Beyond Fixed Formulas: Data-Driven Linear Predictor for Efficient Diffusion Models

Supplementary Material

A. Experiment Settings and Evaluation

Implementation Details and Baselines. We evaluate our proposed method across three advanced diffusion transformer architectures: FLUX.1-dev [2] and Qwen Image [7] for text-to-image generation, and Hunyuan Video [1] for text-to-video tasks. The core learnable predictor was trained for 200 epochs with a learning rate of 0.01, utilizing a dataset of 50 prompts generated by an LLM. For comparative analysis, we benchmark against standard 50-step sampling baselines across all three models. Specifically, we employ the Rectified Flow sampler for FLUX.1-dev and standard samplers for Qwen Image and Hunyuan Video. Additionally, we include the Qwen-Image-Lightning-8steps [7] to assess performance against accelerated distinct baselines. Activate steps were slightly adjusted via prediction residuals as hyperparameters. We further compare our approach with state-of-the-art feature caching and prediction methods using their official implementations to ensure fair comparisons.

Evaluation Metrics. Our evaluation framework comprehensively considers two core dimensions: acceleration efficiency and generation fidelity. For efficiency, we measure wall-clock inference latency and FLOPs to quantify computational load. We also report the speedup ratio relative to the 50-step baselines to demonstrate performance improvements. For fidelity, we assess the consistency between the accelerated outputs and the original 50-step generations. We quantify quality across all image and video results using three standard perceptual metrics: PSNR, SSIM [6], and LPIPS [8].

B. Memory Overhead in Practice

As described in **Sec. 3.3.2, line 352**, L^2P performs prediction using **only the *final-layer* features**. For HunyuanVideo at 480×640 resolution with 65 frames, the cached feature tensor is $(1, 20400, 64)$, incurring at most $\sim$**0.49 GB** cache memory for 50 steps, using $\sim$**38.6**$\times$ less VRAM than the window-based baseline, first-order TaylorSeer. The results on FLUX.1-dev are summarized in Table 1.

Table 1. GPU memory usage comparison (MiB).

Method	w/o Cache	ToCa	TaylorSeer ($\mathcal{O}$=2)	TaylorSeer ($\mathcal{O}$=1)	**Ours**
Memory (MiB)	39,653	49,349	45,473	43,421	**39,673**

C. More Results on T2V Generation

Table 2 reports the comprehensive performance on the HunyuanVideo benchmark. Our method demonstrates a dominant advantage across both computational efficiency and generation fidelity.

In terms of efficiency, our method (with $\mathcal{N} = 7$) achieves the lowest inference latency of 20.95s, corresponding to a 4.72$\times$ wall-clock speedup. This significantly outperforms all competing baselines, including the recent state-of-the-art TeaCache (21.83s, 4.53$\times$) and predictive methods like TaylorSeer (23.66s, 4.18$\times$). Furthermore, our method requires the lowest computational cost (5359.2T FLOPs), validating that our lightweight linear predictor introduces negligible overhead while maximizing skipped steps.

In terms of fidelity, the superiority of our approach is even more pronounced. Existing acceleration methods typically struggle with the complex temporal dynamics of video, resulting in significant quality degradation (e.g., most baselines hover around 17.0 dB in PSNR). In contrast, our method maintains exceptional consistency with the original 50-step generation, achieving a PSNR of 21.10 dB. This represents a remarkable improvement of +2.85 dB over the second-best method, TeaCache. Similarly, we achieve the highest SSIM (0.72) and lowest LPIPS (0.28), proving that our data-driven predictor effectively mitigates the error accumulation that plagues fixed-coefficient predictors like TaylorSeer in long-trajectory video generation.

D. Justification for Linear Design

A natural question arises: ***Why restrict the predictor to a linear combination? Can introducing non-linearity improve performance?*** We categorize potential non-linear enhancements into two paradigms: **(1) explicit a priori non-linear modeling**, and **(2) implicit data-driven non-linear learning.**

For the first paradigm, we conducted extensive experiments incorporating explicit non-linear priors. Our explorations included:

- **Spatially-varying coefficients**: Assigning distinct linear weights to different spatial tokens rather than sharing a global scalar.
- **Non-linear transforms**: Introducing terms such as quadratic (x^2), exponential (exp), logarithmic (log), and radical ($\sqrt{x}$) components into the expansion.
- **Attention mechanisms**: Applying lightweight self-

Table 2. **Quantitative comparison of text-to-video generation** on HunyuanVideo.

Method HunyuanVideo	Acceleration				Perceptual Metrics		
	Latency(s) ↓	Speed ↑	FLOPs(T) ↓	Speed ↑	PSNR↑	SSIM↑	LPIPS↓
Original: 50 steps	98.91	1.00×	29773.0	1.00×	∞	1.00	0.00
22% **steps**	22.98	4.30×	6550.1	4.55×	17.65	0.59	0.42
ToCa [9]($\mathcal{N}=5, \mathcal{R}=90\%$)	26.12	3.79×	7006.2	4.25×	17.04	0.54	0.44
DuCa [10]($\mathcal{N}=5, \mathcal{R}=90\%$)	23.08	4.29×	6483.2	4.48×	17.08	0.54	0.43
TeaCache [3]($l=0.4$)	21.83	4.53×	6550.1	4.55×	18.25	0.61	0.38
FORA [5]($N=5$)	22.61	4.37×	5960.4	5.00×	17.00	0.53	0.44
TaylorSeer [4]($\mathcal{N}=5, O=1$)	23.66	4.18×	5960.4	5.00×	17.29	0.55	0.42
Ours($\mathcal{N}=7$)	**20.95**	**4.72**×	**5359.2**	**5.55**×	**21.10**	**0.72**	**0.28**

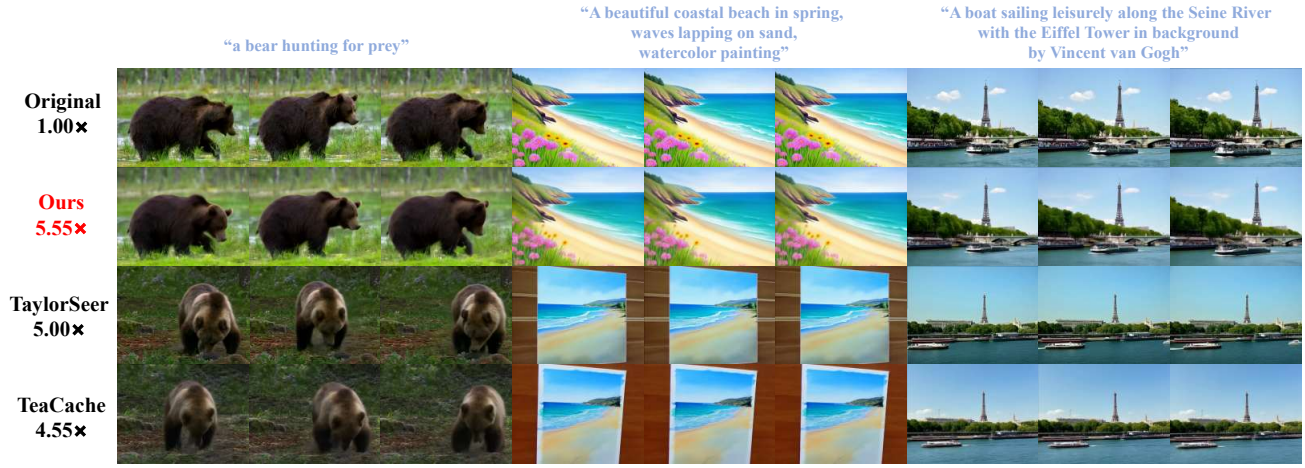

Figure 1. **Qualitative Comparison of Early, Middle, and Late Frames From Generated Videos.** On HunyuanVideo, L^2P constructs continuous dynamic process with excellent subject accuracy under high speedup ratio.

attention to capture inter-token dependencies.

However, none of these variants yielded significant performance gains. Theoretically, the underlying Stochastic Differential Equation (SDE) governing the diffusion process does not exhibit an obvious, analytically consistent prior that favors specific non-linear functional forms for feature temporal evolution.

For the second paradigm, employing deep neural networks (e.g., MLPs or LSTMs) to implicitly learn non-linear dynamics presents two critical drawbacks. First, it requires massive datasets to generalize, whereas our method works with minimal data (50 prompts). Second, and more importantly, the computational overhead of running a neural network predictor contradicts the fundamental goal of inference acceleration.

Conclusion. As demonstrated in **Obs. 2**, the linear subspace already captures over 95% of the feature reconstruction fidelity for the vast majority of steps. Given that the linear framework achieves near-optimal fidelity with minimal cost, we adopt the strictly linear formulation as the optimal trade-off between efficiency and precision.

E. Linear Combinations Seen Through High-Order Expansions

We demonstrated in Obs. 1 that existing difference-based prediction methods are equivalentd to linear combinations. Conversely, it can be proven that any linear combination of historical features can be equivalently converted into a sum of high-order differences. Therefore, our explicitly learned predictor is mathematically isomorphic to an implicit high-order difference expansion:

$$\sum_{j=0}^{t-1} W_{t,j} \cdot \mathcal{F}(x_j) \equiv \sum_{k=0}^{t-1} \omega_k^* \cdot \Delta^k \mathcal{F}(x_{t-1}) \tag{1}$$

where Δ^k denotes the k-th order backward difference operator. Since the discrete difference $\Delta^k \mathcal{F}$ serves as an approximation of the k-th order derivative $\mathcal{F}^{(k)}$, Eq. 1 can be regarded as a generalized Taylor-like expansion. While traditional methods (e.g., Taylorseer) rely on pre-defined, fixed

coefficients derived from generic approximation formulas, our method learns an optimal set of coefficients tailored to the specific feature evolution dynamics. This data-driven approach allows for capturing complex temporal dependencies more accurately than rigid analytical expansions. We provide a formal proof of this equivalence below.

Proof of Equivalence. We verify that for any set of linear weights $\{W_{t,j}\}$, there exists a unique corresponding set of difference coefficients $\{\omega_k^*\}$. First, we define the vector of historical feature values $\mathbf{f}$ and the vector of high-order backward differences $\mathbf{d}$ at step $t-1$ as:

$$\mathbf{f} = [\mathcal{F}(x_{t-1}), \dots, \mathcal{F}(x_0)]^\top \in \mathbb{R}^t \tag{2}$$

$$\mathbf{d} = [\Delta^0\mathcal{F}(x_{t-1}), \dots, \Delta^{t-1}\mathcal{F}(x_{t-1})]^\top \in \mathbb{R}^t \tag{3}$$

By definition, the k-th order backward difference is a linear combination of historical values determined by binomial coefficients:

$$\Delta^k\mathcal{F}(x_{t-1}) = \sum_{m=0}^{k} (-1)^m \binom{k}{m} \mathcal{F}(x_{t-1-m}) \tag{4}$$

This relationship allows us to express the transformation in matrix form:

$$\mathbf{d} = \mathbf{P}\mathbf{f} \tag{5}$$

where $\mathbf{P} \in \mathbb{R}^{t\times t}$ is a lower triangular Pascal matrix with alternating signs. Its entries are given by $P_{k,m} = (-1)^m\binom{k}{m}$ for $k \geq m$, and 0 otherwise.

Crucially, the diagonal elements of $\mathbf{P}$ are non-zero ($P_{k,k} = (-1)^k$), which implies that the determinant is non-zero:

$$\det(\mathbf{P}) = \prod_{k=0}^{t-1} P_{k,k} \neq 0 \tag{6}$$

Thus, $\mathbf{P}$ is invertible, ensuring a bijective mapping $\mathbf{f} = \mathbf{P}^{-1}\mathbf{d}$. Substituting this into the linear predictor form (LHS of Eq. 1):

$$\text{LHS} = \mathbf{W}^\top\mathbf{f} = \mathbf{W}^\top(\mathbf{P}^{-1}\mathbf{d}) = (\mathbf{W}^\top\mathbf{P}^{-1})\mathbf{d} \tag{7}$$

By defining the new coefficient vector via the transformation:

$$(\boldsymbol{\omega}^*)^\top = \mathbf{W}^\top\mathbf{P}^{-1} \tag{8}$$

we arrive at the high-order expansion form $\sum \omega_k^*\Delta^k\mathcal{F}(x_{t-1})$. This concludes the proof.

References

[1] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. 1

[2] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1

[3] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7353–7363, 2025. 2

[4] Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15853–15863, 2025. 2

[5] Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration, 2024. 2

[6] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 1

[7] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Da-Wei Liu, De mei Li, Hang Zhang, Hao Meng, Hu Wei, Ji-Li Ni, Kai Chen, Kuang Cao, Liang Peng, Lin Qu, Min Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiao-Xue Xu, Yi Wang, Yichang Zhang, Yong-An Zhu, Yujian Wu, Yu-Jiao Cai, and Ze-Yang Liu. Qwen-image technical report. *ArXiv*, abs/2508.02324, 2025. 1

[8] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1

[9] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching, 2025. 2

[10] Chang Zou, Evelyn Zhang, Runlin Guo, Haohang Xu, Conghui He, Xuming Hu, and Linfeng Zhang. Rethinking token-wise feature caching: Accelerating diffusion transformers with dual feature caching, 2025. 2