow \lfloor N/m \rfloor - M_c$
- 2: **if**  $M_v > 0$  **then**
- 3:   Extract  $Q_v$  from first vision blocks  $\times M$  tokens
- 4:    $\mathbf{B} \leftarrow \text{BuildMask}(Q_v, K, k, p, M_c \cup \mathbf{B}_{\text{adja}})$
- 5:    $O_v \leftarrow \text{AttenCarve}(Q_{\text{normal}}, K, V, \mathbf{B})$
- 6: **end if**
- 7: **if**  $M_c > 0$  **then**
- 8:   Extract  $Q_c$  from remaining tokens
- 9:    $O_c \leftarrow \text{FullAttention}(Q_c, K, V)$ : Text blocks see all.
- 10: **end if**
- 11: **return**  $\text{concat}(O_v, O_c)$

---

---

**Algorithm 3** Build Block-wise Attention Mask

---

**Require:** Query  $Q_v$ , Key  $K$ , top- $k$ , probability threshold  $p$ , visual blocks  $M_v$ , adjacency mask  $\mathbf{B}_{\text{adja}}$   
**Ensure:** Block selection mask  $\mathbf{B}$

- 1:  $\hat{Q}, \hat{K} \leftarrow \text{BlockPool}(Q_v), \text{BlockPool}(K)$ , mean pooling per block.
- 2: Block attention scores:  $\mathbf{S} \leftarrow \hat{Q} \hat{K}^\top / \sqrt{d_k}$
- 3: Convert to probabilities:  $\mathbf{R} \leftarrow \text{softmax}(\mathbf{S})$
- 4: Sort probabilities:  $\mathbf{R}_{\text{sorted}}, \mathbf{I} \leftarrow \text{sort}(\mathbf{R}, \text{desc} = \text{True})$
- 5:  $\mathbf{C} \leftarrow \text{cumsum}(\mathbf{R}_{\text{sorted}})$
- 6:  $N_k \leftarrow \max(\text{sum}(\mathbf{C} \leq p) + 1, k \cdot M_v)$
- 7: Initialize:  $\mathbf{B}_{\text{top}} \leftarrow \text{zeros}(B, H, M_v, M_{\text{total}})$
- 8: Fill  $\mathbf{B}_{\text{top}}$  using indices  $\mathbf{I}[:, :, :, 0 : N_k]$
- 9:  $\mathbf{B}_{\text{cond}} \leftarrow \{i > M_v \vee j > M_v\}$
- 10:  $\mathbf{B} \leftarrow \mathbf{B}_{\text{top}} \cup \mathbf{B}_{\text{adja}} \cup \mathbf{B}_{\text{cond}}$
- 11: **return**  $\mathbf{B}$

---

336 we must execute reordering after patchification. Subsequently, before unpatchification, we apply  
337 the inverse operation  $\mathcal{G}^{-1}$  from Eq. (2), ensuring that all transformations are performed in the  
338 appropriate space.

339 • *Scheduler re-shift.* Following the re-noise process in Eq. (4), although theoretically we maintain  
340 the same noise strength, the clean state  $\hat{x}_0^s$  still exhibits a discrepancy from the true distribution.  
341 To address this, we employ an approach similar to BottleNeck Sampling [47, 63], progressively  
342 increasing the timestep shift factor  $\alpha$  of the rectified flow scheduler across stages.

## 343 A.2 Details in AttenCarve

344 The implementation of AttenCarve builds upon the official codebase of block-wise MInference [34].  
345 To enhance attention efficiency, we decoupled the vision and text query blocks as  
346  $Q = \text{concat}(Q_v, Q_c)$ , and applied FlashAttention2 [15] directly to the condition blocks. For the cut-  
347 off probability constraint when constructing the importance mask  $\mathbf{B}_{\text{top}}$ , we formulate the optimization

---

**Algorithm 4** Block-Sparse Attention Kernel with Text Amplification
 

---

**Require:** Query  $Q$ , Key  $K$ , Value  $V$ , sequence lengths, qk scale, text amplifier  $\rho$ , text block start index, block mask  $\mathbf{B}$ , block dimensions

**Ensure:** Output features

```

1: start_m  $\leftarrow$  program_id(0) // Current query block
2: off_hz  $\leftarrow$  program_id(1) // Batch * head index
3: Load sequence length and check bounds
4: Initialize offsets for data loading
5: Load query block  $q$  and scale by qk_scale
6: Initialize accumulators  $m_i \leftarrow -\infty, l_i \leftarrow 0, \text{acc} \leftarrow 0$ 
7: for block_idx = 0 to NUM_BLOCKS - 1 do
8:   is_valid_block  $\leftarrow \mathbf{B}[\text{off\_hz}, \text{start\_m}, \text{block\_idx}]$ 
9:   if is_valid_block then
10:    Load key-value block  $k, v$  at offset block_idx  $\times$  BLOCK_N
11:    Compute attention scores  $\text{qk} \leftarrow q \cdot k^T$ 
12:    Apply sequence length mask to qk
13:    // Apply text amplification
14:    is_text_block  $\leftarrow$  block_idx  $\geq$  text_block_start
15:    qk  $\leftarrow$  qk +  $\rho$  if is_text_block else qk
16:    Compute attention weights  $p \leftarrow \exp(\text{qk} - \max(\text{qk}))$ 
17:    Update accumulators with standard attention updates
18:   end if
19: end for
20: Normalize:  $\text{acc} \leftarrow \text{acc}/l_i$ 
21: Write results to output
  
```

---

348 problem as minimizing the number of selected blocks:

$$\min_{\mathbf{B}_{\text{top}}[i]} |\mathbf{B}_{\text{top}}[i]| \quad \text{subject to} \quad \sum_{j \in \mathbf{B}_{\text{top}}[i]} \mathbf{R}[i][j] > p \quad (5)$$

349 To satisfy this constraint, our implementation employs a sort-then-greedily-select approach. For block  
 350 index selection operations, we leverage vectorized indexing techniques to circumvent large-scale for  
 351 loops, thereby substantially improving computational efficiency. In line 2 of Algorithm 3, we address  
 352 an omission in the original Eq. (3) by explicitly incorporating the dimension  $d_k$  in multi-head attention.  
 353 Additionally, we implemented several engineering optimizations based on the MInference [34] block  
 354 selection mechanism, including replacing the original einsum operations with CUBLAS-optimized  
 355 torch.bmm() functions for enhanced latency performance.

### 356 A.3 Index Re-Order and Block Partition

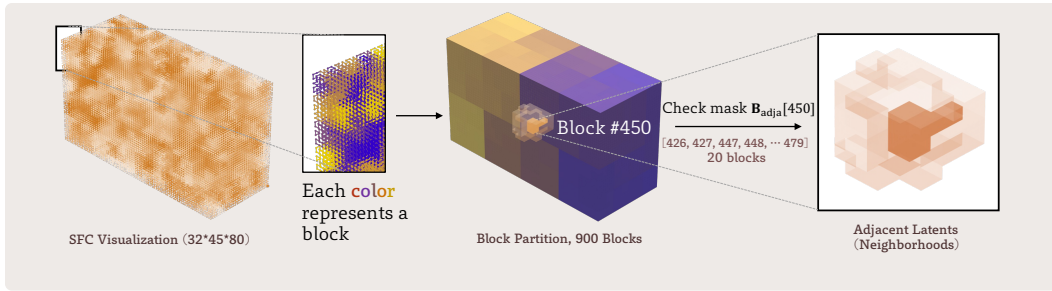


Figure 9: **A real block partition example.** We adopt a resolution-independent Space-Filling Curve (SFC) [52] to accommodate a wider range of resolutions compared to static 3D partitions. The right portion illustrates the local adjacent blocks using a look-up mask  $\mathbf{B}_{\text{adja}}$ .

357 To provide readers with a better understanding of the block partition characteristics in Jenga, beyond  
 358 the toy example in Fig. 3, we demonstrate the Space-Filling Curve (SFC) implementation in a real  
 359 720P video latent space in Fig. 9. We employ Generalized Hilbert curves to overcome the limitation  
 360 of standard Hilbert curves, which are only suitable for  $(2^n, 2^n, 2^n)$  3D spaces. It is important to

Table 4: **Detailed parameters.** We report the error bars for DiT latency measurements. The **bolded** steps indicate the additional steps required during stage transitions.

Settings	AttenCarve				ProRes				Performance	
	NFE	$k$ list	$p$	$S$	$R^S$	step ratio	$\rho$	$\alpha$	latency	VBench
HunyuanVideo [12]	50				$R^S = [32, 45, 80]$				$1625 \pm 15s$	82.74%
Jenga-Base	23	[0.3, 0.2]	0.3	1	$R^S \times [1.0, 1.0]$	[0-24, 25-49]	0.5	[7]	$347 \pm 6s$	83.34%
Jenga-Turbo	24	[0.3, 0.2]	0.3	2	$R^S \times [0.75, 1.0]$	[0-24, <b>25-49]</b>	0.5	[7, 9]	$225 \pm 5s$	83.07%
Jenga-Flash	24	[0.3, 0.2]	0.3	2	$R^S \times [0.75, 1.0]$	[0-24, <b>25-49]</b>	0.5	[7, 9]	$184 \pm 3s$	82.73%
Jenga-3Stage	24	[0.3, 0.2, 0.2]	0.3	3	$R^S \times [0.5, 0.75, 1.0]$	[0-14, 15-24, <b>25-49]</b>	0.5	[7, 9, 11]	$157 \pm 3s$	80.53%
HunyuanVideo-I2V [12]	50				$R^S = [32, 45, 80]$				$1499 \pm 12s$	87.49%
+ Jenga	23	[0.3, 0.2]	0.3	1	$R^S \times [1.0, 1.0]$	[0-24, 25-49]	0.0	[7]	$338 \pm 4s$	87.75%
AccVideo [23]	5				$R^S = [32, 44, 78]$				$161 \pm 4s$	83.84%
+ Jenga	5	[0.3, 0.2]	0.3	1	$R^S \times [1.0, 1.0]$	[0-24, 25-49]	0.5	[7]	$76 \pm 2s$	83.39%
Wan2.1-1.3B [13]	50				$R^S = [20, 30, 52]$				$115 \pm 3s$	83.28%
+ Jenga	15	[0.2, 0.1]	0.9	1	$R^S \times [1.0, 1.0]$	[0-24, 25-49]	0.0	[7]	$24 \pm 2s$	82.68%

note that each block in Jenga is not a regular rectangular prism, but rather a local cluster of tokens that are naturally partitioned. This design provides Jenga with minimal constraints regarding video dimensions—without requiring padding along physical dimensions, it only necessitates that the total token count  $\text{thw}$  be divisible by the block count  $m$ . The continuity property of SFC in the original space also ensures a certain degree of semantic similarity among tokens within each block.

We further demonstrate how to utilize the Adjacency Mask  $\mathbf{B}_{\text{adja}}$  to identify blocks that are spatially adjacent in 3D space based on their SFC representation. As illustrated, for block 450, by identifying the blocks to which neighboring tokens belong, we located 20 adjacent blocks that are subsequently incorporated into the attention computation for the current block.

## B Implementation Details

### B.1 Detailed Parameter Settings

In Tab. 4, we provide a comprehensive list of almost all key parameters used in this work. It is worth noting that although Jenga-Base employs a single-stage pipeline, we utilized different block selection rates (0.3, 0.2) at different timesteps, effectively dividing our steps into two segments. We discovered that using a higher cutoff probability (i.e.,  $p = 0.9$ ) in Wan2.1 [13] significantly improved results without incurring additional computational time, suggesting the presence of a few attention heads that concentrate on global features. We briefly describe our ProRes adaptation specifically for HunyuanVideo [12] (i.e., Jenga-Base). We will implement ProRes adaptation for Wan2.1 [13] in the future.

### B.2 Multi-GPU Adaptation

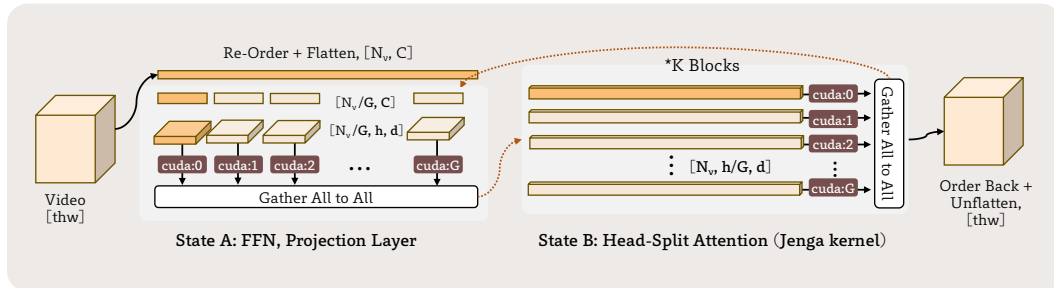


Figure 10: **Multi-GPU adaptation in Jenga.** We highlight the computation for each GPU in yellow.

For multi-GPU parallelism, we adapted our approach based on the xDiT [57] foundation used in HunyuanVideo. As illustrated in Fig. 10, we implemented parallelization across  $G$  GPUs. The parallelism within Transformer blocks remains consistent with the original implementation (i.e., state A:



parallelization along the token dimension before and after attention, and **state B**: parallelization along the head dimension within attention). We modified the corresponding LongContextAttention interface to make AttenCarve compatible with this parallel paradigm. Additionally, we discovered that when utilizing multi-GPU parallelism, the block selection process becomes the performance bottleneck. As explained in Appendix A.2, employing more efficient `torch.bmm` operations significantly accelerates multi-GPU execution (reducing processing time from 77s to 34s with 8 GPUs).

For parallelism outside transformer blocks, since we have naturally serialized tokens using SFC, we can directly partition them according to their SFC indices before feeding them into state A. This straightforward implementation also eliminates the previous requirement that latent sizes be divisible by  $G$  along specific dimensions.

### B.3 Image-to-Video & Distilled Model

For Image-to-Video [12] adaptation, two specific details warrant clarification. Since this model performs specialized modulation operations on image conditions (latent at  $t[0]$ ), we provide an additional token-level mask  $\mathcal{G}(\mathbf{m})$ ,  $\mathbf{m} = \{1 \text{ if } t = 0, \text{ else } 0\}$  when inputting tokens into the model. This enables decoupled modulation operations on the re-ordered latents. Additionally, the condition mask  $\mathbf{B}_{\text{cond}}$  incorporates both text conditions and conditioning features from the first frame. Given that the first frame already contains the overall content of the video, we did not implement the text-attention amplifier.

For the distilled model AccVideo [23], which inherently requires fewer sampling steps, we employed a single-stage Jenga-Base setting as detailed in Tab. 4. Other configurations, including multi-GPU implementation, remain consistent with our HunyuanVideo setup.

### B.4 Compared Baselines

To establish a uniform evaluation standard, we standardized the test prompts, utilized the more widely adopted FlashAttention2 [15], and maintained consistent input video dimensions across experiments. Below are the specific configurations for comparison methods beyond the baseline:

- *CLEAR* [19]. We implemented based on the original FlexAttention [64] with a 3D radius  $r = 32$ . When calculating FLOPs, since CLEAR does not account for GPU parallelism capabilities, we used the actual block sparsity (11.1% instead of the theoretical 56%) to compute effective FLOPs. Combined with the kernel optimization overhead of FlexAttention itself, the resulting generation speed could not even surpass the baseline.
- *MIInference* [34]. As explained in Sec. 4.2, we enhanced the block-wise attention mechanism from MIInference. We removed the causal mask designed for LLMs and implemented a selection rate of  $k = 0.3$ . Notably, several approaches similar to MIInference exist, such as block-sparse attention [65] and MoBA [37], which employ essentially identical methodologies.
- *SVG* [20]. We utilized SVG’s original implementation and resolution, incorporating its optimized RoPE and Normalization kernels with a sparsity setting of 0.2.
- *TeaCache* [29]. We employed the official thresholds (0.1 for slow, 0.15 for fast configurations). For Wan2.1, we set the threshold to 0.2 and enabled the `use_ret_step` parameter, which provided further acceleration while preserving result quality.

### B.5 Details about User Study

Fig. 11 presents the Google Form questionnaire and anonymous website interface used to display video assets in our user study. We randomly sampled 12 prompts from a pool of 63 paired results and randomized the left-right ordering of videos within each comparison pair. To ensure data quality, we excluded invalid responses with completion times less than 5 minutes or greater than 1 hour. We also removed 3 submissions exhibiting highly homogeneous selection patterns (e.g., consistently choosing the "left video" or "same" for all comparisons). The results from the remaining 70 valid questionnaires are presented in Fig. 6.

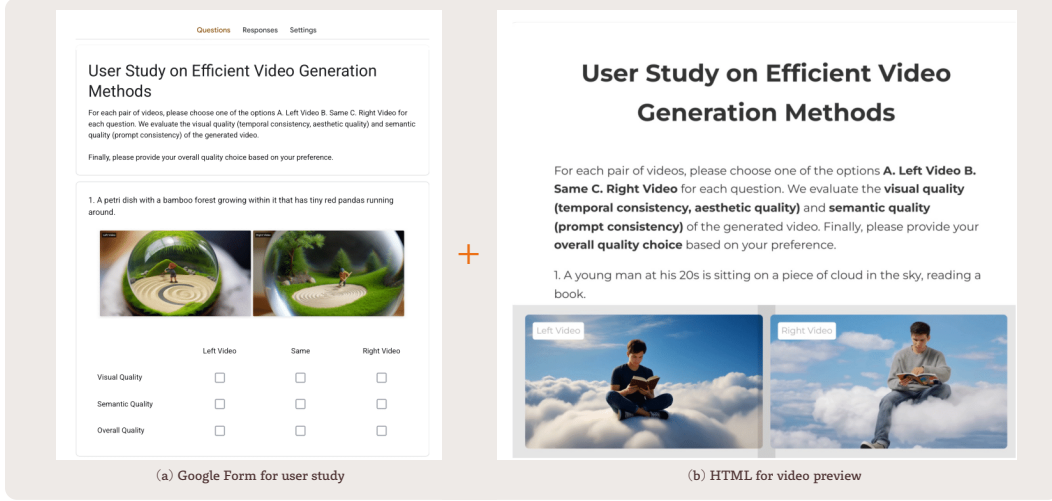


Figure 11: **User study.** (a): Questionnaire form example using Google Form. (b): Anonymous video preview website for live comparison.

## 431 C Discussions and Analysis

### 432 C.1 Limitation Analysis & Alternative Solutions

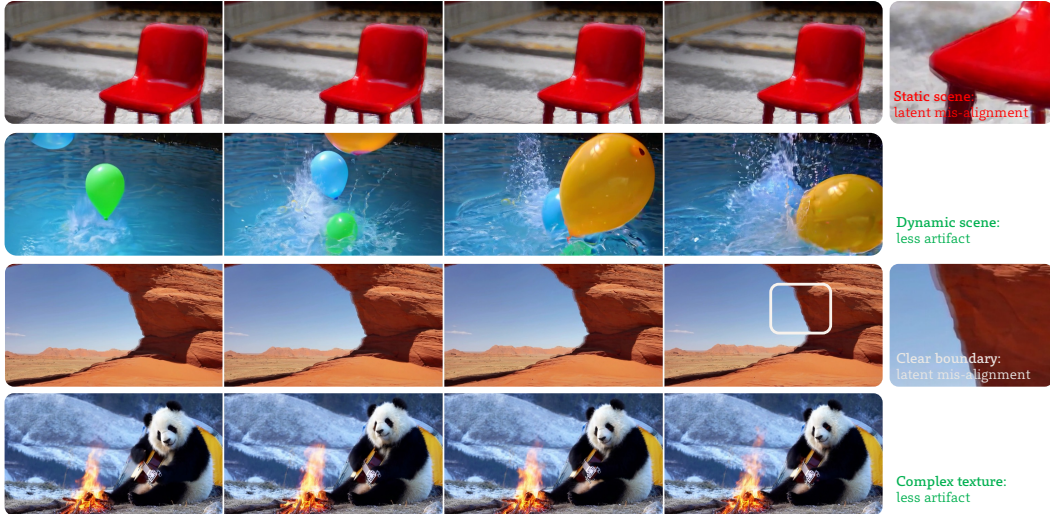


Figure 12: **Some failcases.** We present two potential failure cases that may occur when using more stages ( $S > 3$ ), as well as scenarios where this setting is more suitable.

Table 5: **Results with different prompt formats.** Generation with enhanced prompts can eliminate quality degradation and boost multi-stage results (comparable video quality with  $10.35 \times$  speedup).

	HunyuanVideo [12] 1.00×			Jenga-Turbo (2-stage) 7.22×			Jenga-3Stage (3-stage) 10.35×		
Prompt	VBench Total	VBench-Q	VBench-S	VBench Total	VBench-Q	VBench-S	VBench Total	VBench-Q	VBench-S
Standard	82.74%	85.21%	72.84%	83.07%	84.47%	77.48%	80.53% <sub>-2.21%</sub>	81.66% <sub>-3.55%</sub>	76.00% <sub>+3.16%</sub>
Enhanced	82.61%	83.98%	77.11%	83.29%	84.22%	79.57%	82.34% <sub>-0.27%</sub>	83.65% <sub>-0.33%</sub>	77.08% <sub>-0.03%</sub>

433 As discussed in Sec. 4.3, Jenga faces certain challenges when implementing Progressive Resolution  
 434 (ProRes). Several studies [66, 67] have examined the disparities between latent-space resizing and  
 435 pixel-space resizing. Even with substantial re-noising ( $\sigma_t > 0.9$ ), we cannot guarantee that edges in

the pixel space will be perfectly denoised in the final result. Since our work focuses on transformer acceleration, we opted against using the VAE decode-resize-encode approach, as tiled decode-encode operations during stage transitions would introduce additional latency of nearly 50 seconds. Fig. 12 illustrates some failure cases and usage scenarios of our current solution in 3-stage Jenga (results shown in Tab. 3b,  $10.35\times$  faster). We observed that generation quality occasionally deteriorates in static scenes or scenarios with clear boundaries (as well as in the Image-to-Video scenario). However, these issues tend to diminish when generating more complex textures or scenes with intricate motion patterns. We validated both the baseline and multi-stage results on VBench using enhanced prompts, as shown in Tab. 5. **This enables users to obtain satisfactory video results with significant acceleration when using more complex prompts** (such as Sora-style prompts, as demonstrated in Fig. 5 (b), the SUV case).

Beyond the training-based improvements discussed in Sec. 4.3, another promising direction for optimization is developing enhanced block partition methods. While the current SFC approach possesses many desirable properties, it remains fundamentally static. Extending context-based SFC approaches [68] into 3D video latent space could potentially yield better utilization of block selection.

## C.2 Block Selection: Attention Patterns

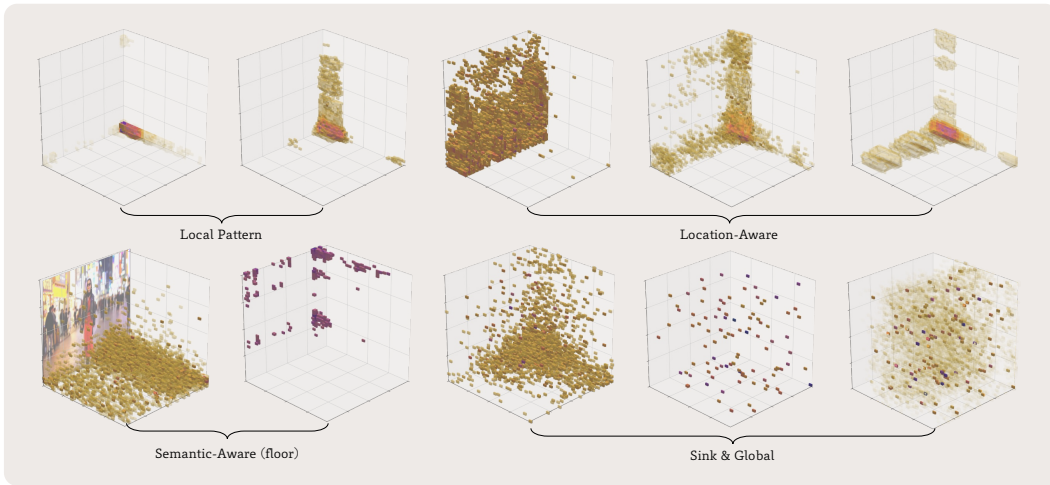


Figure 13: **Attention patterns.** Visualization of attention distributions across different layers and timesteps for the first block (at the corner position) containing 128 latent items.

We visualize the block-aware attention scores in Fig. 13. Our analysis reveals four key characteristics in the attention patterns: (1) In shallow layers, most patterns exhibit strong locality features, or (2) attention patterns highly correlate with position, forming stripe or planar distributions. In deeper layers of the model, (3) semantic-aware attention patterns emerge, where attention shifts according to the video’s semantic content. (4) Simultaneously, we observe hybrid patterns combining the three aforementioned characteristics, as well as global patterns with attention sinks. Our cut-off probability threshold is specifically designed to capture information from these latter heads. These visualized patterns not only demonstrate the inherent sparsity characteristics of attention mechanisms but also highlight the necessity for dynamic block selection in our approach.

## C.3 Resolution-Aware Field of View

In addition to the influence of the text-attention amplifier on Field of View (FOV) demonstrated in Figs. 4 and 5, we present additional examples in Fig. 14 showing dynamic FOV changes achieved by adjusting the factor  $\rho$ . We observed that in certain scenarios, not utilizing the text-attention amplifier results in an overly localized focus, ultimately reducing the content coverage in the frame. By introducing the bias parameter  $\beta$ , we can exert a degree of control over different field-of-view ranges.

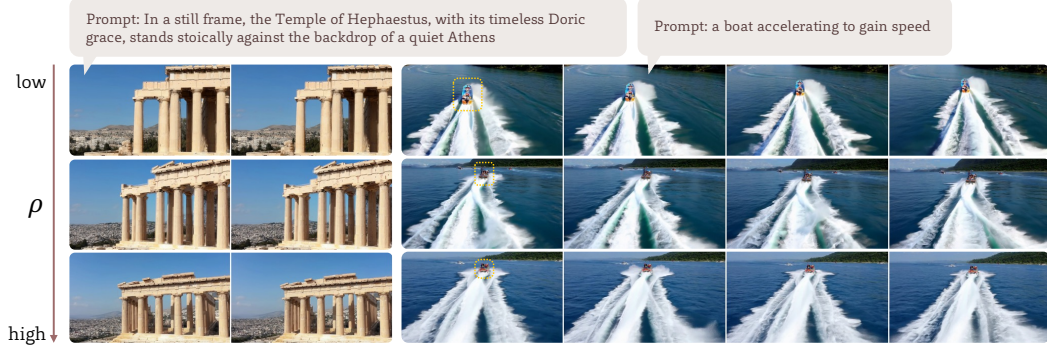


Figure 14: **Dynamic FOV.** We demonstrate the impact of the balancing factor  $\rho$  on field of view in both static and dynamic scenes. Additional ablation examples are presented in the HTML supplement.

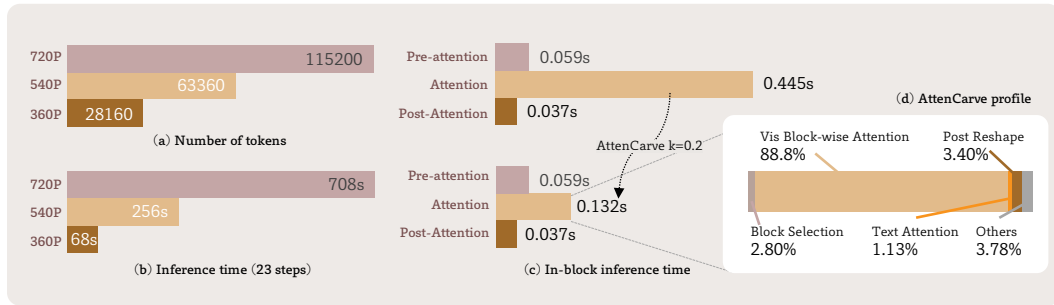


Figure 15: **Latency analysis.** (a, b) Visual token counts and generation times at different resolutions. (c) Acceleration of AttenCarve vs. FlashAttention2 [15]. (d) Time breakdown across AttenCarve components.

#### 468 C.4 Speed Analysis & Additional Overheads

469 In this section, we provide an in-depth analysis of our method’s latency. First, as illustrated in Fig. 15  
 470 (a)-(b), we demonstrate the necessity of directly reducing token count by adjusting resolution. At  
 471 360P, only 1/4 of the input tokens, the generation speed achieves a 10× improvement compared to  
 472 720P. In Fig. 15 (c), we specifically evaluate the acceleration achieved by AttenCarve compared to  
 473 FlashAttention2 [15], which achieves a 3.7× speedup in attention computation. Furthermore, Fig. 15  
 474 (d) provides a detailed time breakdown across different components of AttenCarve, showing that  
 475 Block selection introduces only 2.8% computational overhead. Additionally, we analyzed the memory  
 476 efficiency of our approach. **Without any specialized optimizations**, when generating 720P videos,  
 477 Jenga introduces a minimal additional memory overhead of only 3.7% (71.84 → 74.49 GiB).

478 Despite the series of optimizations in Jenga, numerous avenues remain for potential performance  
 479 improvements. These include incorporating quantization optimizations mentioned in SVG [20] and  
 480 SparseAttn [48], as well as kernel optimizations for RoPE [62] and normalization operations. From a  
 481 hardware perspective, adapting FlashAttention3-based [69] attention kernels on the Hopper architec-  
 482 ture shows significant speed enhancement potential. Additionally, parallelization and sparsification  
 483 strategies for the VAE component have not been fully explored. These directions represent promising  
 484 areas for future engineering optimizations and continued investigation in our work.

## 485 D Additional Results

### 486 D.1 Detailed Benchmarks

487 Tab. 6 provides comprehensive evaluation results across all 16 dimensions of VBench [33]. As shown,  
 488 Jenga achieves notable advantages in multiple semantic score dimensions while maintaining high  
 489 performance in quality metrics.



Table 6: **Detailed VBench [33] results.** We omit the percentage symbol % for better preview.

	Quality Metrics							Semantic Metrics								
	subject consistency	background consistency	temporal flickering	motion smoothness	aesthetic quality	imaging quality	dynamic degree	object class	multiple objects	human action	color	spatial relationship	scene	appearance style	temporal style	overall consistency
Methods																
HunyuanVideo [12]	96.59	98.06	99.63	99.54	61.11	72.23	60.83	82.03	68.75	94.00	93.75	78.86	38.60	20.51	23.22	26.54
CLEAR [19]	97.15	97.82	99.61	99.57	63.03	68.88	45.83	58.59	48.89	92.00	93.27	69.41	44.18	20.97	22.61	26.36
MInference [34]	94.90	97.66	99.41	99.47	61.62	69.78	65.27	75.00	83.08	88.00	93.75	77.18	42.28	20.80	23.08	27.17
SVG [20]	96.40	97.75	99.61	99.55	61.78	69.96	61.11	74.52	63.56	94.00	90.36	77.25	34.16	20.20	23.39	26.23
AttenCarve	95.94	97.85	99.30	99.18	62.47	69.09	70.83	86.71	73.02	93.00	90.67	75.45	47.17	19.50	23.43	26.36
TeaCache-slow [29]	96.70	97.89	99.30	99.49	61.54	69.18	59.72	67.24	63.41	88.00	85.19	72.09	36.11	20.05	23.11	25.80
TeaCache-fast [29]	96.68	97.79	99.32	99.50	61.42	68.59	56.94	64.08	64.71	90.00	85.99	71.22	36.26	20.12	23.12	25.77
ProRes	96.16	97.58	99.72	99.55	63.75	70.36	70.83	82.81	55.15	89.00	88.24	67.26	26.10	20.46	21.89	26.79
ProRes-timeskip	95.57	97.68	99.74	99.54	62.93	68.97	72.22	76.95	59.19	90.00	88.24	67.11	29.04	20.66	21.75	27.04
Jenga-Base	95.09	97.86	99.31	99.18	62.47	69.09	72.22	86.71	73.02	88.00	90.67	75.45	47.17	19.51	23.43	26.36
Jenga-Turbo	93.42	96.85	99.31	98.85	63.89	66.64	77.78	94.14	66.91	94.00	95.31	73.76	50.37	19.85	23.74	27.98
Jenga-Flash	92.75	97.19	99.27	98.57	62.29	66.71	85.71	73.61	63.60	90.00	99.26	71.97	56.25	20.27	24.43	28.05
AccVideo [23]	95.92	97.53	99.35	99.28	61.40	67.98	58.33	89.40	76.30	88.00	92.50	80.29	51.09	20.49	24.43	26.73
+Jenga	95.36	96.97	99.26	99.02	61.38	68.10	66.67	90.37	75.41	86.00	93.62	78.83	46.72	20.57	24.11	26.92
Wan2.1-1.3B [13]	96.46	98.40	99.52	98.72	64.08	67.36	59.72	75.00	47.64	82.00	81.87	71.49	23.11	19.82	23.68	23.59
+ TeaCache [29]	96.40	98.25	99.38	98.70	62.03	65.59	58.33	76.39	47.48	78.00	82.47	69.16	24.13	19.83	23.14	22.99
+Jenga	95.40	97.92	99.44	98.55	61.13	65.37	61.11	74.76	53.89	78.00	88.42	70.08	26.53	20.25	23.34	23.49

Methods	Quality Metrics						I2V Semantic Metrics				Total	
	subject consistency	back ground consistency	motion smoothness	aesthetic quality	imaging quality	dynamic degree	Quality Score	camera motion	subject consistency	back ground consistency	I2V Score	Total Score
HunyuanVideo-I2V [12]	95.67	96.39	99.21	61.55	70.37	21.14	78.30	51.38	98.90	99.38	96.67	87.49
+ timeskip	95.75	96.86	99.22	61.93	70.84	21.54	78.64	51.51	98.92	99.42	96.71	87.67
+Jenga	93.99	95.75	99.00	60.84	70.43	40.65	79.31	49.80	98.43	99.14	96.18	87.74

Regarding detailed results in Tab. 6, there are two key points to clarify. First, we discovered that compared to the static local patterns used in CLEAR [19], our query/head-aware dynamic patterns significantly enhance the dynamic degree of generated results (45.83%  $\rightarrow$  70.83%). Overall, Jenga introduces larger motion amplitude at the quality level, while presenting some trade-offs in subject consistency when the selection rate is small (Jenga-Flash). At the semantic level, Jenga demonstrates substantially better semantic adherence across multiple dimensions (color, object class, scene, and overall consistency).

## D.2 More Visual Results

We showcase additional results of Jenga in different settings, as illustrated in Fig. 16, and Fig. 17. We recommend viewing the video files in the provided HTML to better evaluate the effectiveness of our method.

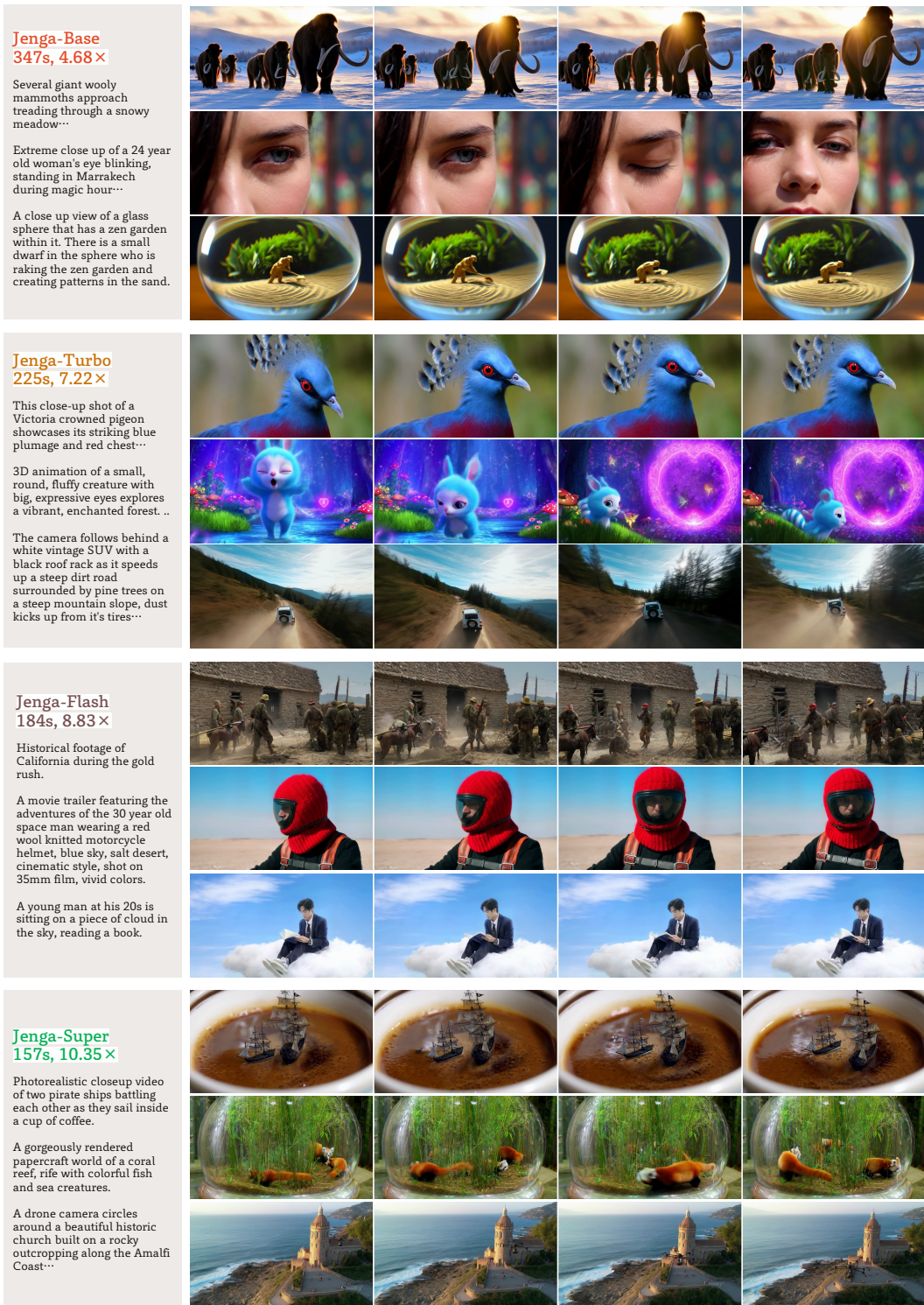


Figure 16: **Visualization results.** From top to bottom, each three videos is from the same setting.



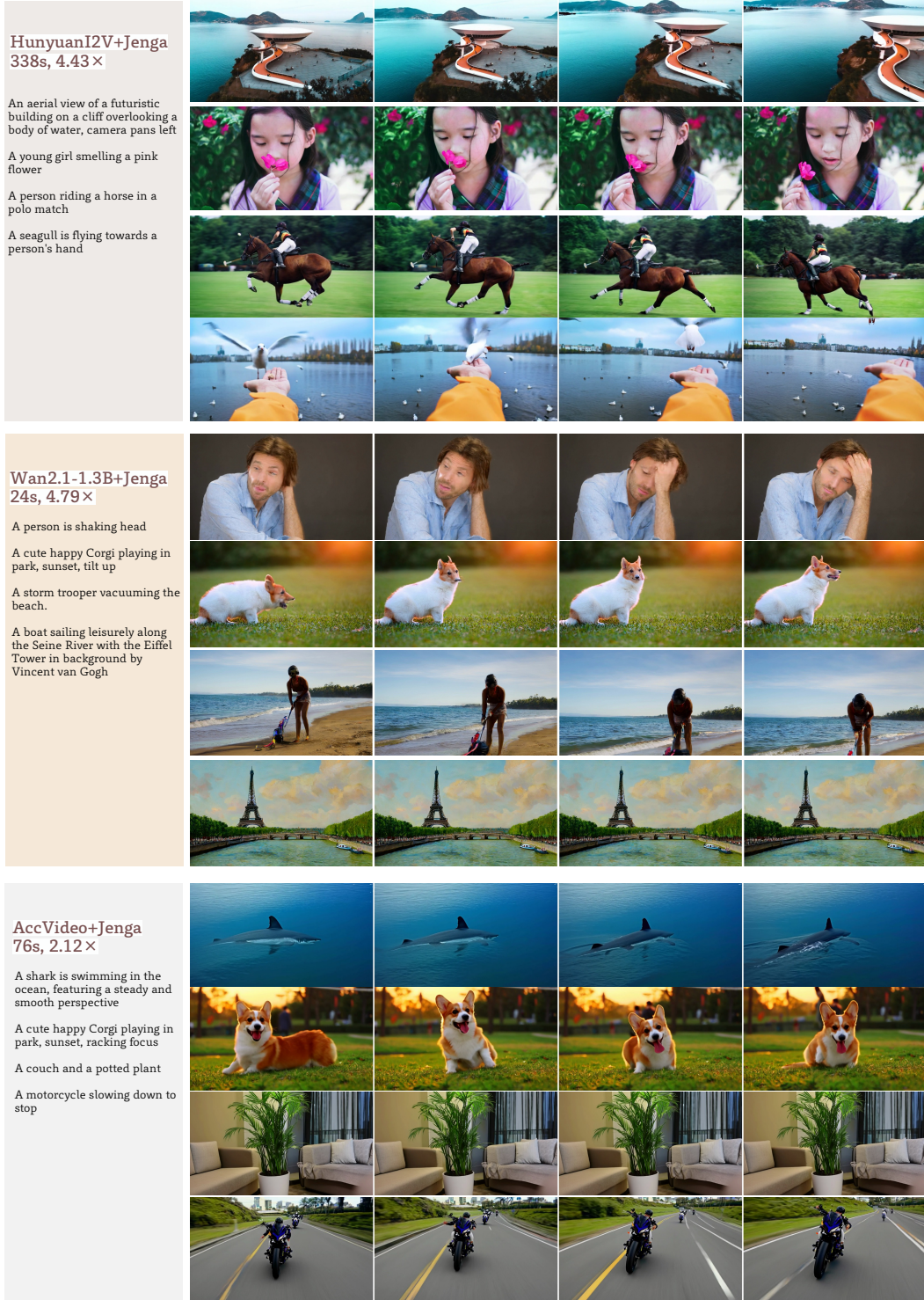


Figure 17: Visualization results for model adaptations. Prompts are from VBench [33].

## E Social Impacts

This paper introduces a novel framework for efficient video generation that is based on current pretrained Diffusion Transformers. Although this application has the potential to be misused by malicious actors for disinformation purposes, significant advancements have been achieved in detecting malicious generation. Consequently, we anticipate that our work will contribute to this domain. In forthcoming iterations of our method, we intend to introduce the NSFW (Not Safe for Work) test for detecting possible malicious generations. Through rigorous experimentation and analysis, our objective is to enhance comprehension of video generation techniques and alleviate their potential misuse.

## References

- [1] 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. 2, 3
- [2] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 2
- [3] 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. *NeurIPS*, 35:36479–36494, 2022. 2
- [4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2
- [5] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. Technical report, OpenAI, 2023. 2
- [6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 2
- [7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 2
- [8] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 2, 3
- [9] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *TMLR*, 2025. 2
- [10] Jiaqi Xu, Xinyi Zou, Kunzhe Huang, Yunkuo Chen, Bo Liu, MengLi Cheng, Xing Shi, and Jun Huang. Easyanimate: A high-performance long video generation method based on transformer architecture. *arXiv preprint arXiv:2405.18991*, 2024. 2
- [11] 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
- [12] 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, 3, 6, 7, 8, 13, 14, 15, 18
- [13] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 3, 6, 8, 13, 18
- [14] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. 2
- [15] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 2, 4, 11, 14, 17
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

- [17] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *CVPR*, pages 10124–10134, 2023. 2
- [18] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025. 2, 3, 4
- [19] Songhua Liu, Zhenxiong Tan, and Xinchao Wang. Clear: Conv-like linearization revs pre-trained diffusion transformers up. *arXiv preprint arXiv:2412.16112*, 2024. 2, 3, 7, 8, 14, 18
- [20] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025. 2, 3, 7, 8, 14, 17, 18
- [21] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *ICML*, 2023. 2
- [22] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, pages 14297–14306, 2023. 2
- [23] Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. Accvideo: Accelerating video diffusion model with synthetic dataset. *arXiv preprint arXiv:2503.19462*, 2025. 2, 3, 6, 8, 13, 14, 18
- [24] Hangliang Ding, Dacheng Li, Runlong Su, Peiyuan Zhang, Zhijie Deng, Ion Stoica, and Hao Zhang. Efficient-vdit: Efficient video diffusion transformers with attention tile, 2025. 2, 3, 6
- [25] Jintao Zhang, Haofeng Huang, Pengl Zhang, Jun Zhu, Jianfei Chen, et al. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024. 2, 3
- [26] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *ICCV*, pages 17535–17545, 2023. 2
- [27] Chenglin Yang, Celong Liu, Xueqing Deng, Dongwon Kim, Xing Mei, Xiaohui Shen, and Liang-Chieh Chen. 1.58-bit flux. *arXiv preprint arXiv:2412.18653*, 2024. 2
- [28] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *CVPR*, pages 6211–6220, 2024. 2
- [29] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It’s time to cache for video diffusion model. *arXiv preprint arXiv:2411.19108*, 2024. 2, 3, 6, 7, 8, 14, 18
- [30] Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. *arXiv preprint arXiv:2503.06923*, 2025. 2, 3, 6
- [31] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *ICLR*, 2022. 2, 5
- [32] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ICLR*, 2023. 2
- [33] 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 *CVPR*, pages 21807–21818, 2024. 2, 6, 7, 8, 17, 18, 20
- [34] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *NeurIPS*, 37:52481–52515, 2024. 3, 7, 8, 11, 12, 14, 18
- [35] Heejun Lee, Geon Park, Youngwan Lee, Jina Kim, Wonyoung Jeong, Myeongjae Jeon, and Sung Ju Hwang. Hip attention: Sparse sub-quadratic attention with hierarchical attention pruning. *arXiv e-prints*, pages arXiv–2406, 2024. 3
- [36] Heejun Lee, Geon Park, Jaduk Suh, and Sung Ju Hwang. Infinitehip: Extending language model context up to 3 million tokens on a single gpu. *arXiv preprint arXiv:2502.08910*, 2025. 3
- [37] Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025. 3, 4, 14



- [38] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *ICLR*, 2020. 3
- [39] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *ICLR*, 2023. 3
- [40] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*, 2024. 3
- [41] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, YX Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025. 3
- [42] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 3
- [43] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *ICCV*, pages 5961–5971, 2023. 3
- [44] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *CVPR*, pages 14420–14430, 2023. 3
- [45] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoni 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. 3
- [46] Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers. *arXiv preprint arXiv:2502.10389*, 2025. 3
- [47] Ye Tian, Xin Xia, Yuxi Ren, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Yunhai Tong, Ling Yang, and Bin Cui. Training-free diffusion acceleration with bottleneck sampling. *arXiv preprint arXiv:2503.18940*, 2025. 3, 5, 11
- [48] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025. 3, 4, 17
- [49] Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation. *arXiv preprint arXiv:2502.21079*, 2025. 3
- [50] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ICLR*, 2023. 3, 5
- [51] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 35:16344–16359, 2022. 4
- [52] Jakub Červený and contributors. Gilbert: Space-filling curve for rectangular domains of arbitrary size. <https://github.com/jakubcerveny/gilbert>, 2025. Accessed: April 16, 2025. 4, 6, 12
- [53] Hans Sagan. *Space-filling curves*. Springer Science & Business Media, 2012. 4
- [54] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *CVPR*, pages 4840–4851, 2024. 4
- [55] Shilong Zhang, Wenbo Li, Shoufa Chen, Chongjian Ge, Peize Sun, Yida Zhang, Yi Jiang, Zehuan Yuan, Binyue Peng, and Ping Luo. Flashvideo: Flowing fidelity to detail for efficient high-resolution video generation. *arXiv preprint arXiv:2502.05179*, 2025. 5
- [56] Philippe Tillet. Introducing triton: Open-source gpu programming for neural networks. <https://openai.com/index/triton/>, 2021. 6
- [57] Jiarui Fang, Jinzhe Pan, Xibo Sun, Aoyu Li, and Jiannan Wang. xdit: an inference engine for diffusion transformers (dits) with massive parallelism. *arXiv preprint arXiv:2411.01738*, 2024. 6, 13
- [58] Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023. 6
- [59] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *EMNLP*, 2021. 6, 8

- 646 [60] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit,  
647 Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video  
648 generative models. *arXiv preprint arXiv:2411.13503*, 2024. 6, 8
- 649 [61] OpenAI. Video generation models as world simulators. 8
- 650 [62] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced  
651 transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 10, 17
- 652 [63] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi,  
653 Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution  
654 image synthesis. In *ICML*, 2024. 11
- 655 [64] Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming  
656 model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024. 14
- 657 [65] Junxian Guo, Haotian Tang, Shang Yang, Zhekai Zhang, Zhijian Liu, and Song Han. Block Sparse  
658 Attention. <https://github.com/mit-han-lab/Block-Sparse-Attention>, 2024. 14
- 659 [66] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion:  
660 Democratising high-resolution image generation with no \$. In *CVPR*, pages 6159–6168, 2024. 15
- 661 [67] Zhen Yang, Guibao Shen, Liang Hou, Mushui Liu, Luozhou Wang, Xin Tao, Pengfei Wan, Di Zhang, and  
662 Ying-Cong Chen. Rectifiedhr: Enable efficient high-resolution image generation via energy rectification.  
663 *arXiv preprint arXiv:2503.02537*, 2025. 15
- 664 [68] Revital Dafner, Daniel Cohen-Or, and Yossi Matias. Context-based space filling curves. In *Computer  
665 Graphics Forum*, volume 19, pages 209–218. Wiley Online Library, 2000. 16
- 666 [69] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3:  
667 Fast and accurate attention with asynchrony and low-precision. *NeurIPS*, 37:68658–68685, 2024. 17