Adaptive Spectral Feature Forecasting for Diffusion Sampling Acceleration

Supplementary Material

A. Proofs

A.1. Proof of Theorem 3.1

Proof. The proof follows directly from Taylor's theorem with the Lagrange form of the remainder. Let $f \in \mathcal{F}_{P+1}(L)$. For any $\tau_j, \tau_k \in [0,1]$, Taylor's theorem states that there exists a ξ between τ_k and τ_j such that:

$$f(\tau_j) = \sum_{p=0}^{P} \frac{f^{(p)}(\tau_k)}{p!}(\tau_j - \tau_k)^p + \frac{f^{(P+1)}(\xi)}{(P+1)!}(\tau_j - \tau_k)^{P+1}.$$

The predictor $T_P[f](\tau_j)$ is defined as the first term (the Taylor polynomial). Thus, the approximation error is the absolute value of the remainder term:

$$|f(\tau_j) - T_P[f](\tau_j)| = \left| \frac{f^{(P+1)}(\xi)}{(P+1)!}(\tau_j - \tau_k)^{P+1} \right|.$$

Since $f \in \mathcal{F}_{P+1}(L)$, we have $\|f^{(P+1)}\|_\infty \leq L$, which implies $|f^{(P+1)}(\xi)| \leq L$. Substituting the step size $\tau_j - \tau_k = (j-k)\delta_t$, we obtain the upper bound:

$$|f(\tau_j) - T_P[f](\tau_j)| \leq \frac{L}{(P+1)!} \left((j-k)\delta_t\right)^{P+1}.$$

To show that this is the supremum (worst-case error), consider the specific function $f^*(\tau) = \frac{L}{(P+1)!}(\tau - \tau_k)^{P+1}$. The $(P+1)$-th derivative of f^* is identically L, so $f^* \in \mathcal{F}_{P+1}(L)$. For this function, the derivatives $f^{*(p)}(\tau_k)$ are 0 for all $p \leq P$, meaning $T_P[f^*](\tau_j) = 0$. The error is exactly $|f^*(\tau_j)| = \frac{L}{(P+1)!}((j-k)\delta_t)^{P+1}$, which attains the bound. □

A.2. Proof of Theorem 3.2

Proof. This result is a standard theorem in approximation theory, often referred to as Bernstein's Theorem for Chebyshev approximation. Recall that any function f analytic on the Bernstein ellipse E_ρ can be expanded in a Chebyshev series $f(\tau) = \sum_{k=0}^{\infty} a_k T_k(\tau)$. The coefficients a_k satisfy the geometric decay bound $|a_k| \leq 2B\rho^{-k}$ for $k \geq 1$, where $B = \sup_{z \in E_\rho} |f(z)|$.

The truncation error $f(\tau) - p_M(\tau)$ consists of the tail of the series:

$$f(\tau) - p_M(\tau) = \sum_{k=M+1}^{\infty} a_k T_k(\tau).$$

Taking the infinity norm on $[-1,1]$ and using the property that $|T_k(\tau)| \leq 1$ for $\tau \in [-1,1]$:

$$\|f - p_M\|_\infty \leq \sum_{k=M+1}^{\infty} |a_k| \leq \sum_{k=M+1}^{\infty} 2B\rho^{-k}.$$

The right-hand side is a geometric series with ratio $\rho^{-1} < 1$:

$$\sum_{k=M+1}^{\infty} 2B\rho^{-k} = 2B\frac{\rho^{-(M+1)}}{1-\rho^{-1}} = \frac{2B\rho^{-M}}{\rho - 1}.$$

Thus, $\|f - p_M\|_\infty \leq \frac{2B}{\rho-1}\rho^{-M}$. □

A.3. Proof of Theorem 3.3

Let $h_i(t)$ be the true feature function and $p_M(t) = \boldsymbol{\phi}(\tau)\mathbf{C}^*$ be its optimal Chebyshev approximation of degree M, where $\mathbf{C}^*$ is the vector of ideal Chebyshev coefficients. From Theorem 3.2, we define the truncation error function $e(t) = h_i(t) - p_M(t)$, bounded by $|e(t)| \leq \epsilon_M$.

We observe data at cached timesteps $\mathbb{C}_{t_j} = \{(\mathbf{h}_{t_k}, t_k)\}$. Let $\mathbf{H} \in \mathbb{R}^K$ be the vector of observed values for channel i (dropping indices t_j for simplicity). We can write the observation model as:

$$\mathbf{H} = \boldsymbol{\Phi}\mathbf{C}^* + \mathbf{E},$$

where $\boldsymbol{\Phi}$ is the design matrix and $\mathbf{E}$ is the vector of truncation errors at the cached points, with $\|\mathbf{E}\|_\infty \leq \epsilon_M$. The fitted coefficients $\widehat{\mathbf{C}}$ are obtained via ridge regression:

$$\widehat{\mathbf{C}} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^\top\mathbf{H}.$$

Substituting $\mathbf{H}$:

$$\widehat{\mathbf{C}} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\mathbf{C}^* + \mathbf{E}).$$

We want to bound the prediction error at a new time t_j (with $\tau_j = g(t_j)$). The true value is $h_i(t_j) = \boldsymbol{\phi}(\tau_j)\mathbf{C}^* + e(t_j)$, and the prediction is $\widehat{h}_i(t_j) = \boldsymbol{\phi}(\tau_j)\widehat{\mathbf{C}}$. The error is:

$$\widehat{h}_i(t_j) - h_i(t_j) = \boldsymbol{\phi}(\tau_j)(\widehat{\mathbf{C}} - \mathbf{C}^*) - e(t_j).$$

Let $\mathbf{A} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\mathbf{I}$. Note that $\mathbf{A}^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \mathbf{I} - \lambda\mathbf{A}^{-1}$. The coefficient error is:

$$\begin{aligned}
\widehat{\mathbf{C}} - \mathbf{C}^* &= \mathbf{A}^{-1}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\mathbf{C}^* + \mathbf{E}) - \mathbf{C}^* \\
&= (\mathbf{A}^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \mathbf{I})\mathbf{C}^* + \mathbf{A}^{-1}\boldsymbol{\Phi}^\top\mathbf{E} \\
&= -\lambda\mathbf{A}^{-1}\mathbf{C}^* + \mathbf{A}^{-1}\boldsymbol{\Phi}^\top\mathbf{E}.
\end{aligned}$$

Substituting this back into the prediction error:

$$\widehat{h}_i(t_j) - h_i(t_j) = \boldsymbol{\phi}(\tau_j)\mathbf{A}^{-1}\boldsymbol{\Phi}^\top\mathbf{E} - \lambda\boldsymbol{\phi}(\tau_j)\mathbf{A}^{-1}\mathbf{C}^* - e(t_j).$$

Using the triangle inequality:

$$|\widehat{h}_i - h_i| \leq |\boldsymbol{\phi}(\tau_j)\mathbf{A}^{-1}\boldsymbol{\Phi}^\top\mathbf{E}| + |\lambda\boldsymbol{\phi}(\tau_j)\mathbf{A}^{-1}\mathbf{C}^*| + |e(t_j)|.$$

Now we bound each term.

For the first term, we have $|\boldsymbol{\phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top\mathbf{E}| \leq \|\boldsymbol{\phi}\|_2\|\mathbf{A}^{-1}\|_2\|\boldsymbol{\Phi}^\top\mathbf{E}\|_2$. Since $|T_m| \leq 1$, we have $\|\boldsymbol{\phi}(\tau_j)\|_2 \leq \sqrt{M+1}$. We also have $\|\mathbf{A}^{-1}\|_2 \leq \frac{1}{\sigma^2_{\min}(\boldsymbol{\Phi})+\lambda}$, and $\|\boldsymbol{\Phi}^\top\mathbf{E}\|_2 = \|\sum_{k=1}^K \boldsymbol{\phi}(\tau_k)^\top E_k\|_2 \leq \sum_{k=1}^K \|\boldsymbol{\phi}(\tau_k)\|_2|E_k| \leq K\sqrt{M+1}\epsilon_M$. Combining the above, we have $|\boldsymbol{\phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top\mathbf{E}| \leq \frac{(M+1)K}{\sigma^2_{\min}+\lambda}\epsilon_M$.

For the second term, we have $|\lambda\boldsymbol{\phi}\mathbf{A}^{-1}\mathbf{C}^*| \leq \lambda\|\boldsymbol{\phi}\|_2\|\mathbf{A}^{-1}\|_2\|\mathbf{C}^*\|_2$. By the coefficient bound, we get

$$\|\mathbf{C}^*\|_2 \leq \sqrt{\sum_{m=0}^{\infty} 4B^2\rho^{-2m}} = 2B\sqrt{\frac{1}{1-\rho^{-2}}}.$$

Combining with the bound in the first term, we have $|\lambda\boldsymbol{\phi}\mathbf{A}^{-1}\mathbf{C}^*| \leq \frac{\lambda\sqrt{M+1}}{\sigma^2_{\min}+\lambda}\frac{2B}{\sqrt{1-\rho^{-2}}}$.

For the third term, $|e(t_j)| \leq \epsilon_M$.

Therefore, combining the bound for the three terms above gives the final bound:

$$|h_i(t_j) - \widehat{h}_i(t_j)| \leq \epsilon_M\left(1 + \frac{(M+1)K}{\sigma^2_{\min}+\lambda}\right) + \frac{\lambda\sqrt{M+1}}{\sigma^2_{\min}+\lambda}\frac{2B}{\sqrt{1-\rho^{-2}}}.$$

B. More Method Details

B.1. Adaptive Scheduling

Most previous works [22, 39, 59] on diffusion caching employ a uniform scheduling in selecting $\mathbb{U}$ *i.e.*, the set of timesteps to perform full network evaluation. For this strategy, a full network evaluation is performed every fixed interval of length $\mathcal{N} \in \mathbb{N}^+$. Under our framework, this corresponds to $\mathbb{U}_{\text{uniform}} = \{1\} \cup \{\tau_j : j = r\mathcal{N}, r \in \mathbb{N}^+, 1 \leq j \leq N\}$, where $N = 50$ is the total number of discretization timesteps. However, this scheduling does not account for the fact that errors made in early diffusion steps accumulate and disproportionately affect later steps, eventually degrading the final output more significantly.

Other works [21, 25, 29, 48] attempt to address this by adaptively adjusting the scheduling $\mathbb{U}$, but their methods typically require tracking inference time metrics, which introduces additional computation overheads. In contrast, we propose a simple, precomputed adaptive scheduling that progressively increases the interval length between adjacent timesteps, formulated as:

$$\mathbb{U} = \{\tau_j : j = \lfloor \alpha \frac{r(r+1)}{2} \rfloor, r \in \mathbb{N}^+, 1 \leq j \leq N\},$$

where $\alpha \geq 0$ characterizes the *rate of increase* of the interval $\tau_j - \tau_k$ between adjacent timesteps j and k.

Unified parameterization. Besides this progressively increasing recomputation intervals, our actual implementation has two additional components: the initial interval size $\mathcal{N}$ and the number of warm-up steps $\mathcal{W}$, which are *directly inherited* from the original setup in TaylorSeer [22]. The initial interval size $\mathcal{N}$ determines the length of the first interval, from which subsequent intervals grow. The warm-up step $\mathcal{W}$ corresponds to the length of the initial phase during which full network evaluations are computed at every step, before the caching and forecasting procedure begins. The detailed formulation of *Spectrum*'s adaptive scheduling incorporated with these two hyperparameters is as follows:

$$\mathbb{U} = \{\tau_j : j \in \mathbb{N}^+,\ 1 \leq j \leq \mathcal{W}\} \cup \left\{\tau_j : j = \mathcal{W} + \left\lfloor (r+1)\mathcal{N} + \alpha \frac{r(r+1)}{2} \right\rfloor, r \in \mathbb{N},\ 1 \leq j \leq N\right\}.$$

Therefore, the uniform scheduling $\mathbb{U}_{\text{uniform}}$ in previous works can be exactly recovered by setting $\alpha = 0.0$.

Under this unified parameterization, we provide detailed specifications of the baselines, which are parameterized by $\mathcal{N}$ and $\mathcal{W}$, and *Spectrum*, which is additionally parameterized by α, in Table 6. For clarity, we also include their corresponding number of network evaluations (NFE), which is closely related to their actual wall clock time reported in the main tables.

C. More Experiment Details

C.1. Model Settings

Text-to-image. We employ FLUX.1-dev[3] and Stable Diffusion 3.5-Large[4] for text-to-image generation. For the main experiments, we generate images with 1024×1024 resolution. We apply the default guidance scale, which is 3.5 for FLUX and 7.0 for Stable Diffusion 3.5-Large.

Text-to-video. We adopt Wan2.1-14B[5] and HunyuanVideo[6] for text-to-video generation. We generate 480p videos with 81 frames following the prompt suite in VBench [12] and strictly follow the benchmark protocol of VBench to evaluate the quality metrics and VBench Quality Score. Following convention, we apply a guidance scale of 5.0 for Wan2.1-14B and 6.0 for HunyuanVideo throughout all experiments.

C.2. Baseline Settings

For the baselines that employ a uniform activation scheduling (FORA, ToCa, and TaylorSeer), we set interval size $\mathcal{N}$ to be 4 in the slow acceleration setting and 6 in the high acceleration setting. We set $\mathcal{W} = 1$ for FORA, $R = 90\%, \mathcal{W} = 3$ for ToCa, and $\mathcal{W} = 5$ for TaylorSeer to match the evaluation setting in Liu et al. [22]. TeaCache employs a non-deterministic scheduling that may vary across different runs, but we found that setting the caching threshold δ to 0.2 and 0.8 results in similar acceleration rates.

For *Spectrum*, we use the same number of warm-up steps ($\mathcal{W} = 5$) as TaylorSeer and set $\mathcal{N} = 2$ for the settings that use the adaptive activation scheduling. We set $\alpha = 0.75$ and $\alpha = 3.0$, respectively, for the slow and high acceleration settings.

[3] `black-forest-labs/FLUX.1-dev`
[4] `stabilityai/stable-diffusion-3.5-large`
[5] `Wan-AI/Wan2.1-T2V-14B`
[6] `hunyuanvideo-community/HunyuanVideo`

Table 6. **Detailed specification on the scheduler and the total number of network evaluations (NFE) for all methods.** "Reference" refers to the tables in the main paper where the corresponding method was mentioned.

	Reference	$\mathcal{N}$	$\mathcal{W}$	α	NFE
FORA ($\mathcal{N} = 4$)	Table 1, 2	4	1	0.0	13
ToCa ($\mathcal{N} = 4$)	Table 1, 2	4	3	0.0	14
TaylorSeer ($\mathcal{N} = 4$)	Table 1, 2	4	5	0.0	16
Spectrum ($\alpha = 0.75$)	Table 1, 2	2	5	0.75	14
FORA ($\mathcal{N} = 6$)	Table 1, 2	6	1	0.0	9
ToCa ($\mathcal{N} = 6$)	Table 1, 2	6	3	0.0	10
TaylorSeer ($\mathcal{N} = 6$)	Table 1, 2	6	5	0.0	12
Spectrum ($\alpha = 3.0$)	Table 1, 2	2	5	3.0	10
TaylorSeer ($\mathcal{N} = 8$)	Table 3	8	5	0.0	10
TaylorSeer ($\alpha = 3.0$)	Table 3	2	5	3.0	10
Spectrum ($\mathcal{N} = 8$)	Table 3	8	5	0.0	10
Spectrum ($\alpha = 3.0$)	Table 3	2	5	3.0	10

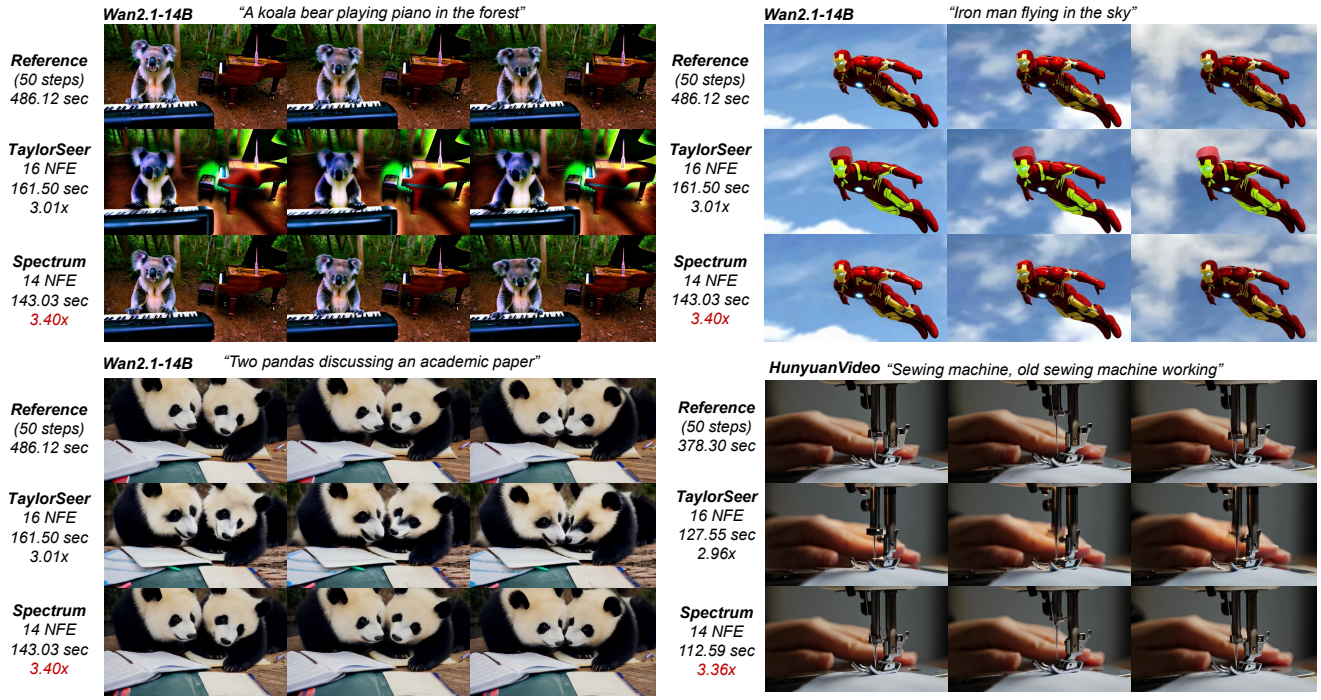


Figure 7. **Additional qualitative comparison on text-to-video generation** using Wan2.1-14B and HunyuanVideo.

C.3. Ablation Study Settings

We conduct the ablation on the adaptive activation scheduling in the high-speed up setting to better visualize its impact. To ensure a fair comparison under the same computation budget, we set the Taylor polynomial method's interval size to $\mathcal{N} = 8$, which lets it consume the same number of network evaluations (NFE) as *Spectrum* with $\alpha = 3.0$. We conducted the ablation study on the effect of last-block forecasting at the preliminary stage of our project, so we used the uniform activation scheduling with $N = 8$ for both the Taylor method and *Spectrum*. We use adaptive scheduling for the ablation study on the regularization weight λ and the degree of the Chebyshev polynomial M. The 5 different acceleration ratios in Figs. 5 and 6 correspond to the α values $[0.2, 0.4, 0.75, 1.5, 3.0]$, which incur the following NFEs $[20, 17, 14, 12, 10]$.

Table 7. Benchmark results of **text-to-image generation** on COCO2017 [19] using FLUX.1. We use 50 steps as the reference (†). Our *Spectrum* achieves higher speedup while maintaining better sample quality.

	FLUX.1 [17]						
	Acceleration		Quality			Image Reward↑	CLIP↑
	Latency(s)↓	Speedup↑	PSNR↑	SSIM↑	LPIPS↓		
50 steps†	26.46	1.00	-	-	-	1.13	25.92
FORA ($\mathcal{N} = 6$) [39]	6.37	4.16	13.94	0.604	0.523	1.04	**26.10**
ToCa ($\mathcal{N} = 6, \mathcal{R} = 0.9$) [59]	13.51	1.96	16.14	0.652	0.462	1.10	**26.10**
TeaCache ($\delta = 0.8$) [21]	6.73	3.94	16.07	0.664	0.436	1.03	25.88
TaylorSeer ($\mathcal{N} = 6, \mathcal{O} = 1$) [22]	6.52	4.06	19.92	0.782	0.297	1.10	25.98
Spectrum ($\alpha = 3.0$)	**5.58**	**4.75**	**22.29**	**0.811**	**0.288**	**1.11**	25.95

Figure 8. **Qualitative comparison on text-to-image generation** using Stable Diffusion 3.5-Large

D. More Results

Evaluation on COCO2017 captions dataset. We provide additional results on the COCO2017 captions dataset [19] using FLUX.1 in Table 7. *Spectrum* consistently exhibits exceptional performance on the new set of prompts.

More qualitative results. We provide additional qualitative comparisons on text-to-image generation with FLUX.1 in Fig. 9

Figure 9. **Additional qualitative comparison on text-to-image generation** using FLUX.1.

Table 8. RMSE between the predicted latent features and oracle.

	Diffusion step	10	20	30	40	50
Hunyuan	TaylorSeer	0.0047	0.0102	0.0206	0.0467	0.1562
	Spectrum	0.0021	0.0046	0.0081	0.0182	0.0818

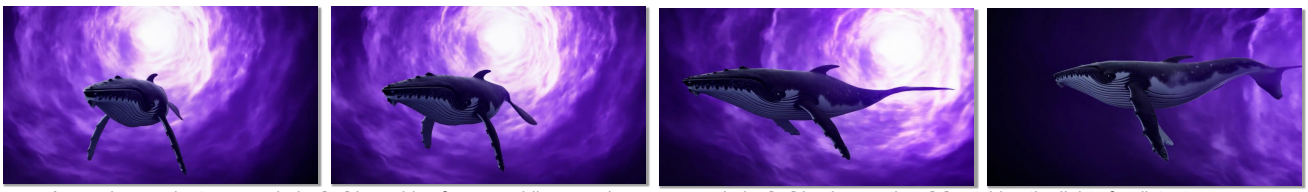
	Taylor	*Spectrum*
PNSR↑	21.87	24.95
SSIM↑	0.768	0.793
LPIPS↓	0.271	0.253
Speedup↑	2.6×	2.8×

Figure 10. Additional results on SDXL (U-Net architecture).

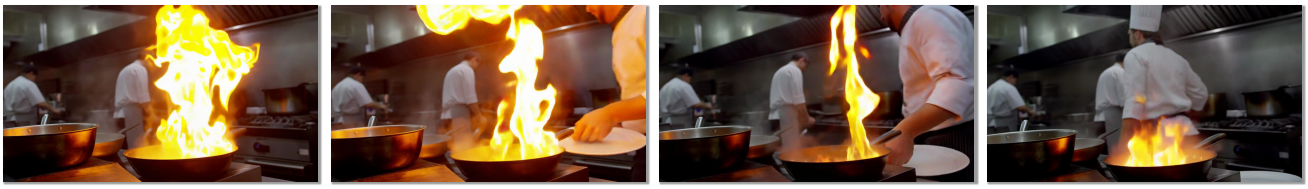
and with Stable Diffusion 3.5-Large in Fig. 8. We also provide more qualitative comparisons on text-to-image generation with Wan2.1-14B and HunyuanVideo in Fig. 7. We offer more visualizations of the generated video samples with *Spectrum* using HunyuanVideo in Fig. 11.

More latent feature results. We provide RMSE on latent features on HunyuanVideo in Table 8.

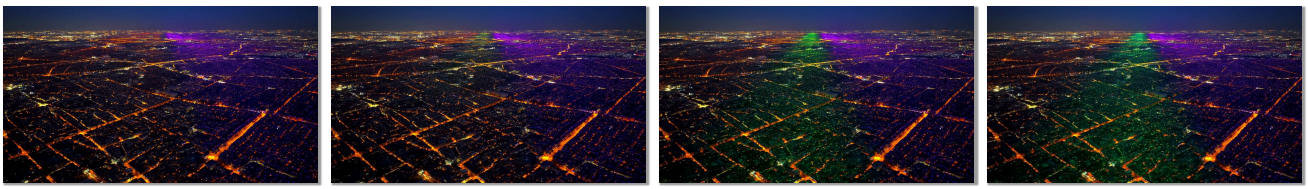
Generalization to U-Net architecture. We additionally investigate the efficacy of *Spectrum* on SDXL with U-Net architecture. Results are summarized in Fig. 10, which shows that *Spectrum* remains highly effective on SDXL, outperforming TaylorSeer by a significant margin.



A massive, ancient space whale, […] breaching from a swirling, purple gaseous nebula. […] in slow motion, [..] catching the light of a distant supernova.

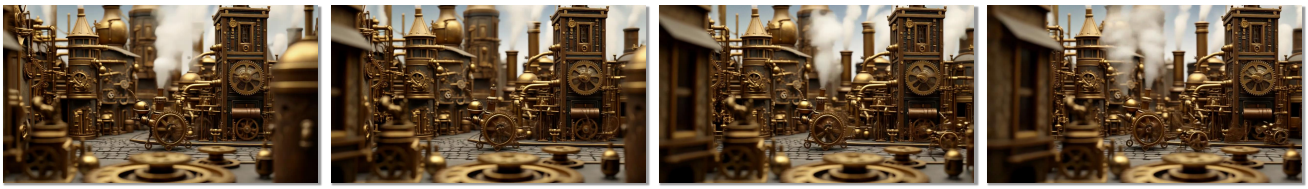


A chef in a bustling kitchen […] The chef chops and plates food with incredible speed, while flames erupt from pans, steam billows from pots […]

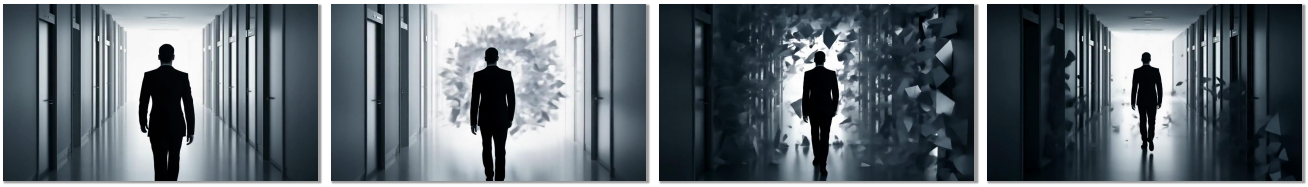


An aerial shot of a vast city at night, but each individual light source […] pulses and shifts through a full spectrum of colors […]

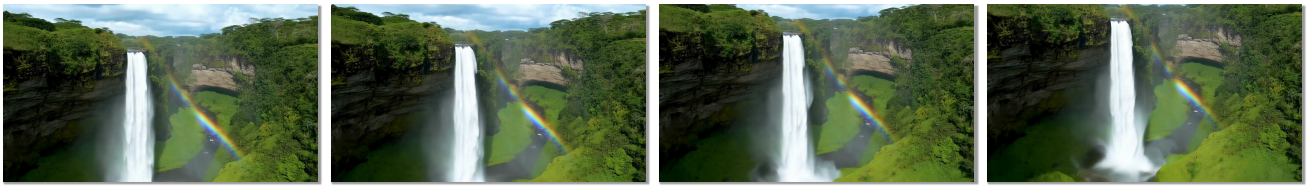
A hallway made entirely of mirrors, reflecting a person walking, but the reflections are out of sync and show different versions of the person



A macro dolly-in shot of a complex, clockwork city, tiny brass gears turning and steam escaping from miniature pipes as miniature automatons walk the streets



A person walking down a hallway, but with every step, the walls, floor, and ceiling fracture into geometric shards that swirl around them […]



A drone shot flying straight up the face of a colossal, thundering waterfall in a lush, tropical gorge […] while a rainbow forms in the drifting mist

Figure 11. More **text-to-video generation samples** on HunyuanVideo with *Spectrum*. Samples were generated using only **14 network evaluations**, leading to a significant speedup of 3.5× without quality degradation.