

## A Appendix

### A.1 Background

The success of Sora Brooks et al. [2024] has shown the strong potential of diffusion transformers for text-to-video generation, inspiring a wave of open-source models such as Open-Sora [Zheng et al., 2024], Open-Sora-Plan [Lin et al., 2024a], Latte [Ma et al., 2024], CogVideoX [Yang et al., 2024], Vchitect [Fan et al., 2025], and Mochi [Genmo, 2024].

Figure 8 shows a high-level overview of DiT-based text-to-video models. These models take as input Gaussian noise-initialized latent video frames, timesteps for scheduling, and a text prompt. They use Spatial-Temporal DiT (ST-DiT) blocks: Spatial-DiT captures intra-frame spatial structure, while Temporal-DiT captures inter-frame temporal dynamics. Cross-attention injects text conditioning into each block, and feedforward networks (FFNs) apply learned nonlinearities. Timestep embeddings guide denoising across both spatial and temporal layers.

### A.2 Workload Characterization

To analyze inference bottlenecks in text-to-video generation, we profile the Open-Sora [Zheng et al., 2024] model across various resolutions and timeframes, using a fixed 30-step RFlow scheduler using a single NVIDIA A100 (80GB) GPU with flash attention [Dao et al., 2022].

Figure 9 shows the end-to-end latency and its breakdown by operator type. As resolution increases from 480p to 720p, latency rises 2.5 $\times$ , driven largely by the quadratic complexity of attention operations. Attention modules (spatial, temporal, and cross-attention) account for 50% of inference time, FFNs for 15%, and non-linear layers—LayerNorm, scaling, and residuals—for 35%. The sizable cost of non-linear operations highlights the need to target them in optimization efforts.

To identify system bottlenecks during inference, we measured compute and memory throughput for Spatial and Temporal Attention blocks in the Open-Sora [Zheng et al., 2024] model using a single NVIDIA A100 (80GB) GPU with batch size 1 (Figure 10). For Spatial Attention, we varied resolution from 144p to 1080p with a fixed timeframe of 8 seconds. For Temporal Attention, we varied timeframes from 2 to 16 seconds at a fixed resolution of 720p.

In Spatial Attention, increasing resolution increases the number of spatial tokens, leading to higher compute demands due to the quadratic complexity of attention. This keeps the block compute-bound. In contrast, Temporal Attention sees a smaller increase in sequence length with longer timeframes, while the batch size—equal to the number of token patches in a frame—remains large. As a result, flash attention becomes suboptimal and performance becomes memory-bound. Overall, ST-DiT layers are primarily compute-bound, suggesting that reusing redundant computations can improve inference efficiency.

### A.3 Feature Variations across different Video Configurations

To analyze feature variation across video configurations, we vary one parameter at a time while keeping others fixed, and measure changes in intermediate features at specific layers. Figure 11 shows

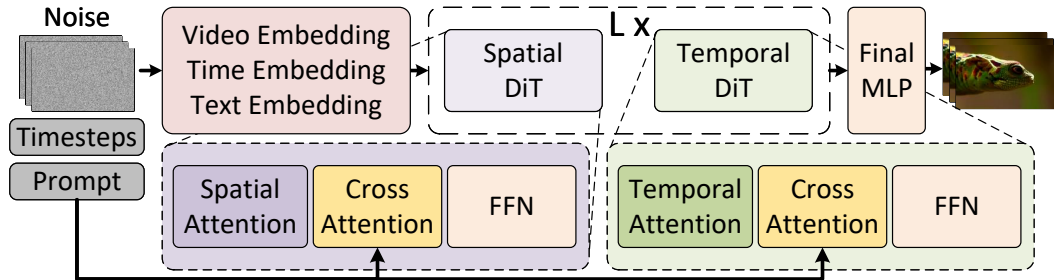


Figure 8: Overview of diffusion transformer (DiT) based text-to-video generation models. It comprises of mainly spatial and temporal diffusion transformer blocks with spatial and temporal attention. Cross attention takes into account prompt conditioning for generated video.

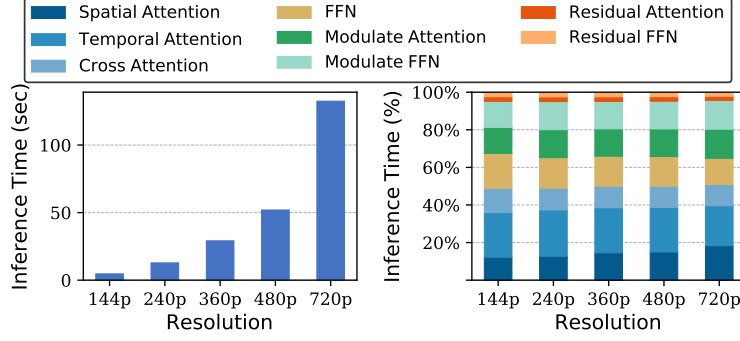


Figure 9: **Left:** End-to-end inference time across resolutions. **Right:** Inference time breakdown by operator. Results based on Open-Sora [Zheng et al., 2024] using a single NVIDIA A100 (80GB) GPU with batch size 1.

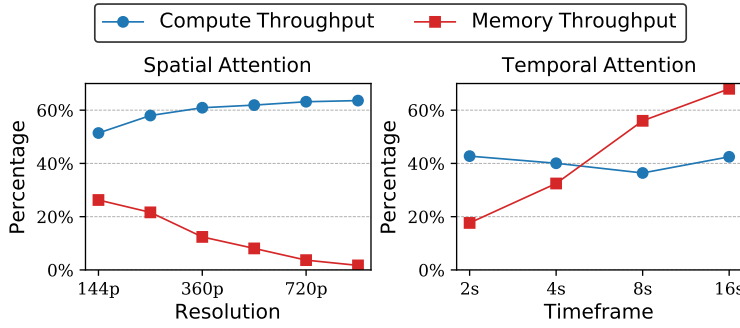


Figure 10: Compute vs. memory throughput for Spatial and Temporal Attention blocks in Open-Sora [Zheng et al., 2024], measured on a single NVIDIA A100 (80GB) GPU with batch size 1.

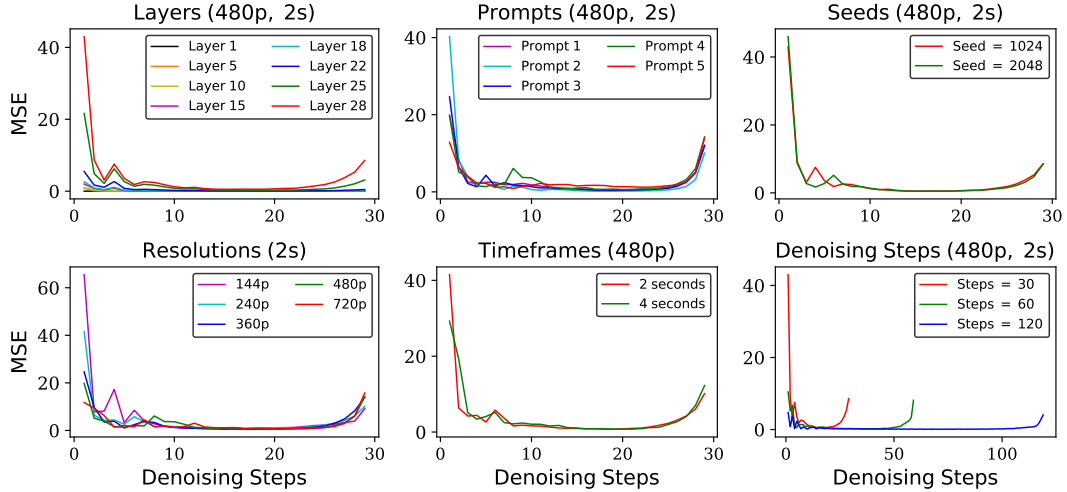


Figure 11: Quantitative analysis of Spatial DiT output using mean squared difference across different layer, prompts, seeds, video resolutions, video timeframes and denoising steps for OpenSora Zheng et al. [2024] model Layer 28 if layer not specified. The prompt is “A narrow, cobblestone alleyway, bathed in the soft glow of vintage street lamps, stretches between tall, weathered brick buildings adorned with ivy. The scene begins with a gentle drizzle, creating a reflective sheen on the cobblestones. As the camera pans, a black cat with piercing green eyes darts across the path, adding a touch of mystery. The alley is lined with quaint, shuttered windows and wooden doors, some slightly ajar, hinting at hidden stories within. A soft breeze rustles the leaves of potted plants and hanging flower baskets, while distant, muffled sounds of city life create a serene yet vibrant atmosphere.”

the effect of varying prompts, noise seeds, resolution, timeframes, and denoising steps. The results indicate that intermediate features are sensitive to these configurations. Therefore, adaptive reuse must account for such variations to minimize quality loss in video generation.

#### A.4 Reuse Metric

To analyze the dynamic behavior of reuse and intermediate feature variation, we measure how features evolve across layers and timesteps using cosine similarity.

**Across Condition** Figure 12 illustrates the evolution of spatial features during conditional generation, across layers and denoising steps.

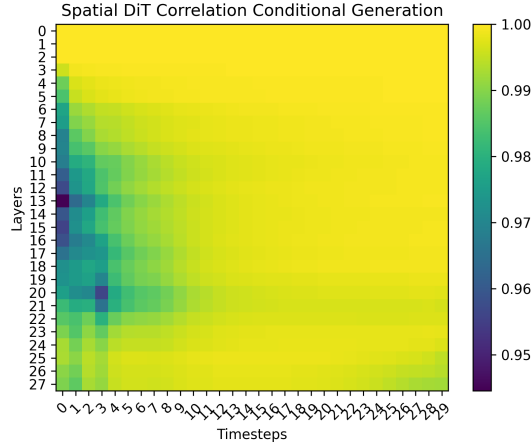


Figure 12: Cosine similarity of Spatial-DiT features across conditioning steps in the OpenSora model.

**Across Layers** Figure 13 shows the cosine similarity of spatial-DiT features across layers and denoising steps for the OpenSora model.

**Across Denoising Steps** Figure 14 shows the cosine similarity of Spatial-DiT features across denoising steps for different layers of the OpenSora model. Later layers exhibit greater feature variation than early and middle layers.

#### A.5 Evaluation Metrics

We evaluate video generation quality using established metrics that assess both perceptual quality and similarity to baseline outputs. Specifically, we use VBench [Huang et al., 2024], a comprehensive benchmark capturing multiple perceptual dimensions, and complement it with similarity metrics comparing generated videos to baseline outputs without reuse.

**VBench** [Huang et al., 2024] is a video generation benchmark that evaluates model quality across 16 well-defined dimensions. It uses 11 prompt categories, each designed to probe specific aspects of these dimensions, with weighted scores assigned to each. For evaluation, we select the first 50 prompts from each category, totaling 550 prompts.

**PSNR** measures pixel-level mean squared error between a baseline video (without reuse) and a reused-version. Higher PSNR indicates lower error and better quality.

**SSIM** [Wang and Bovik, 2002] measures image similarity by comparing structural information, luminance, and contrast between two images.

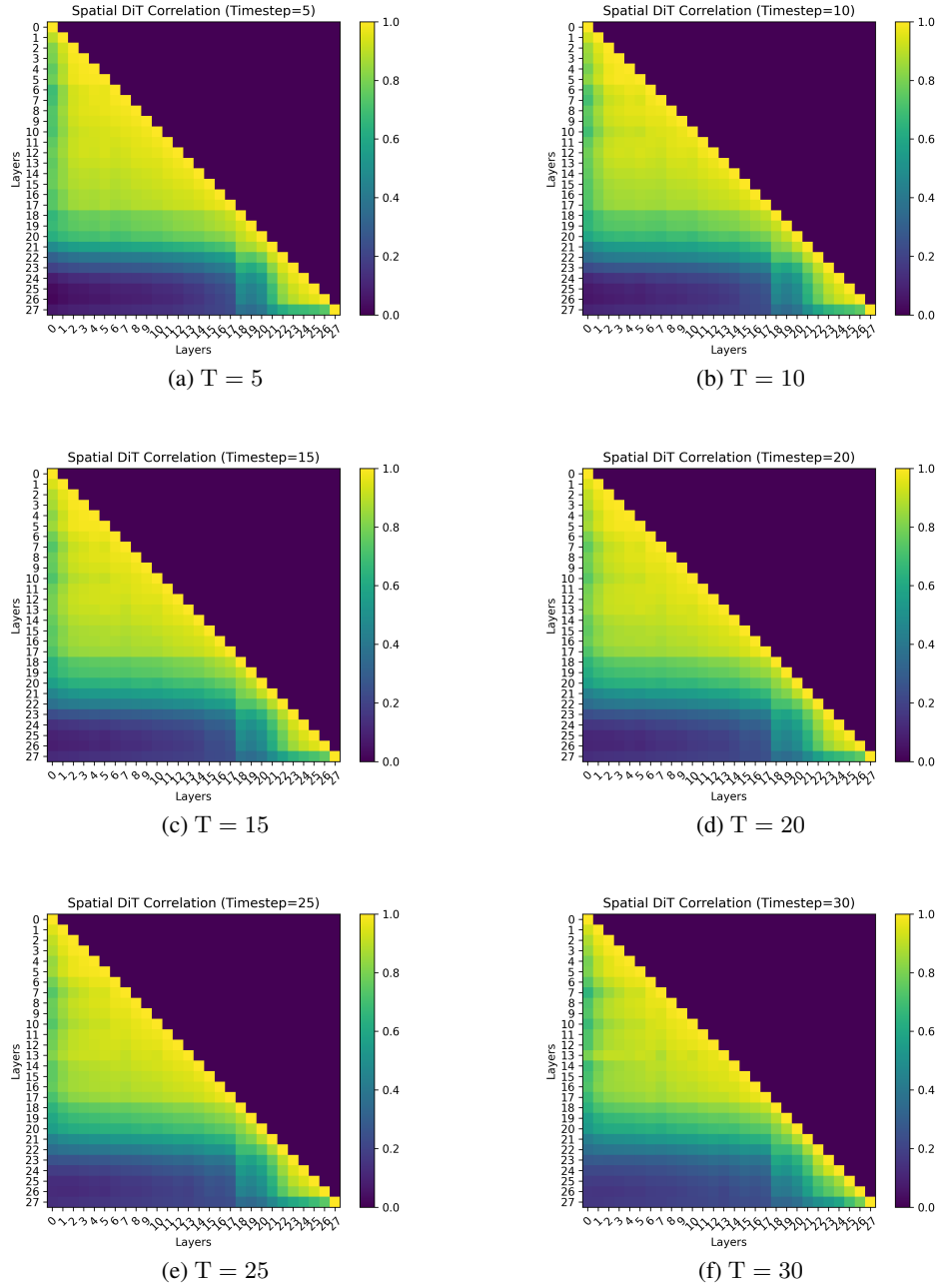


Figure 13: Cosine similarity of Spatial-DiT features across layers at different denoising steps in the OpenSora model.

834 LPIPS [Zhang et al., 2018] measures perceptual similarity between images using deep neural  
 835 network features, capturing differences more effectively than pixel-based metrics. It computes the  
 836 distance between features extracted from pre-trained models trained on large-scale datasets.

837 FVD [Unterthiner et al., 2019] based on the Fréchet Inception Distance (FID) for images, extends  
 838 the concept to videos. It measures the distance between real and generated video distributions in the  
 839 feature space of a pretrained network. By capturing both spatial quality and temporal consistency,  
 840 FVD reflects how closely generated videos match real ones in appearance and motion.

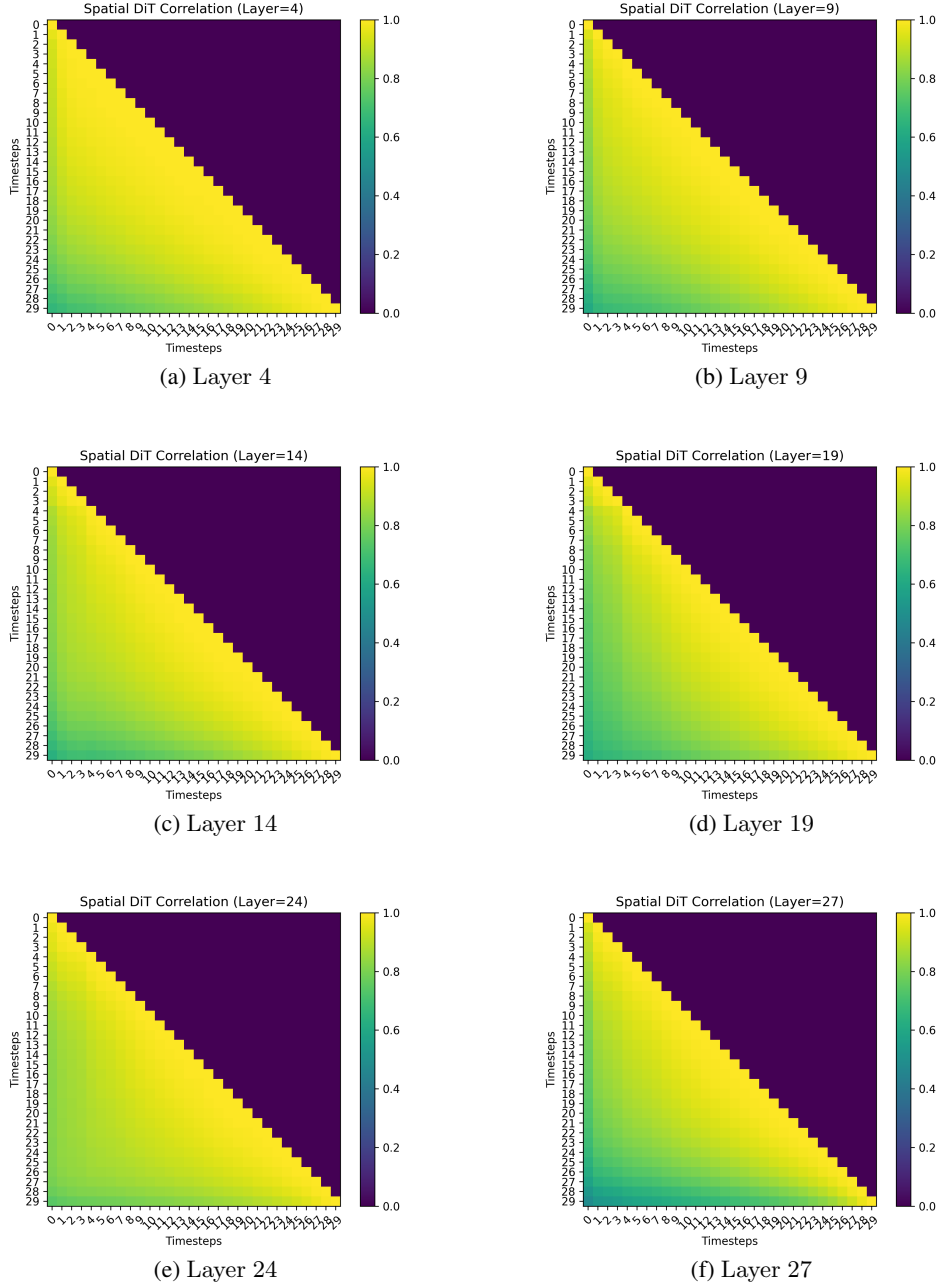


Figure 14: Cosine similarity of Spatial-DiT features across denoising steps for different layers of the OpenSora model.

841 We compute all the above similarity metrics per frame and report the average across all frames as the  
842 final video score.

#### 843 A.6 Baselines Generation Settings

844 We evaluate four static cache reuse methods—Static,  $\Delta$ -DiT[Chen et al., 2024b], T-GATE[Liu  
845 et al., 2024b], and PAB[Zhao et al., 2024b]—using the configurations detailed below.

Static method caches coarse-grained features using a reuse window of size  $N$  and updates the cache at a fixed compute interval  $R$ , as shown in Table 4.

Table 4: Static baseline Settings

Static	Denoising Steps ( $T$ )	Reuse Window ( $N$ )	Compute Interval ( $R$ )
Open-Sora	30	1	2
Latte	50	1	2
CogVideoX	50	1	2

$\Delta$ -DiT caches feature map deviations instead of full feature maps. It applies to back blocks during early outline generation and to front blocks during the detail refinement stage. Cache reuse is controlled by a gating hyperparameter  $b$ , which defines the boundary between front and back blocks, and a cache interval  $k$  as shown in Table 5.

Table 5:  $\Delta$ -DiT baseline Settings

$\Delta$ -DiT	Denoising Steps ( $T$ )	Cache Interval ( $k$ )	Gate Step ( $b$ )	Block Range
Open-Sora	30	2	25	[0, 5]
Latte	50	2	48	[0, 2]
CogVideoX	50	2	48	[0, 2]

T-GATE divides the diffusion process into two phases: semantics planning and fidelity improvement, with the transition defined by gate step  $m$ . During the semantics-planning phase, cross-attention (CA) remains active, while self-attention (SA) is computed and reused every  $k$  steps after an initial warm-up. After step  $m$ , CA is replaced by cached features, while SA continues as shown in Table 6.

Table 6: T-GATE baseline Settings

T-GATE	Denoising Steps ( $T$ )	Cache Interval ( $k$ )	Gate Step ( $b$ )
Open-Sora	30	2	12
Latte	50	2	20
CogVideoX	50	2	20

PAB employs Pyramid Attention Broadcast, where the broadcast range forms a hierarchy: cross-attention ( $\gamma$ ) at the base, temporal attention ( $\beta$ ) in the middle, and spatial attention ( $\alpha$ ) at the top. Reuse occurs during designated broadcast timesteps. Each DiT block’s MLP follows a separate reuse schedule, as detailed in Table 7, based on empirical evaluation.

## A.7 Additional Results

To quantify the adaptive behavior of Foresight, we plot absolute latency for all methods across prompts from the Open-Sora set. As shown in Figure 15, static reuse methods exhibit consistent latency due to fixed reuse schedules. In contrast, Foresight adjusts latency based on prompt complexity, enabling dynamic reuse for improved video quality and inference speedup.

To complement Section 4.2, which reports results using VBench [Huang et al., 2024] prompts and standard similarity metrics, we further evaluate Foresight on two additional prompt sets: UCF-101 [Soomro et al., 2012] and EvalCrafter [Liu et al., 2024d]. We include CLIPSIM and CLIP-Temp to assess text-video alignment and temporal consistency, along with DOVER’s [Wu et al., 2023] VQA metrics for both aesthetic and technical quality, as shown in Table 8.

Table 7: PAB baseline Settings

PAB	Denoising Steps ( $T$ )	Broadcast Range and Timesteps				Reuse	MLP Broadcast	
		Spatial ( $\alpha$ )	Temporal ( $\beta$ )	Cross ( $\gamma$ )	Timesteps		Blocks	Timesteps
Open-Sora	30	2	4	6	[930-450]	2	[0, 1, 2, 3, 4]	[864, 788, 676]
Latte	50	2	4	6	[800-100]	2	[0, 1, 2, 3, 4]	[720, 640, 560, 480, 400]
CogVideoX	50	2	-	-	[850, 100]	-	-	-

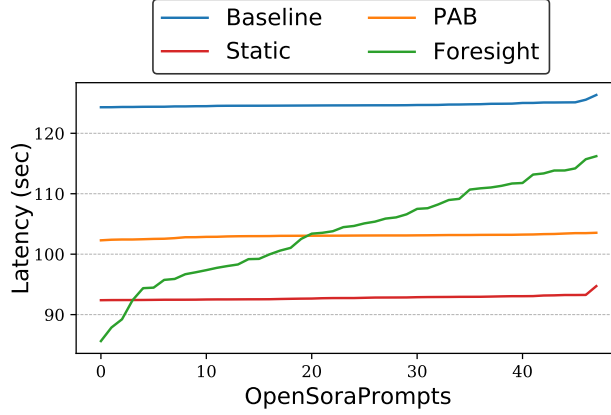


Figure 15: Latency variation across prompts from the Open-Sora set. Baseline and Static methods yield constant inference latency due to fixed reuse schedules. In contrast, Foresight adapts reuse based on prompt complexity, improving video quality with dynamic latency. Prompts are sorted by latency (ascending) using the Open-Sora model at 720p for 2-second generations.

Model	Method	CLIP SIM	CLIP Temp	VQA Aesthetic	VQA Technical	VQA Overall	Latency(s)	Speedup
<b>UCF-101 [Soomro et al., 2012] (101 Prompts)</b>								
<b>Open-Sora</b>	Baseline	20.51	99.64	14.56	25.02	19.75	12.71 ( $\pm 0.04$ )	-
	PAB	20.47	99.76	6.57	15.66	10.32	10.05 ( $\pm 0.06$ )	1.26 $\times$
	Foresight ( $N_1 R_2$ )	20.73	99.88	15.17	26.68	21.11	9.72 ( $\pm 0.84$ )	1.30 $\times$
	Foresight ( $N_2 R_3$ )	20.70	99.86	14.01	26.11	20.32	7.32 ( $\pm 0.80$ )	1.73 $\times$
<b>Latte</b>	Baseline	20.09	99.39	20.71	12.93	14.10	32.48 ( $\pm 0.02$ )	-
	PAB	20.09	99.39	20.71	12.69	13.79	25.11 ( $\pm 0.01$ )	1.29 $\times$
	Foresight ( $N_1 R_2$ )	20.12	99.14	22.72	21.04	20.34	21.48 ( $\pm 0.05$ )	1.51 $\times$
	Foresight ( $N_2 R_3$ )	20.09	99.16	21.95	19.14	18.92	15.56 ( $\pm 0.22$ )	2.08 $\times$
<b>CogVideoX</b>	Baseline	20.22	99.43	40.25	42.34	41.19	29.35 ( $\pm 0.02$ )	-
	PAB	20.20	99.43	39.44	41.86	40.52	22.65 ( $\pm 0.05$ )	1.29 $\times$
	Foresight ( $N_1 R_2$ )	20.21	99.42	40.64	40.65	40.17	19.57 ( $\pm 0.96$ )	1.49 $\times$
	Foresight ( $N_2 R_3$ )	20.17	99.44	41.46	43.34	42.15	17.00 ( $\pm 1.18$ )	1.72 $\times$
<b>EvalCrafter [Liu et al., 2024d] (150 prompts)</b>								
<b>Open-Sora</b>	Baseline	20.07	99.55	17.48	29.94	23.49	12.59 ( $\pm 0.09$ )	-
	PAB	19.91	99.76	9.16	20.82	14.15	10.07 ( $\pm 0.03$ )	1.24 $\times$
	Foresight ( $N_1 R_2$ )	20.05	99.54	16.85	29.56	23.10	9.36 ( $\pm 0.77$ )	1.34 $\times$
	Foresight ( $N_2 R_3$ )	20.05	99.50	15.67	29.21	22.24	7.06 ( $\pm 0.93$ )	1.78 $\times$
<b>Latte</b>	Baseline	20.69	99.50	53.05	45.59	48.37	32.48 ( $\pm 0.02$ )	-
	PAB	19.98	99.65	53.35	35.39	40.84	25.11 ( $\pm 0.01$ )	1.29 $\times$
	Foresight ( $N_1 R_2$ )	20.60	99.51	55.85	45.80	49.27	28.37 ( $\pm 1.14$ )	1.14 $\times$
	Foresight ( $N_2 R_3$ )	20.54	99.50	54.00	44.03	47.29	25.59 ( $\pm 1.36$ )	1.26 $\times$
<b>CogVideoX</b>	Baseline	20.66	99.51	54.72	57.67	56.53	30.43 ( $\pm 0.12$ )	-
	PAB	20.66	99.51	52.49	56.10	54.85	22.95 ( $\pm 0.07$ )	1.32 $\times$
	Foresight ( $N_1 R_2$ )	20.66	99.55	53.34	56.73	55.64	25.80 ( $\pm 0.84$ )	1.17 $\times$
	Foresight ( $N_2 R_3$ )	20.64	99.55	52.54	56.34	54.98	24.56 ( $\pm 1.08$ )	1.23 $\times$

Table 8: Qualitative comparison of Foresight and PAB on UCF [Soomro et al., 2012] and EvalCrafter [Liu et al., 2024d] prompts set. Videos are generated at 240p with OpenSora, 512x512 with Latte, and 480x720 with CogVideoX, all with a fixed duration of 2 seconds. Metrics including EvalCrafter’s Liu et al. [2024d] CLIP score and DOVER’s [Wu et al., 2023] VQA in Aesthetics, Technical and Overall score.