

# From Reusing to Forecasting: Accelerating Diffusion Models with TaylorSeers

## Supplementary Material

### 6. Experimental Details

In this section, more details of the experiments are provided.

#### 6.1. Model Configuration

As mentioned in 4.1.1, experiments on three models from different tasks, FLUX [18] for text-to-image generation, HunyuanVideo [22, 47] for text-to-video generation, and DiT [32] for class-conditional image generation, are presented. In this section, a more detailed hyperparameter configuration scheme is provided.

- **FLUX:** The FORA [41] method employs a uniform activation interval with  $\mathcal{N}=3$ . The ToCa [64] method uses  $\mathcal{N}=4$  with a caching ratio of 90%, adopting a non-uniform activation interval, with sparse activations at the beginning and dense activations towards the end, utilizing an attention-based token selection method. The DuCa [63] method sets conservative caching steps on even-numbered steps following fresh steps, while aggressive caching steps are set on odd-numbered steps. The activation interval and caching ratio are consistent with the ToCa method, also using a non-uniform activation interval and employing the attention-based token selection method.
- **HunyuanVideo:** The FORA [41] method utilizes an activation interval of  $\mathcal{N}=5$ , whereas both the ToCa [64] and DuCa methods employ  $\mathcal{N}=4$  with a caching ratio of 90%. For each activation step (complete computational step), aggressive caching is applied to odd-numbered steps, and conservative caching is applied to even-numbered steps. Due to memory limitations that result in an "out of memory" error when using a non-uniform activation scheme, the ToCa method in HunyuanVideo is configured with a uniform activation interval, as indicated by a \* in the corresponding table.
- **DiT:** The FORA [41] method uses a uniform activation interval with  $\mathcal{N}=3$ . The ToCa [64] method also uses  $\mathcal{N}=3$  with an average caching ratio of  $R = 95\%$ , employing a non-uniform activation interval, with sparse activations at the beginning and dense activations towards the end, utilizing the attention-based token selection method. The DuCa [63] method sets conservative caching steps on even-numbered steps following fresh steps, while aggressive caching steps are set on odd-numbered steps. The activation interval and caching ratio are consistent with the ToCa method, also using a non-uniform activation interval with sparse-to-dense activation and employing the attention-based token selection method.  $\Delta$ -DiT adopts a layer-skipping strategy

where, in the early stages (49-25 steps), layers 14-27 are skipped, and in the later stages (24-0 steps), layers 0-13 are skipped.

### 7. Supplementary Results for Ablation Studies

We conduct ablation experimental parameter  $\mathcal{N}$  and the Taylor expansion order  $\mathcal{O}$  on computations on DiT-XL/2 [32] to evaluate *TaylorSeer*, focusing on the impact of the interlational efficiency and generation quality. The results demonstrate the importance of these design choices in balancing performance and speed.

Table 4. **Ablation Study with Different Configurations** on ImageNet with DiT-XL/2.

Configuration	FLOPs(T)↓	Speed↑	sFID↓	FID↓
( $\mathcal{N}=3, \mathcal{O}=0$ )	8.56	2.77×	6.36	3.55
( $\mathcal{N}=4, \mathcal{O}=0$ )	6.66	3.56×	8.43	4.75
( $\mathcal{N}=5, \mathcal{O}=0$ )	5.24	4.53×	11.29	6.58
( $\mathcal{N}=6, \mathcal{O}=0$ )	4.76	4.98×	14.84	9.24
( $\mathcal{N}=7, \mathcal{O}=0$ )	3.82	6.22×	18.57	12.67
( $\mathcal{N}=3, \mathcal{O}=1$ )	8.56	2.77×	4.82	2.49
( $\mathcal{N}=4, \mathcal{O}=1$ )	6.66	3.56×	5.54	2.66
( $\mathcal{N}=5, \mathcal{O}=1$ )	5.24	4.53×	6.48	3.05
( $\mathcal{N}=6, \mathcal{O}=1$ )	4.76	4.98×	7.62	3.59
( $\mathcal{N}=7, \mathcal{O}=1$ )	3.82	6.22×	8.76	4.29
( $\mathcal{N}=3, \mathcal{O}=2$ )	8.56	2.77×	4.69	2.44
( $\mathcal{N}=4, \mathcal{O}=2$ )	6.66	3.56×	5.21	2.51
( $\mathcal{N}=5, \mathcal{O}=2$ )	5.24	4.53×	5.87	2.79
( $\mathcal{N}=6, \mathcal{O}=2$ )	4.76	4.98×	6.65	3.18
( $\mathcal{N}=7, \mathcal{O}=2$ )	3.82	6.22×	7.26	3.66
( $\mathcal{N}=3, \mathcal{O}=3$ )	8.56	2.77×	4.69	2.34
( $\mathcal{N}=4, \mathcal{O}=3$ )	6.66	3.56×	5.21	2.53
( $\mathcal{N}=5, \mathcal{O}=3$ )	5.24	4.53×	5.82	2.78
( $\mathcal{N}=6, \mathcal{O}=3$ )	4.76	4.98×	6.50	3.15
( $\mathcal{N}=7, \mathcal{O}=3$ )	3.82	6.22×	7.08	3.63
( $\mathcal{N}=3, \mathcal{O}=4$ )	8.56	2.77×	4.69	2.35
( $\mathcal{N}=4, \mathcal{O}=4$ )	6.66	3.56×	5.19	2.52
( $\mathcal{N}=5, \mathcal{O}=4$ )	5.24	4.53×	5.82	2.78
( $\mathcal{N}=6, \mathcal{O}=4$ )	4.76	4.98×	6.50	3.15
( $\mathcal{N}=7, \mathcal{O}=4$ )	3.82	6.22×	7.07	3.63

### 8. Anonymous Page for Video Presentation

To further showcase the advantages of TaylorSeer in video generation, we have created an anonymous GitHub page.

For a more detailed demonstration, please visit <https://taylorseer.github.io/TaylorSeer/>. Additionally, the videos are also available in the Supplementary Material.

## 9. Supplementary Visualization Examples

To further illustrate the qualitative improvements of our method, we present visualization examples on FLUX and HunyuanVideo. These results showcase the superior fidelity and consistency of our method in generating high-quality outputs across diverse scenarios.

## 10. Supplementary Visualization of Feature Trajectories in Diffusion Models

In this section, we provide additional visualizations of feature trajectories and their derivatives in diffusion models. These results further illustrate the stability and predictability of feature dynamics across different timesteps, supporting our findings in the main text. The PCA projections of features (0th-order) and their derivatives (1st to 4th-order) demonstrate consistent patterns, highlighting the potential for efficient feature prediction in diffusion models.

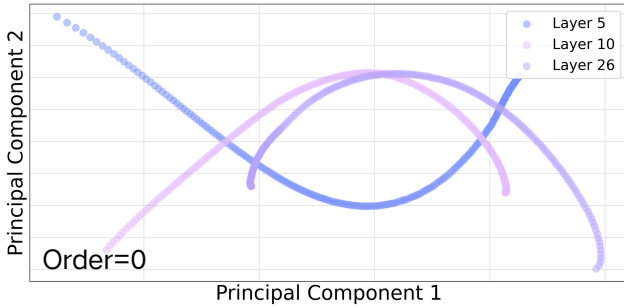


Figure 8. **PCA projections of features in diffusion models.** The features at different timesteps form stable trajectories, demonstrating the predictability of feature evolution over time.

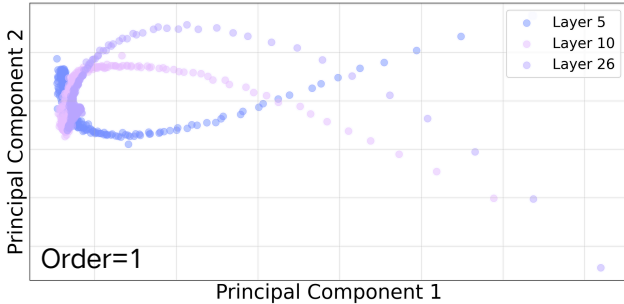


Figure 9. **PCA projections of first-order feature derivatives.** The first-order derivatives exhibit consistent patterns, further supporting the predictability of feature dynamics in diffusion models.

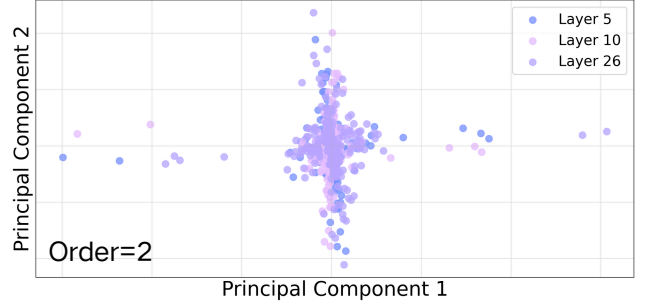


Figure 10. **PCA projections of second-order feature derivatives.** The second-order derivatives reveal higher-order dynamics, highlighting the smoothness of feature transitions.

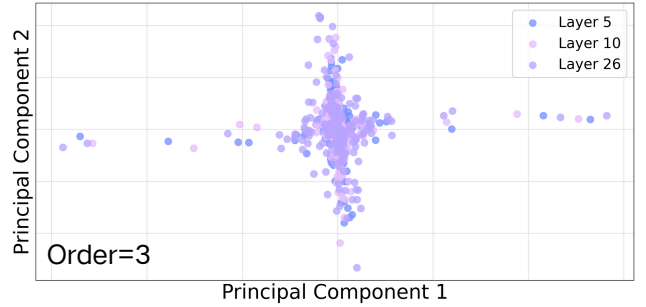


Figure 11. **PCA projections of third-order feature derivatives.** The third-order derivatives capture more complex temporal patterns, indicating the richness of feature dynamics.

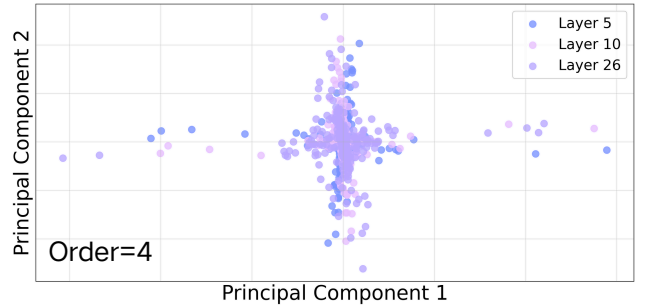


Figure 12. **PCA projections of fourth-order feature derivatives.** The fourth-order derivatives provide insights into fine-grained temporal variations, further validating the predictability of feature evolution.

## 11. More Results for TaylorSeer

To further validate the broad applicability and adaptability of *TaylorSeer*, we conducted comprehensive experiments on several other mainstream text-to-video DiT models, including HiDream [3], FramePack [56], and WAN2.1 [48]. The results demonstrate that *TaylorSeer* consistently achieves superior performance across these platforms.

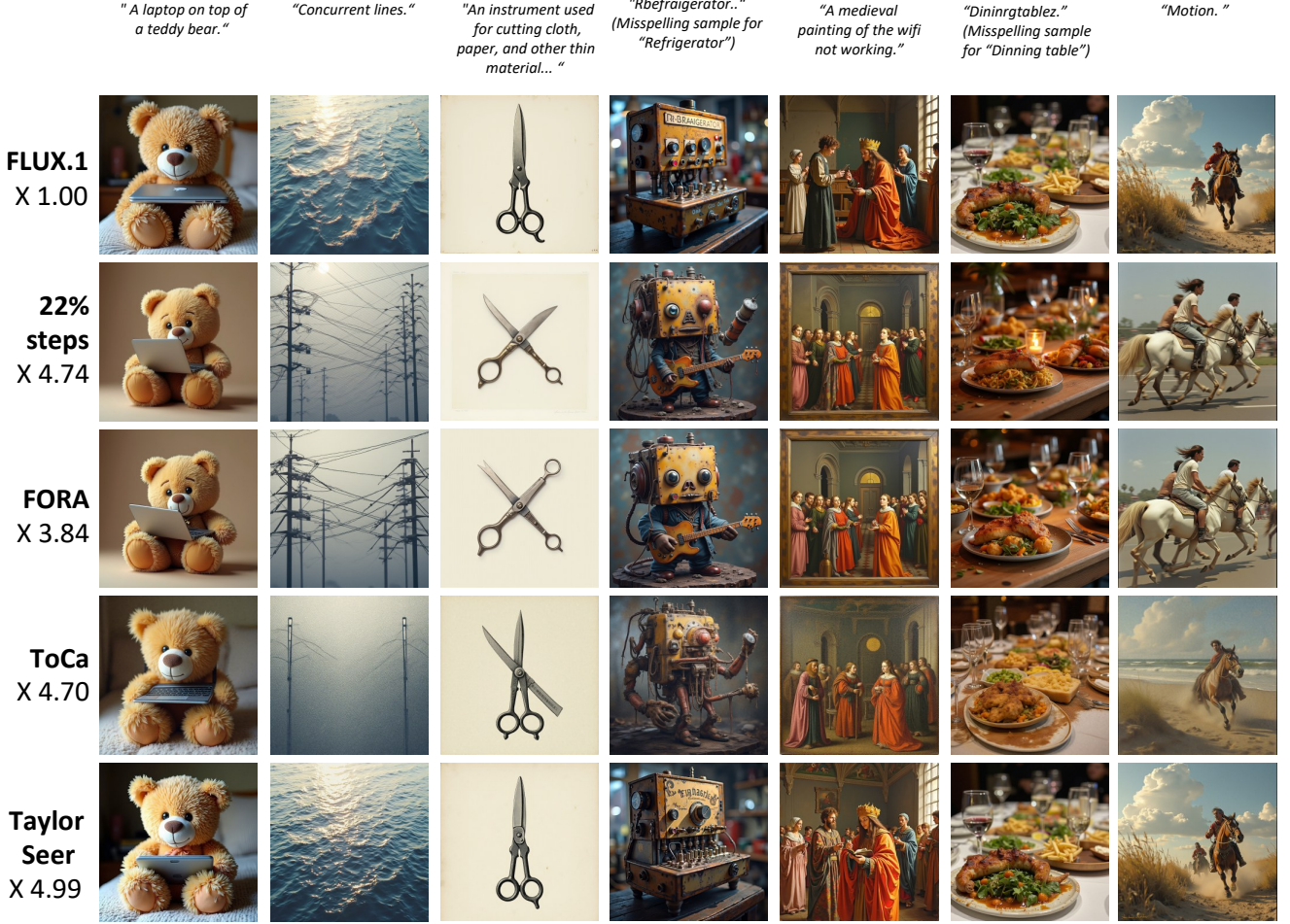


Figure 13. Visualization results for different acceleration methods on FLUX.1-dev.

As a representative example, Table 5 presents a detailed performance comparison on the HiDream model. While achieving a 4.0 *times* acceleration, *TaylorSeer* significantly outperforms the competing method, *TeaCache*, across all key evaluation metrics, including PSNR, SSIM, LPIPS, and ImageReward.

Furthermore, our project page showcases detailed comparison videos between *TaylorSeer* and competing methods on the HunyuanVideo model, which visually demonstrate the significant advantages of our approach. Moreover, *TaylorSeer* exhibits remarkable versatility by delivering unexpectedly impressive results when extended to *super-resolution tasks* and audio generation.

Table 5. Performance comparison on HiDream.

Method HiDream	Acceleration		Image Reward↑	PSNR↑	SSIM↑	LPIPS↓
	TFLOPs↓	Speed↑				
HiDream-Full	7780.0	1.0×	1.1285	-	-	-
TeaCache( $l_1 = 1$ )	2047.4	3.8×	0.9849	28.139	0.6036	0.565
<i>TaylorSeer</i> ( $N = 4, O = 2$ )	<b>1945.0</b>	<b>4.0×</b>	<b>1.0833</b>	<b>28.248</b>	<b>0.6084</b>	<b>0.532</b>

## 12. Performance Analysis at Low Speedup Ratios

Our study reveals that *TaylorSeer* not only accelerates the inference process but can also surpass the baseline performance, particularly at reduced acceleration ratios. As detailed in Table 1 and Table 2, our empirical results substantiate this finding. Specifically, when applied to FLUX, the ImageReward score improved from 0.9898 to 1.0181. Similarly, for HunyuanVideo, the VBench score increased from 80.66 to 80.74. These results validate the effectiveness of *TaylorSeer* across the full spectrum of acceleration factors.

We attribute this counter-intuitive performance enhancement to the inherent parameter redundancy within the original large-scale models. This observation is analogous to established techniques such as model pruning and Low-Rank Adaptation (LoRA), where strategically reducing parameter utilization can sometimes lead to improved generalization and overall performance. We hypothesize that *TaylorSeer* introduces a regularization-like effect during the in-



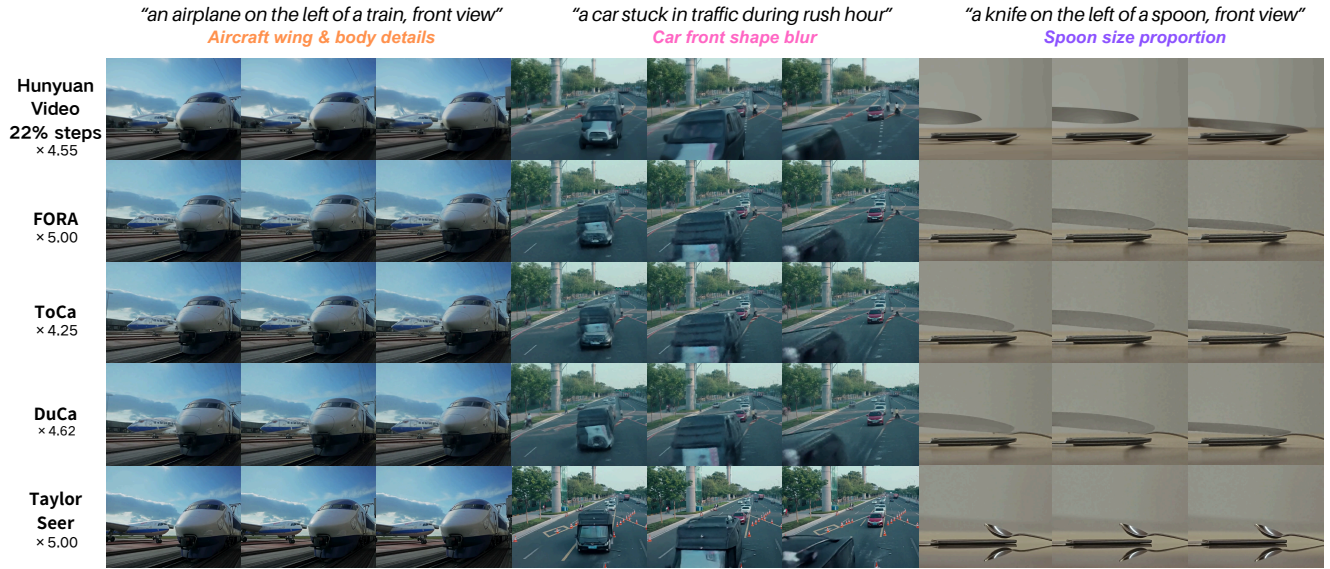


Figure 14. Visualization results for different acceleration methods on HunyuanVideo.

ference steps. Since both FLUX and HunyuanVideo are based on Flow Matching, the Taylor approximation-based simplification at each step may guide the generation process along a more robust trajectory, effectively regularizing the model’s output. This finding provides valuable insights for both model acceleration and future training methodologies, suggesting that targeted inference-time optimization can unlock latent performance gains.