

A Technical Appendices and Supplementary Material

A.1 Prompts in Figure 1

We utilize the following three prompts to generate the average magnitude ratio, magnitude ratio variability, and residual cosine distance. In all experiments, we only utilize *Prompt 1* to calibrate the average magnitude ratio.

- *Prompt 1*: A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.
- *Prompt 2*: In a still frame, a stop sign
- *Prompt 3*: a laptop, frozen in time

Table 4: The list of prompts in Figure 1. In all experiments, we only utilize prompt 1 to calibrate the magnitude ratio for MagCache.

A.2 Definition of Statistics in Figure 1

In Figure 1, we define three metrics: the average magnitude ratio, magnitude ratio variability, and residual cosine distance. The average magnitude ratio γ is defined in Equation 7. Specifically, Equation 7 first computes the L2 norm of the residuals \mathbf{r}_t and \mathbf{r}_{t-1} along the channel dimension, then takes the token-wise ratio, and finally averages the result across the sequence length dimension to obtain γ_t . *The mean operation is omitted in Equation 7.*

The magnitude ratio variability σ and residual cosine distance $dist$ can be represented as follows:

Magnitude Ratio Variability.

$$\sigma_t = std\left(\frac{\|\mathbf{r}_t\|_2}{\|\mathbf{r}_{t-1}\|_2}\right), \quad (12)$$

where $\mathbf{r}_t \in \mathbb{R}^{N \times d}$ denotes the residual at timestep t , and $\|\cdot\|_2$ represents the L2 norm computed along the channel dimension d . The standard deviation is then calculated across the sequence length dimension N .

Residual Cosine Distance.

$$dist_t = \frac{1}{N} \sum_i^N (1 - \cos(\mathbf{r}_t^i, \mathbf{r}_{t-1}^i)). \quad (13)$$

Here, the cosine distance is computed for each token between residuals at timesteps t and $t-1$, and the final residual cosine distance $dist_t$ is obtained by averaging across all tokens.

A.3 Experiments on More Tasks and Models

In this section, we further conduct experiments on a wider range of tasks and models to validate the magnitude law and the effectiveness of our proposed MagCache. *The magnitude law is universally applicable to more diffusion-based models and tasks*, such as HunyuanVideo and Flux in text-to-image task. As shown in Figure 4, if we exclude the first 20%, we observe that in the 20%–80% range, the average magnitude ratio steadily decreases, and both the magnitude ratio variability and the token-wise cosine distance remain close to zero. In the final 20% of steps, the magnitude ratio drops rapidly. These findings are consistent with our earlier observations in Figure 1. It suggests that intermediate steps exhibit considerable redundancy, and the magnitude ratio provides a stable and reliable way to quantify residual differences across timesteps. Meanwhile, the observation in Figure 4 also demonstrates that the first 20% diffusion steps with high uncertainty are important to the whole diffusion process and should be preserved.

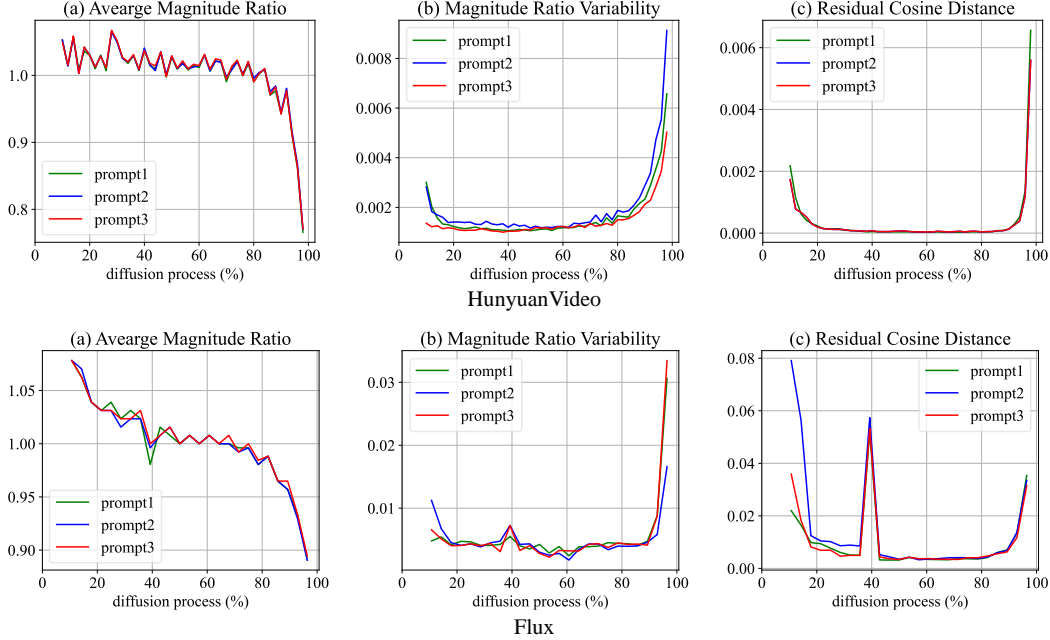


Figure 4: Relationships of HunyuanVideo and Flux between residuals across diffusion timesteps. Excluding the first 20% of steps with high variability, the observations of HunyuanVideo and Flux are consistent with those in Figure 1. These early steps are crucial to the diffusion process and are preserved in MagCache. Notably, in Flux, an abnormally high residual cosine distance occurs around the 40% mark of the diffusion process. To avoid its negative effect, MagCache preserves this step by ensuring it is always computed.

Table 5: Quantitative evaluation of inference efficiency and visual quality in Hunyuan Video (video generation) and Flux (text-to-image generation). MagCache consistently achieves superior speed and significantly better visual quality in these new settings. TeaCache results are reported using official code and configurations.

Method	Efficiency			Visual Quality		
	FLOPs (P) ↓	Speedup ↑	Latency (s) ↓	LPIPS ↓	SSIM ↑	PSNR ↑
HunyuanVideo (129 frames, 536P)						
HunyuanVideo ($T = 50$)	45.93	1×	1163	-	-	-
TeaCache-slow [30]	27.56	1.63×	712	0.1832	0.7876	23.87
TeaCache-fast [30]	20.21	2.26×	514	0.1971	0.7744	23.38
MagCache-slow	20.21	2.25×	516	0.0377	0.9459	34.51
MagCache-fast	18.37	2.63×	441	0.0626	0.9206	31.77
Flux (Text-to-Image 1024 × 1024)						
Flux ($T = 28$)	1.66	1×	14.26	-	-	-
TeaCache-slow [30]	0.77	2.00×	7.11	0.2687	0.7746	20.14
TeaCache-fast [30]	0.59	2.52×	5.65	0.3456	0.7021	18.17
MagCache-slow	0.59	2.57×	5.53	0.2043	0.8883	24.46
MagCache-fast	0.53	2.82×	5.05	0.2635	0.8093	21.35

519 A.3.1 Video Generation on Hunyuan Video

520 We utilize the first 100 prompts in Vbench to generate videos for evaluation. In the comparison on
521 HunyuanVideo presented in Table 5, MagCache-slow reduces computation from 45.93 PFLOPs to
522 20.21 PFLOPs, achieving a 2.25× speedup with a latency of 516 seconds, all while largely preserving
523 visual quality. In contrast, TeaCache-fast, despite a comparable latency, results in substantially lower
524 visual fidelity, with an LPIPS of 0.1971, SSIM of 0.7744, and PSNR of 23.38 dB. MagCache-fast
525 further reduces computation to 18.37 PFLOPs, achieving a 2.63× speedup and a latency of 441

seconds, while maintaining high visual quality—LPIPS of 0.0626, SSIM of 0.9206, and PSNR of 31.77 dB. Both MagCache variants offer significant gains in efficiency and visual fidelity compared to standard Hunyuan Video inference and the existing TeaCache method. For HunyuanVideo, we retain the first 20% of steps unchanged. MagCache-slow uses $K = 4$, $\sigma = 0.12$; MagCache-fast uses $K = 4$, $\sigma = 0.24$. See Figure 9 for qualitative comparisons.

A.3.2 Text-to-Image Generation on Flux

We utilize the 553 textual prompts in GenEval to generate images for evaluation. On 1024×1024 image generation with Flux, MagCache-slow reduces FLOPs from 1.66 P to 0.59 P and speeds up inference by 2.57 \times with a runtime of 5.53 s while improving visual quality to an LPIPS of 0.2043, an SSIM of 0.8883 and a PSNR of 24.46 dB. MagCache-fast further cuts FLOPs to 0.53 P and accelerates processing by 2.82 \times to a runtime of 5.05 s while keeping high image fidelity with an LPIPS of 0.2635, an SSIM of 0.8093 and a PSNR of 21.35 dB. In contrast, the image quality of TeaCache-fast degrades significantly with an LPIPS of 0.3456, an SSIM of 0.7021 and a PSNR of 18.17 dB. These results demonstrate that MagCache-fast not only offers the highest inference speed but also brings a significant improvement in visual quality. MagCache-slow is configured with $K = 4$, $\sigma = 0.24$, and keeps the first 20% of steps unchanged. MagCache-fast uses $K = 5$, $\sigma = 0.24$, and preserves the first 10% of steps. See Figure 10 for qualitative comparisons.

A.4 More Ablations

A.4.1 Computation of Skip Error in Equation 6

In Section 3.3, we adopt the multiplicative formulation in Equation 6 to compute the single-step skip error $\varepsilon_{\text{skip}}(\hat{t}, t)$ between the cached residual $\mathbf{r}_{\hat{t}}$ at timestep \hat{t} and the ground-truth residual \mathbf{r}_t at timestep t . The multiplicative formulation is reasonable according to our following empirical observation and ablation experiments.

Empirical Observation. We first define the ground-truth magnitude ratio between the residual \mathbf{r}_t and $\mathbf{r}_{\hat{t}}$ as $\Gamma(t, \hat{t})$. According to our empirical observation in Figure 5, the magnitude ratio $\Gamma(t, \hat{t})$ can be approximated by the product $\prod_{i=\hat{t}+1}^t \gamma_i$, i.e.:

$$\Gamma(t, \hat{t}) = \text{mean}\left(\frac{\|\mathbf{r}_t\|_2}{\|\mathbf{r}_{\hat{t}}\|_2}\right) \approx \prod_{i=\hat{t}+1}^t \gamma_i = \prod_{i=\hat{t}+1}^t \text{mean}\left(\frac{\|\mathbf{r}_i\|_2}{\|\mathbf{r}_{i-1}\|_2}\right). \quad (14)$$

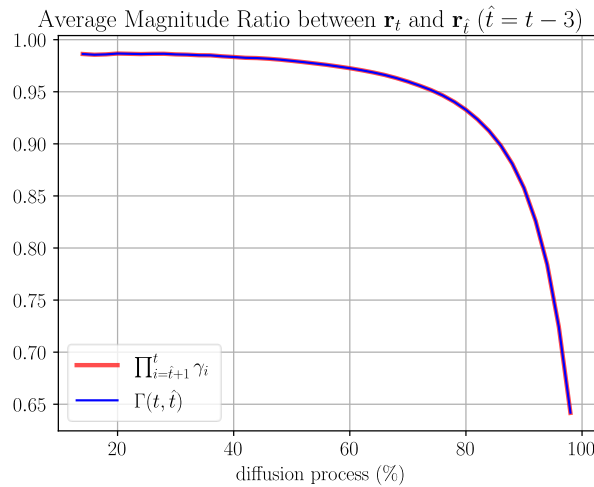


Figure 5: Average Magnitude Ratio between \mathbf{r}_t and $\mathbf{r}_{\hat{t}}$, where $\hat{t} = t - 3$. The $\Gamma(t, \hat{t})$ is the ground-truth magnitude ratio, while the $\prod_{i=\hat{t}+1}^t \gamma_i$ is the predicted magnitude ratio using the multiplicative formulation in Equation 6.

Table 6: Different error modeling methods of single-step skip error $\varepsilon_{\text{skip}}$ on Wan 2.1. Our multiplicative formulation in Equation 6 performs better than the naive baseline in Equation 15.

Error Modeling	Latency (s) ↓	LPIPS ↓	SSIM ↑	PSNR ↑
Wan 2.1	187	-	-	-
Multiplicative Equation 6	87	0.1053	0.8275	24.32
Naive Equation 15	84	0.1154	0.8137	24.06

Besides, the difference between $\Gamma(t, \hat{t})$ and $\prod_{i=\hat{t}+1}^t \gamma_i$ is less than 1e-5 in value. Therefore, the multiplicative formulation in Equation 6 accurately captures the ground-truth magnitude ratio and thus serves as a reliable surrogate.

Ablation Experiments. As a naive baseline, we consider a simplified error modeling method that ignores the accumulated error from previously skipped timesteps and considers only the instantaneous magnitude ratio γ_t at timestep t . The corresponding skip error is defined as:

$$\varepsilon_{\text{skip}}(\hat{t}, t) = 1 - \gamma_t. \quad (15)$$

As shown in Table 6, our multiplicative formulation (Equation 6) consistently outperforms the naive baseline (Equation 15) across all evaluation metrics. This result aligns with our empirical observation that the multiplicative product $\prod_{i=\hat{t}+1}^t \gamma_i$ provides an accurate approximation of the ground-truth magnitude ratio $\Gamma(t, \hat{t})$ between residuals \mathbf{r}_t and $\mathbf{r}_{\hat{t}}$.

It is also worth noting that when the magnitude ratio exceeds 1.0, we take the absolute value of the skip error, as is done in models like HunyuanVideo and Flux.

A.4.2 The Influence of the Initial Steps.

In this section, we investigate the impact of preserving different numbers of initial steps during inference. As shown in Table 7, the first 10 steps are crucial to the overall quality of the generated video. Reducing the number of unchanged initial steps from 10 to 5 leads to a significant degradation in video quality, with LPIPS increasing from 0.1053 to 0.2431, SSIM dropping from 0.8275 to 0.6423, and PSNR falling from 24.32 to 18.80.

While retaining more steps generally improves video quality, it also increases latency and computational cost. To strike a balance between visual fidelity and efficiency, we adopt a default setting that preserves the first 20% of steps, corresponding to 10 steps for Wan 2.1 and 6 steps for Open-Sora.

Table 7: Ablation study on the number of initial unchanged steps for Wan 2.1. The model Wan 2.1 has 50 inference steps in total. †: Default setting where the first 10 steps (20%) are preserved.

Initial Unchanged Steps	Ratio	Latency (s) ↓	LPIPS ↓	SSIM ↑	PSNR ↑
Wan 2.1 $T = 50$	-	187	-	-	-
5	10%	73	0.2431	0.6423	18.80
10 †	20%	87	0.1053	0.8275	24.32
20	40%	98	0.0664	0.8966	27.71

A.5 More Visualization Cases

In this section, we present additional qualitative results, including both videos and images, to further demonstrate the effectiveness of MagCache. Compared with TeaCache, MagCache consistently achieves superior visual quality while maintaining comparable or lower latency. Specifically, MagCache consistently delivers better alignment with ground-truth content, improved preservation of fine visual details, and enhanced rendering of textual elements, such as clearer and more accurate text generation in both videos and images. The qualitative results span four widely used video generation models and one state-of-the-art image generation model: Wan 2.1 1.3B in Figure 6, Wan 2.1 14B in Figure 7, Open-Sora in Figure 8, HunyuanVideo in Figure 9, and Flux(Image Generation Model) in Figure 10.

Wan 2.1 1.3B

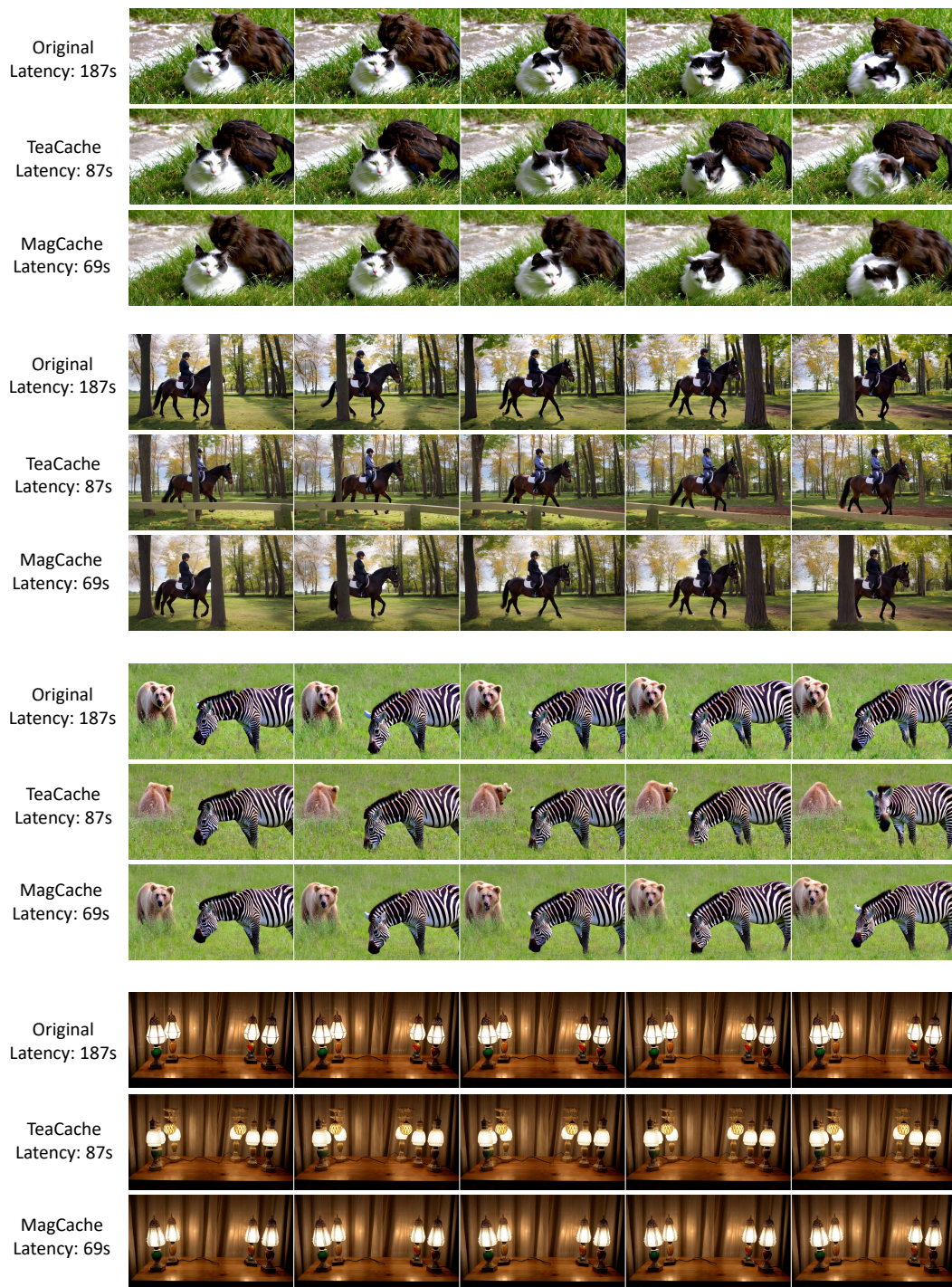
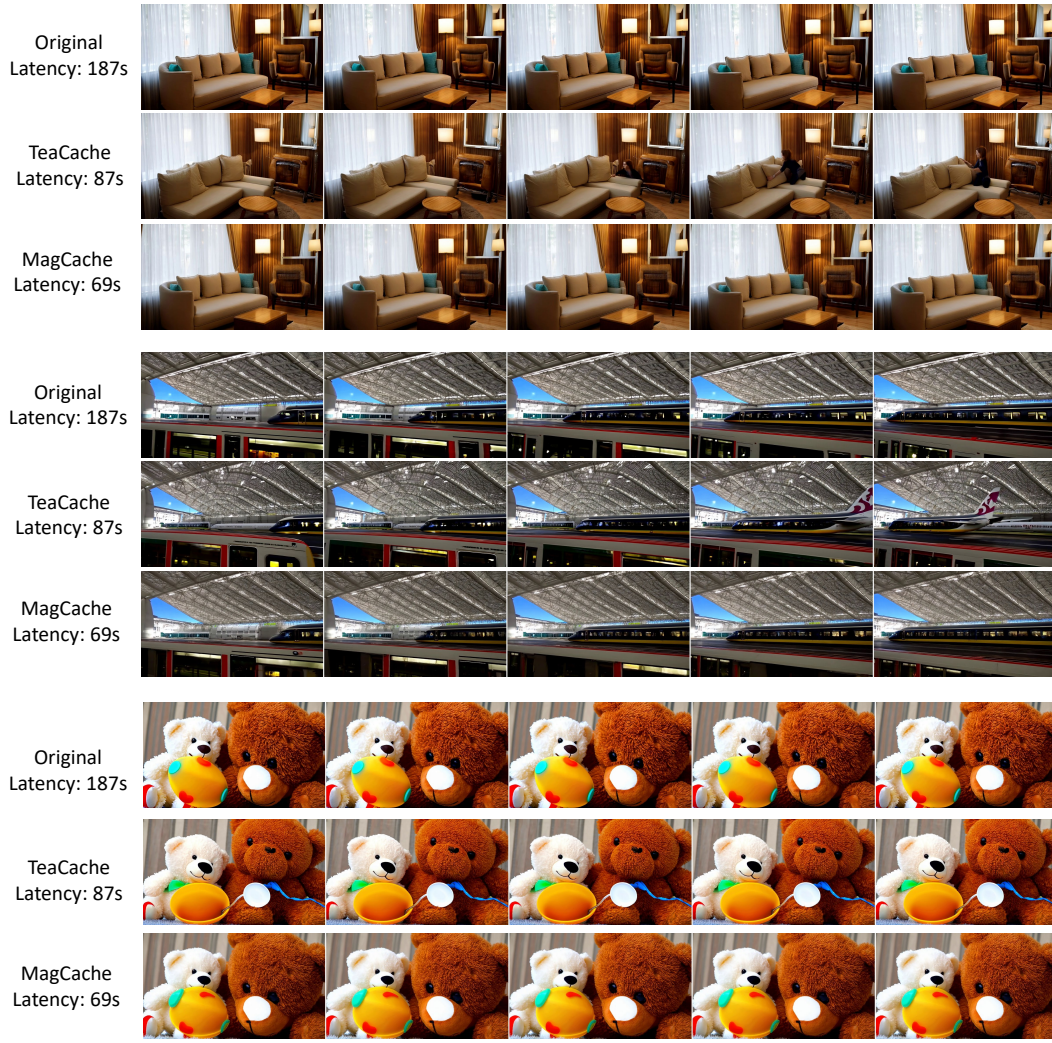


Figure 6: Videos generated by Wan 2.1 1.3B using original model, Teacache-Fast, and our MagCache-Fast. Best-viewed with zoom-in.

Wan 2.1 1.3B



Wan 2.1 14B

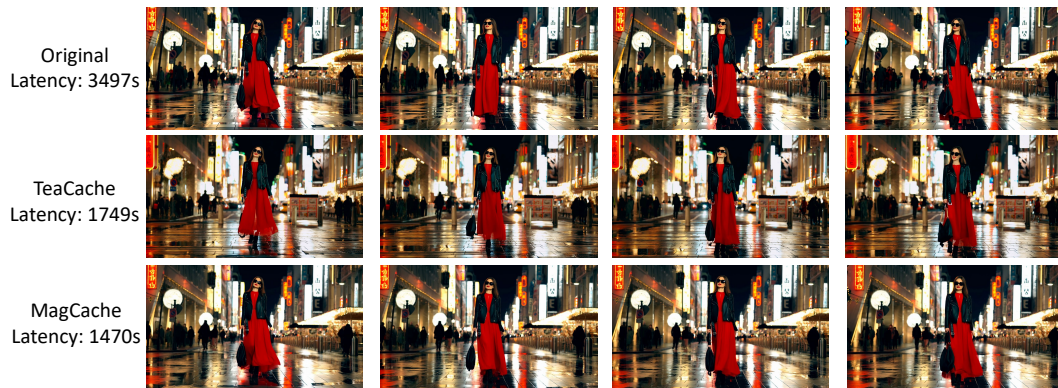


Figure 7: Videos generated by Wan 2.1 1.3B and Wan 2.1 14B using original model, Teacache-Fast, and our MagCache-Fast. Best-viewed with zoom-in.

Open-Sora

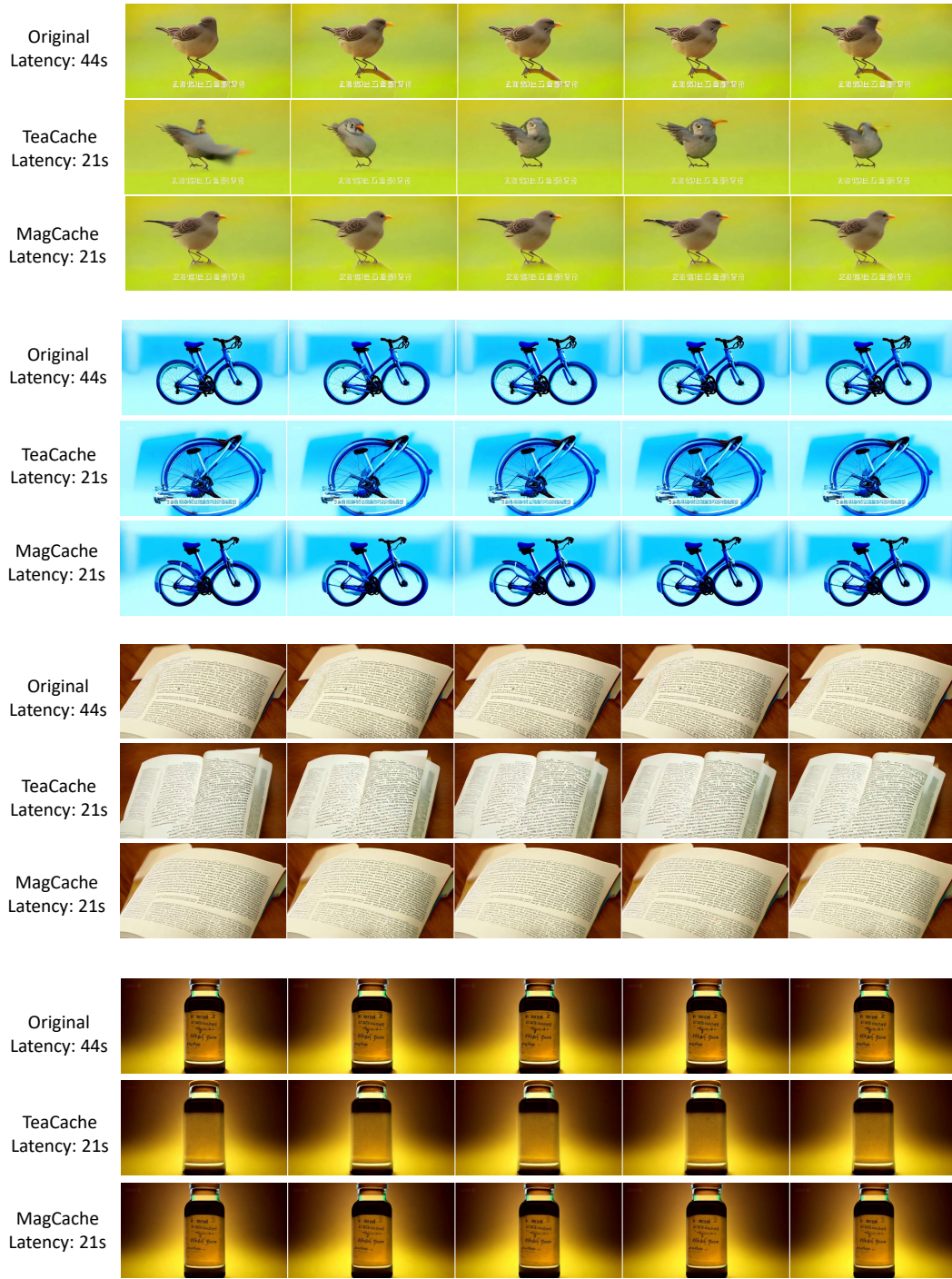


Figure 8: Videos generated by Open-Sora using original model, Teacache-Fast, and our MagCache-Fast. Best-viewed with zoom-in.

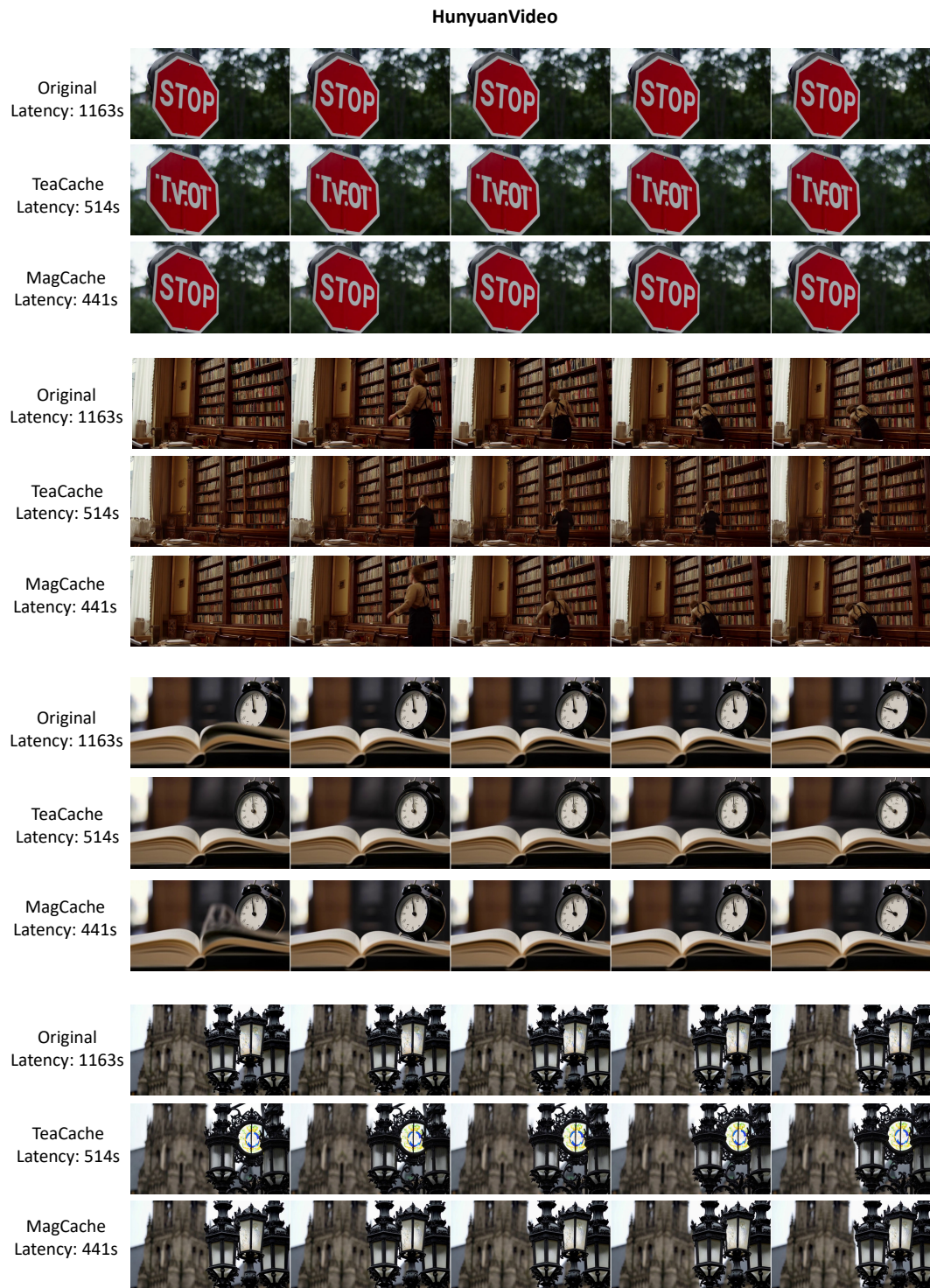


Figure 9: Videos generated by HunyuanVideo using original model, Teacache-Fast, and our MagCache-Fast. Best-viewed with zoom-in.

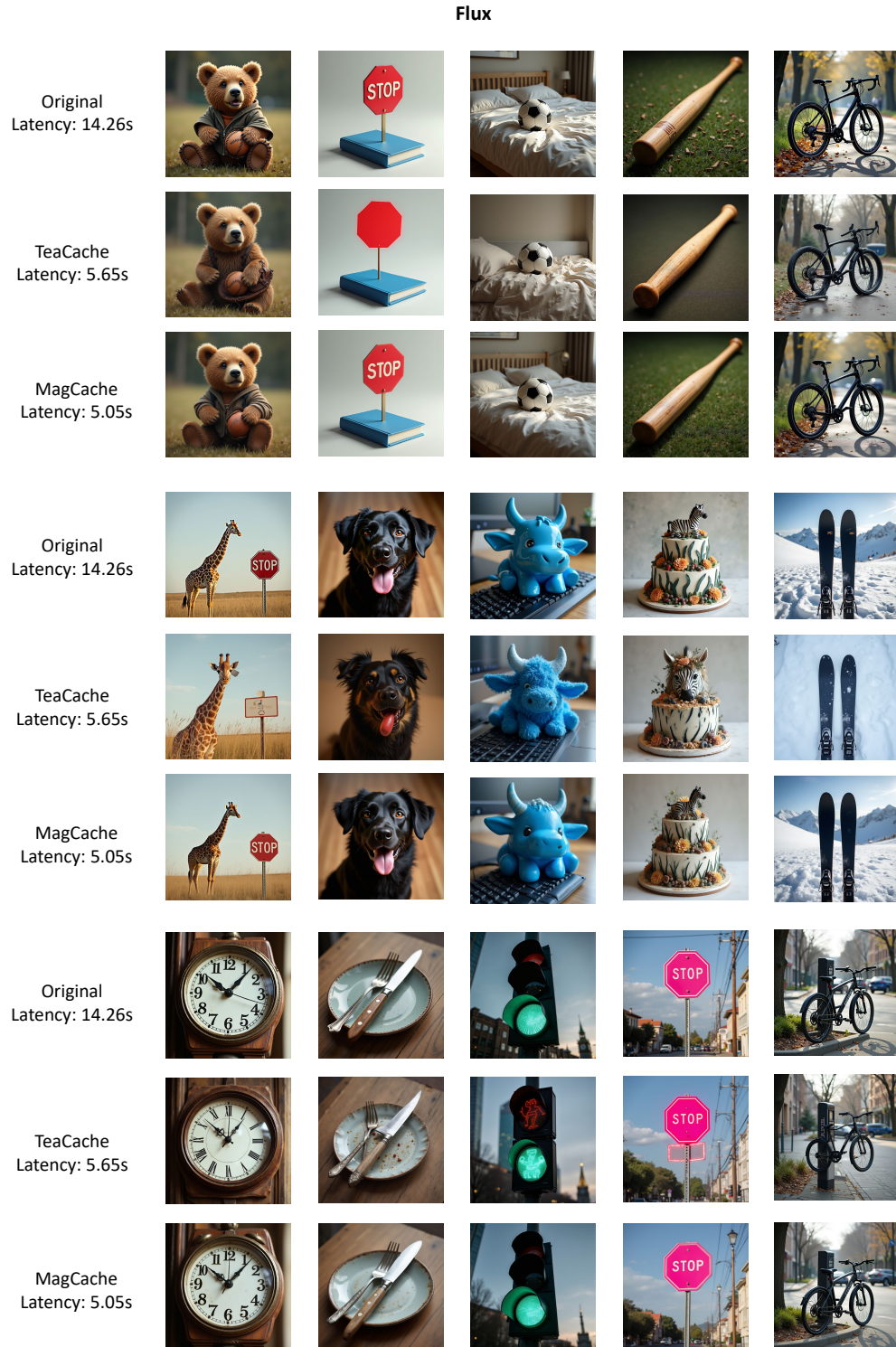


Figure 10: Images generated by Flux using original model, Teacache-Fast, and our MagCache-Fast. Best-viewed with zoom-in.