# SPECTRALCACHE: ACCELERATING DIFFUSION-BASED WORLD MODELS VIA SPECTRAL FEATURE CACHING

**Anonymous authors**
Paper under double-blind review

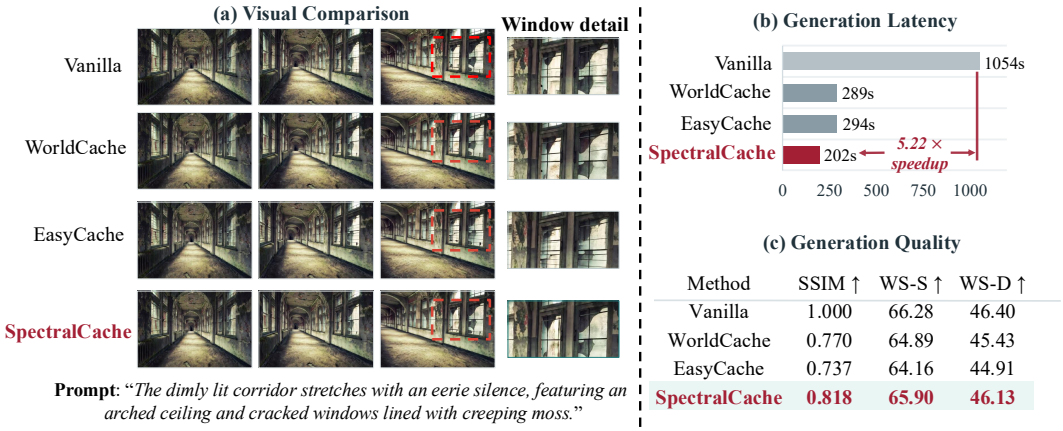



Figure 1: (a) Visual comparison, (b) latency comparison, and (c) quality comparison of SpectralCache with baselines on HunyuanWorld-Voyager-13B. **SpectralCache achieves about** $5.22\times$ **acceleration while preserving high-fidelity video quality**.

## ABSTRACT

Diffusion-based world models enable high-quality interactive environment generation but suffer from substantial inference overhead due to repeated Transformer evaluations during denoising. Existing caching methods mainly exploit temporal redundancy at the feature or token level, leaving the underlying mathematical structure of diffusion features largely unexplored. In this work, we reveal that world-model features exhibit highly stable singular subspaces across nearby denoising steps, while their singular values follow predictable evolution patterns. Building on this observation, we propose **SpectralCache**, a training-free spectral caching framework that reuses stable singular subspaces and estimates only low-dimensional singular values through linear extrapolation. We further exploit the spectral consistency between neighboring full-computation features to skip selected expensive backbone evaluations via singular value scaling. Extensive experiments on representative world models demonstrate that SpectralCache consistently improves inference efficiency while preserving generation quality. On HunyuanWorld-Voyager-13B, SpectralCache achieves $5.22\times$ acceleration while maintaining a WorldScore of 65.90 for static scenes, substantially outperforming existing training-free caching methods in inference efficiency.

## 1 INTRODUCTION

Diffusion-based world models enable controllable visual simulation by predicting future observations conditioned on scenes and actions Bar et al. (2025); Zhu et al. (2025). These capabilities support applications ranging from visual planning to explorable 3D scene generation Bar et al. (2025); Huang et al. (2025). However, their iterative generation process requires repeated evaluations of large diffusion transformers (DiTs), incurring substantial computational cost and limiting responsiveness in interactive applications Song et al. (2025); Feng et al. (2026b). Accelerating world-model inference while preserving generation quality is therefore a key challenge for practical deployment.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A wide range of diffusion acceleration methods have been proposed to improve efficiency Feng et al. (2025b); Xi et al. (2025); Feng et al. (2025a); Zhang et al. (2025), including refining ODE trajectories Lu et al. (2022); Song et al. (2020), model pruning Zhang et al. (2024); Guo et al. (2025), and quantization Shang et al. (2023); Wu et al. (2025). Among them, caching methods based on token structural characteristics have attracted increasing attention for diffusion acceleration Chen et al. (2026); Selvaraju et al. (2024), due to their training-free nature and ability to preserve superior generation quality by reusing or predicting intermediate representations.

However, most existing caching methods target conventional image and video diffusion models. They do not readily generalize to diffusion-based world models and direct adaptation often results in substantial quality degradation Song et al. (2025). Although recent studies have explored accelerating world model generation with caching methods Song et al. (2025); Feng et al. (2026b), they mainly focus on identifying world models' differences from conventional diffusion models and exploiting how step-wise intermediate feature representations change for acceleration. Different from these works, in this paper, we explore the evolvement of underlying mathematical structure and parameter-level characteristics throughout generation, by using singular value decomposition (SVD).

We perform SVD on the intermediate DiT features throughout world model generation and analyze how their underlying subspaces and singular values evolve across generation steps. Our insightful analysis reveals that: (1) Singular subspaces remain highly stable with similar singular vectors across consecutive steps. (2) Singular values of neighboring steps exhibit linear property, which enables accurate estimation of singular values through linear extrapolation or scaling.

Building on these insights, we propose **SpectralCache**, a training-free caching framework for diffusion-based world models that efficiently accelerates video scene generation while preserving high generation quality. Specifically, across consecutive denoising steps, we approximate subsequent features by interpolating singular values within the singular subspace of the preceding fully computed feature, thereby skipping redundant full computations. When a full computation step is required due to large enough accumulation error Feng et al. (2026b), instead of always using full computations, we alternatively adopt full computation and approximation of singular values through scaling, yielding additional acceleration. Our comprehensive evaluation demonstrates that SpectralCache achieves superior speedup with non-marginal improvements than SOTA world-model acceleration methods Feng et al. (2026b); Song et al. (2025). Meanwhile, it leads to best generation quality compared with baselines due to highly accurate approximation. Our contributions are summarized as follows:

- We propose **SpectralCache**, a training-free spectral feature caching framework for accelerating diffusion-based world models. SpectralCache reuses stable singular subspaces and estimates only low rank singular values through linear extrapolation for cache and scaling for skipping more full computation, substantially reducing expensive overhead while preserving generation quality.
- We reveal several interesting spectral structures in world-model features: they exhibit strong low-rank structure, while their dominant singular subspaces remain highly stable across both consecutive denoising steps and nearby cache-triggered full computation steps. Moreover, their singular values exhibit predictable linear extrapolation and approximately constant scaling patterns in these two settings, respectively, providing a new spectral perspective for understanding and exploiting temporal redundancy in diffusion-based world models.
- Across two representative world models, HunyuanWorld-Voyager-13B and Aether-5B, SpectralCache consistently outperforms existing training-free caching methods in both generation quality and inference efficiency. On HunyuanWorld-Voyager-13B, SpectralCache achieves a 5.22$\times$ speedup while maintaining WorldScores of 65.90 and 46.13 for static and dynamic scenes, respectively, compared with 3.65$\times$ speedup and WorldScores of 64.89 and 45.43 achieved by WorldCache.

## 2 Related Works

**Generation-driven Diffusion World Models.** World models seek to learn predictive representations of the environment that support future-state anticipation, planning, and decision-making Hafner et al. (2019; 2023). Policy-centric world models Hafner et al. (2019; 2023); Robine et al. (2023); Wu et al. (2023) acquire environmental dynamics through agent-environment interaction, where

action-conditioned observations and feedback are progressively incorporated into an internal model for subsequent planning and control. In contrast, generative world models Agarwal et al. (2025); Bar et al. (2025); Russell et al. (2025); Zhu et al. (2025) learn environmental dynamics from large-scale visual or video data and simulate plausible future scene evolution with diffusion method.

**Model-level Diffusion Acceleration.** Despite their strong generative capabilities Yang et al. (2023); Croitoru et al. (2023), diffusion models still incur substantial inference costs, motivating various acceleration strategies. Optimization-based methods primarily modify the sampling process without altering the underlying denoising network, using deterministic sampling or advanced high-order solvers to reduce the number or cost of denoising steps Guo et al. (2024); Lu et al. (2022); Song et al. (2020). Recent studies further optimize ODE trajectories through distillation Luo et al. (2024); Yan et al. (2024); Yin et al. (2024), enabling generation with substantially fewer sampling steps, but often requiring additional optimization or training Salimans & Ho (2022). In contrast, model-structural acceleration methods directly reduces the computational complexity of the denoising network itself, such as compressing model parameters through quantization Shang et al. (2023); Wu et al. (2025) or removing redundant structures through pruning Zhang et al. (2024); Guo et al. (2025).

**Diffusion Acceleration from Representation Cache.** Caching has recently emerged as an effective training-free approach for accelerating diffusion inference while largely preserving generation quality Kahatapitiya et al. (2025); Ma et al. (2024). Adaptive token caching selectively reuses tokens based on their importance or variation while recomputing the remaining tokens Zou et al. (2024); Zheng et al. (2025). Forecasting-based methods instead predict the evolution of intermediate representations to skip subsequent full computations across denoising steps Liu et al. (2025b); Chen et al. (2026). However, most existing caching methods are designed for conventional image or video generation and often suffer from quality degradation when directly applied to world models with complex hierarchical feature dynamics Song et al. (2025).

## 3 Deep Insights with Singular Value Decomposition

**Low-rank spectral state of visual features.** We adopt SVD to analyze visual features. The $t^{th}$ denoising step visual feature $\mathbf{F}_t \in \mathbb{R}^{f\times h\times w\times D}$ from final-layer of DiT backbone (with $f$ frames, height $h$, and width $w$) can be transformed to $\mathbf{X}_t = \mathrm{mat}(\mathbf{F}_t) \in \mathbb{R}^{M\times D}$, where $M = f \times h \times w$ denotes the number of visual tokens and $D$ is the channel dimension. We can perform SVD on $\mathbf{X}_t$ as

$$\mathbf{X}_t = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^\top + \mathbf{R}_t, \tag{1}$$

where $\mathbf{U}_t \in \mathbb{R}^{M\times r}$ and $\mathbf{V}_t \in \mathbb{R}^{D\times r}$ are the leading singular subspaces, $\mathbf{\Sigma}_t \in \mathbb{R}^{r\times r}$ is diagonal, and $\mathbf{R}_t$ is the truncated residual. We call $\mathcal{Q}_t = \{\mathbf{U}_t, \mathbf{\Sigma}_t, \mathbf{V}_t, \mathbf{R}_t\}$ the spectral state at $t$-th step.

Our key observations with SVD provides deep insights for features during diffusion. Specifically, we perform SVD as $\mathbf{X}_t = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^\top$ for the visual feature $\mathbf{F}_t$ generated at the $t^{th}$ denoising step. By carefully analyzing the singular vectors and values, we can make the following key observations.

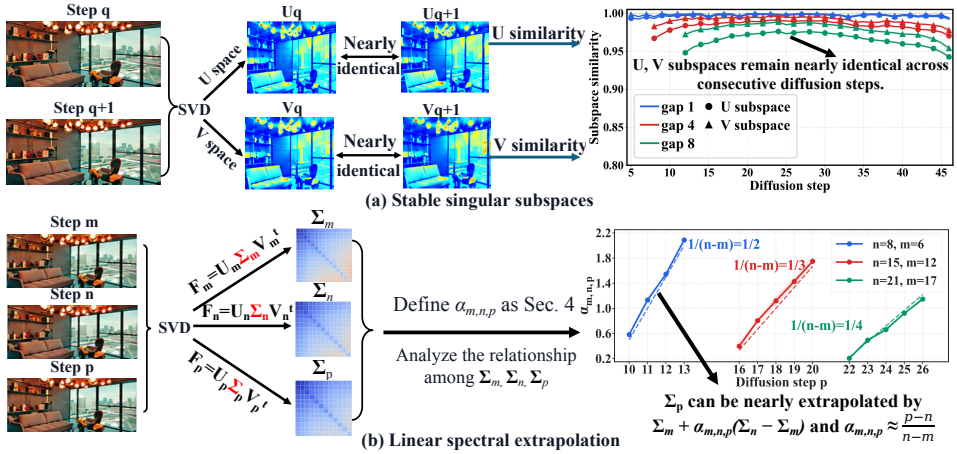
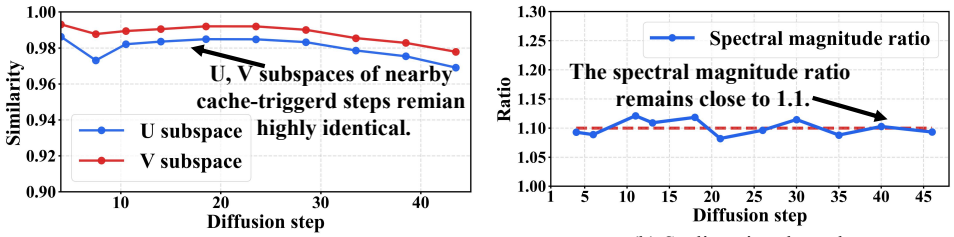



Figure 2: Spectral dynamics across denoising steps in HunyuanWorld-Voyager. (a) The dominant singular subspaces remain highly stable across consecutive denoising steps. (b) Singular values exhibit a consistent linear extrapolation pattern across denoising steps.




(a) Identical subspace (b) Scaling singular values

Figure 3: Spectral consistency across nearby full-computation steps. (a) The dominant singular subspaces remain highly consistent. (b) The spectral magnitude ratio remains approximately constant.

**Denoising features are low-rank.** We first observe that visual features exhibit a strong low-rank structure, with the top 128 singular values capturing 96% of the total feature energy on average.

**Singular subspaces remain highly stable with similar singular vectors.** To characterize how these singular subspaces evolve throughout denoising, we measure their alignment between different steps with principal-angle similarity. Given $\mathbf{X}_t = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^\top$ and $\mathbf{X}_{t+k} = \mathbf{U}_{t+k}\mathbf{\Sigma}_{t+k}\mathbf{V}_{t+k}^\top$, we compute

$$\begin{aligned} &\cos(\theta_i^{(\mathbf{U},t,k)}) = \left[\mathbf{U}_t^\top \mathbf{U}_{t+k}\right]_{i,i}, \qquad \cos(\theta_i^{(\mathbf{V},t,k)}) = \left[\mathbf{V}_t^\top \mathbf{V}_{t+k}\right]_{i,i}, \\ &\overline{\cos(\theta^{(\mathbf{U},t,k)})} = \frac{1}{a}\sum_{i=1}^{a}\cos(\theta_i^{(\mathbf{U},t,k)}), \qquad \overline{\cos(\theta^{(\mathbf{V},t,k)})} = \frac{1}{a}\sum_{i=1}^{a}\cos(\theta_i^{(\mathbf{V},t,k)}). \end{aligned} \tag{2}$$

$[\cdot]_{i,i}$ denotes the $i^{th}$ diagonal element of the matrix. $\cos(\theta_i^{(\mathbf{U},t,k)})$ is the cosine similarity of the $i^{th}$ singular vectors between different steps, and $a$ denotes the number of evaluated singular directions. A higher average cosine similarity indicates stronger subspace consistency across denoising steps.

We plot the cosine similarity to preceding steps at each diffusion step under various gap $k$ configurations. As shown in Figure 2(a), the leading $\mathbf{U}$ and $\mathbf{V}$ subspaces remain highly aligned at temporal gap of up to eight denoising steps, with a cosine similarity mostly above 0.95.

**Singular values exhibit a linear extrapolation pattern.** We observe that the singular values exhibit an approximately linear extrapolation pattern with respect to those from preceding diffusion steps. Specifically, we denote the singular values of the fully computed features at steps $m$, $n$, and $p$ as $\mathrm{diag}(\mathbf{\Sigma}_m)$, $\mathrm{diag}(\mathbf{\Sigma}_n)$, and $\mathrm{diag}(\mathbf{\Sigma}_p)$ $(m < n < p)$, respectively. We compute $\alpha_{m,n,p} = \frac{\langle \mathrm{diag}(\mathbf{\Sigma}_p) - \mathrm{diag}(\mathbf{\Sigma}_n), \mathrm{diag}(\mathbf{\Sigma}_n) - \mathrm{diag}(\mathbf{\Sigma}_m) \rangle}{\|\mathrm{diag}(\mathbf{\Sigma}_n) - \mathrm{diag}(\mathbf{\Sigma}_m)\|_2^2}$. We randomly sample a few combinations of $(m, n)$, and plot the value of $\alpha_{m,n,p}$ with respect to $p$. As shown in Fig. 2(b), under fixed $m$ and $n$, $\alpha_{m,n,p}$ increases almost linearly as $p$ increases, with a slope close to $\frac{1}{n-m}$, i.e., $\alpha_{m,n,p} \approx \frac{p-n}{n-m}$. It suggests that singular values of subsequent features can be extrapolated from those of two preceding steps:

$$\mathrm{diag}(\hat{\mathbf{\Sigma}}_p) \approx \mathrm{diag}(\mathbf{\Sigma}_n) + \frac{p-n}{n-m}\left(\mathrm{diag}(\mathbf{\Sigma}_n) - \mathrm{diag}(\mathbf{\Sigma}_m)\right), \qquad p > n > m. \tag{3}$$

This approximation empirically introduces only a $0.71\%$ error relative to the corresponding fully computed features throughout denoising, motivating us to approximate subsequent features by extrapolating their preceding singular values, thereby reducing computational overhead by caching.

**Singular values can be approximated with a constant scaling factor.** We further investigate the relationship between feature singular values across nearby full computation steps separated by several steps. Given the nearly identical $\mathbf{U}, \mathbf{V}$ subspaces between the two steps (Figure 3(a)) , we examine the ratio of singular value norms $\frac{\|\mathrm{diag}(\mathbf{\Sigma}_t)\|_2}{\|\mathrm{diag}(\mathbf{\Sigma}_q)\|_2}$ between two nearby steps $t$ and $q$, as shown in Figure 3(b). The ratio $\frac{\|\mathrm{diag}(\mathbf{\Sigma}_t)\|_2}{\|\mathrm{diag}(\mathbf{\Sigma}_q)\|_2}$ remains nearly constant, suggesting that the singular values of neighboring features differ mainly by a constant scaling factor. Motivated by this, we can approximate the current feature by simply rescaling previous step's singular values, thereby avoiding expensive computations.

## 4 METHOD

Based on our key observations and deep insights for the singular vectors and values of denoising features across different steps, we propose a spectral feature caching framework to accelerate diffusion-based world models. The core idea is to approximate diffusion features by reusing cached

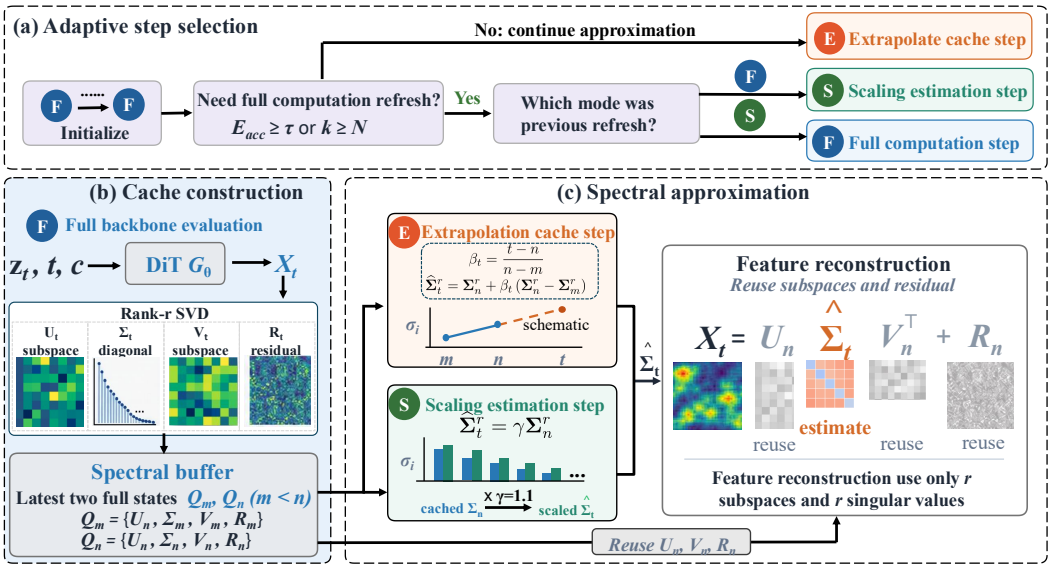



Figure 4: Overview of SpectralCache.

results of preceding full computation steps, thus skipping computations and achieving acceleration. Specifically, we define two kinds of diffusion steps: full computation step and approximation step. In full computation step, we go through the normal full diffusion computations and cache the singular vectors/values. In the approximation step, we do not perform the diffusion computations. Instead, the approximated features reuse the highly consistent singular subspaces from preceding full computation steps, while the singular values are estimated through linear extrapolation or scaling, following their respective spectral evolution patterns based on our observations and insights.

### 4.1 Full Computation Step

At a full computation step $t$, we perform a complete DiT backbone evaluation to obtain the final layer visual feature $\mathbf{F}_t$ and reshape it into $\mathbf{X}_t = \mathrm{mat}(\mathbf{F}_t) \in \mathbb{R}^{M \times D}$. Then we compute its rank-$r$ spectral decomposition for potential subsequent cache steps:

$$\mathbf{X}_t = \mathbf{U}_t^r \mathbf{\Sigma}_t^r (\mathbf{V}_t^r)^\top + \mathbf{R}_t, \tag{4}$$

where $\mathbf{U}_t^r$, $\mathbf{\Sigma}_t^r$, and $\mathbf{V}_t^r$ denote the top-$r$ spectral subspaces and singular values, and $\mathbf{R}_t$ is the truncated residual. Based on our deep insights, we choose $r = 128$ since it captures 96% feature energy. We store the resulting spectral state $\mathcal{Q}_t = \{\mathbf{U}_t^r, \mathbf{\Sigma}_t^r, \mathbf{V}_t^r, \mathbf{R}_t\}$ for subsequent steps. The spectral states of only two most recent full computation steps are cached for reuse.

### 4.2 Approximation Steps

At an approximation step, we bypass the full DiT backbone computation and reconstruct the visual feature by reusing the saved singular subspaces and residual while estimating only the dominant singular values. Specifically, the current visual feature at step $t$ is approximated as

$$\widehat{\mathbf{X}}_t = \mathbf{U}_n^r \widehat{\mathbf{\Sigma}}_t^r (\mathbf{V}_n^r)^\top + \mathbf{R}_n. \tag{5}$$

where $\mathbf{U}_n^r$ and $\mathbf{V}_n^r$ are spectral subspaces of the most recent full computation step. Since the leading singular subspaces remain highly stable across nearby denoising steps as demonstrated in Section 3, we directly reuse $\mathbf{U}_n^r$ and $\mathbf{V}_n^r$. The residual $\mathbf{R}_n$ is also directly reused since it accounts for only about $3\%$ of the total feature energy on average throughout generation.

$\widehat{\mathbf{\Sigma}}_t^r$ is estimated from singular values of previous full computation steps. We employ two estimation strategies according to the current caching state: linear extrapolation for consecutive cache steps and singular value scaling when a full computation step is triggered but can be safely skipped. Thus, each approximation step requires updating only $r$ singular values while reusing other spectral components from the most recent full computation step, avoiding expensive DiT backbone computations.

#### 4.2.1 SINGULAR VALUE CACHE THROUGH LINEAR EXTRAPOLATION

We can estimate singular values through linear extrapolation for consecutive denoising steps. Specifically, let $m < n < t$ denote denoising steps, where $m$ and $n$ are the two most recent full computation steps and $t$ is the current step to be approximated. Based on approximately linear evolution of singular values observed in Section 3, we can extrapolate singular values of current step $t$ from step $m$ and $n$:

$$\widehat{\mathbf{\Sigma}}_t^r = \mathbf{\Sigma}_n^r + \beta_t \left(\mathbf{\Sigma}_n^r - \mathbf{\Sigma}_m^r\right), \qquad \beta_t = \frac{t-n}{n-m}, \tag{6}$$

As shown above, $\widehat{\mathbf{\Sigma}}_t^r$ is mainly determined by the step distance of $m, n, t$, and the difference of cached visual features $\mathbf{\Sigma}_n^r$ and $\mathbf{\Sigma}_m^r$. Our analysis shows that this approximation empirically introduces only a 0.71% error relative to the corresponding ground truth fully computed features throughout denoising, showing that this approximation is highly accurate with minor errors.

#### 4.2.2 SINGULAR VALUE ESTIMATION THROUGH SCALING

We propose another way to estimate singular values. As observed in Section 3, the singular value magnitudes differ primarily by a constant scaling factor between nearby cache-triggered full computation steps. Thus, we can estimate singular values at full computaion step $t$ as

$$\widehat{\mathbf{\Sigma}}_t^r = \gamma \mathbf{\Sigma}_n^r, \tag{7}$$

where $\gamma$ is a fixed scaling factor controlling the magnitude adjustment, and $n < t$ denotes its most recent full computation step. We use $\gamma = 1.1$ (measured in Section 3) for all main experiments.

### 4.3 APPROXIMATION ERROR ACCUMULATION TO TRIGGER FULL COMPUTATION

For each diffusion step, we need to determine whether it is a full computation step or an approximation step. To complete this, we adopt an approximation error accumulation strategy to trigger full computation steps. In detail, although our efficient approximation steps introduces minor errors in each step, consecutive approximations across multiple denoising steps may gradually accumulate the estimation errors. When this accumulation error is large enough, we stop approximation and a full computation step is triggered to ensure high quality generation.

Following Feng et al. (2026b), we monitor approximation error over chaotic tokens to determine when cached feature approximation should be terminated. Specifically, we accumulate the average curvature-weighted feature variation across consecutive extrapolation-based cache steps:

$$E_{\text{acc}} \leftarrow E_{\text{acc}} + \frac{1}{|\mathcal{T}_{\text{chaotic}}|} \sum_{q \in \mathcal{T}_{\text{chaotic}}} e_q(t), k \leftarrow k+1. \tag{8}$$

where $\mathcal{T}_{\text{chaotic}}$ denotes the set of tokens exhibiting less predictable temporal feature variations, $k$ denotes the total number of cache steps performed, and $e_q(t)$ denotes the curvature-weighted approximation error of the $q$-th chaotic token following Feng et al. (2026b). In general, if the accumulated error is smaller than a threshold $E_{\text{acc}} < \tau$ and the maximum cache interval $N$ is not reached, the diffusion goes through the approximation step for efficiency. Once $E_{\text{acc}} \geq \tau$ or the cache interval $k$ reaches $N$ Feng et al. (2026b), a full computation step is triggered to prevent excessive approximation error.

### 4.4 OVERALL PIPELINE

We show our overall pipeline in Figure 4 with detailed algorithm in Appendix A.2. After two initial full computation steps, it goes through consecutive approximation steps with linear extrapolation for singular values, meanwhile recording the accumulated error $E_{\text{acc}}$ Feng et al. (2026b). If $E_{\text{acc}} > \tau$ or the maximum cache interval is reached, ideally a full computation step is triggered for a feature refresh to reduce error. In practice, instead of always executing the expensive full computation step when triggered, we alternatively adopt full computation or approximation with scaling of singular values, yielding additional acceleration. Specifically, if the previous triggered step is full computation, we use approximation step with scaling to estimate singular values for the current triggered step. Otherwise if the previous triggered step is approximation step, we adopt a full computation step. Then after the triggered step, it continues to approximation steps with linear extrapolation until triggered again by accumulation error.

# 5 EXPERIMENTS

## 5.1 EXPERIMENTAL SETTINGS

**Models.** We evaluate our method on two SOTA mainstream diffusion based world models: HunyuanVoyager-13B Huang et al. (2025) and Aether-5B Zhu et al. (2025). Both models take images, text, and camera trajectories as inputs and jointly generate RGB videos and corresponding depth maps. These models take an initial image, an instruction, and camera poses as conditioning inputs to generate 3D videos comprising both RGB frames and scene depth.

**Datasets and Evaluation Metric.** For the video generation datasets and evaluation metrics, we follow the experimental settings of prior works Song et al. (2025); Feng et al. (2026b). For generation quality evaluation, we similarly adopt WorldScore Duan et al. (2025), a comprehensive benchmark that assesses world generation in terms of both controllability and visual quality. Additionally, we evaluate the perceptual consistency of the generated videos by measuring their discrepancy from the no-cache baseline using PSNR, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018). Following the evaluation protocol of existing work Song et al. (2025), we also evaluate the 3D reconstruction metrics of generation videos with Sintel dataset Butler et al. (2012), which contains AbsRel, $\delta_1, \delta_2$, ATE, and RPE-t/RPE-r. We also report speedup and peak memory.

**Baselines.** We compare our method against several effective training-free diffusion model caching methods, including DuCa Zou et al., ToCa Zou et al. (2024), TaylorSeer Liu et al. (2025b), Hi-Cache Feng et al. (2026a), TeaCache Liu et al. (2025a), EasyCache Zhou et al. (2025), HERO Song et al. (2025), and WorldCache Feng et al. (2026b).

**Implementation Details.** All experiments are conducted on two NVIDIA Pro 6000 96GB GPUs. We use a rank of 128 for all experiments and set $\gamma = 1.1$. For other settings, we follow prior works Feng et al. (2026b); Zhu et al. (2025); Song et al. (2025) to make a fair comparison (see Appendix A.3).

| Method | Benchmark Evaluation | | Perceptual Metrics | | | Acceleration & Memory | |
|---|---|---|---|---|---|---|---|
| | WS Static↑ | WS Dynamic↑ | PSNR↑ | SSIM↑ | LPIPS↓ | Speedup↑ | Memory(GB)↓ |
| **HunyuanVoyager-13B** (Huang et al., 2025) ($512 \times 768$, frames = 49) | | | | | | | |
| Voyager | 66.28 | 46.40 | $\infty$ | 1.000 | 0.000 | 1.00× | 50.44 |
| DuCa | 53.87 | 37.71 | 16.66 | 0.508 | 0.486 | 1.30× | 109.70 |
| ToCa | 47.49 | 33.24 | 15.51 | 0.409 | 0.558 | 1.01× | 107.35 |
| TaylorSeer | 62.46 | 43.72 | 18.32 | 0.615 | 0.293 | 0.88× | 163.79 |
| HiCache | 63.80 | 44.66 | 18.56 | 0.623 | 0.281 | 0.96× | 163.79 |
| TeaCache | 60.88 | 42.61 | 16.25 | 0.565 | 0.372 | 3.38× | 56.52 |
| EasyCache | 64.16 | 44.91 | 21.76 | 0.737 | 0.208 | 3.58× | 50.98 |
| HERO | 62.37 | 43.67 | 17.71 | 0.601 | 0.315 | 0.96× | 173.42 |
| WorldCache | 64.89 | 45.43 | 23.49 | 0.770 | 0.176 | 3.65× | 50.58 |
| **SpectralCache (Ours)** | **65.90** | **46.13** | **25.06** | **0.818** | **0.154** | **5.22**× | 50.54 |
| **Aether-5B** (Zhu et al., 2025) ($480 \times 720$, frames = 41) | | | | | | | |
| Aether | 64.60 | 45.22 | $\infty$ | 1.000 | 0.000 | 1.00× | 46.58 |
| DuCa | 60.17 | 42.12 | 26.68 | 0.838 | 0.151 | 1.63× | 61.44 |
| ToCa | 60.15 | 42.11 | 26.68 | 0.839 | 0.151 | 1.62× | 61.78 |
| TaylorSeer | 57.11 | 39.97 | 22.92 | 0.713 | 0.324 | 1.66× | 77.32 |
| HiCache | 58.96 | 41.27 | 24.93 | 0.784 | 0.226 | 1.65× | 77.32 |
| TeaCache | 60.95 | 42.67 | 26.60 | 0.843 | 0.138 | 1.57× | 46.78 |
| EasyCache | 62.89 | 44.02 | 22.84 | 0.720 | 0.186 | 1.49× | 46.59 |
| HERO | 58.62 | 41.04 | 23.56 | 0.741 | 0.259 | 1.36× | 75.08 |
| WorldCache | 63.68 | 44.72 | 31.87 | 0.924 | 0.066 | 1.68× | 46.59 |
| **SpectralCache (Ours)** | **65.44** | **45.82** | **32.73** | **0.959** | **0.050** | **2.20**× | 46.59 |

Table 1: Quantitative comparison on Image-to-World generation. **Bold** indicates the best result. Layer-wise methods marked with * exceed single-GPU memory limits and require CPU offloading.

## 5.2 MAIN RESULTS

**Generation Results.** Table 1 presents the quantitative results on HunyuanVoyager-13B and Aether-5B. Our method consistently achieves the best overall generation quality among all acceleration baselines while providing substantial inference acceleration. On HunyuanVoyager-13B, our method achieves the highest WorldScores of 65.94 and 46.16 for static and dynamic scenes, respectively, outperforming WorldCache by 1.05 and 0.73 points. It also improves perceptual fidelity, increasing

PSNR from 23.49 to 25.06 and SSIM from 0.770 to 0.831, while delivering a $5.22\times$ end-to-end speedup with nearly unchanged GPU memory consumption (50.54 GB vs. 50.44 GB). Similar improvements are observed on Aether-5B, where our method achieves the best WorldScores of 65.44 and 45.82 for static and dynamic scenes, surpassing WorldCache by 1.76 and 1.10 points. It further improves PSNR from 31.87 to 32.73 and SSIM from 0.924 to 0.959, while achieving a $2.20\times$ speedup compared with $1.68\times$ for WorldCache. These results demonstrate that exploiting the low-rank structure and temporal evolution of singular components effectively improves inference efficiency while preserving high generation quality.

| **Method** | **Video Depth Estimation** | | | **Camera Pose Estimation** | | | **Acceleration & Memory** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Abs Rel↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | ATE↓ | RPE trans.↓ | RPE rot.↓ | Latency(s)↓ | Speed↑ | Memory (GB)↓ |
| Aether | 0.340 | 0.502 | 0.738 | 0.177 | 0.068 | 0.780 | 55.42 | 1.00× | 50.19 |
| DuCa | 0.341 | 0.475 | 0.694 | 0.209 | 0.069 | 0.904 | 28.15 | 1.97× | 52.70 |
| ToCa | 0.341 | 0.476 | 0.694 | 0.209 | 0.069 | 0.904 | 28.02 | 1.98× | 52.70 |
| TaylorSeer | 0.361 | 0.460 | 0.718 | 0.197 | 0.068 | 1.134 | 26.71 | 2.07× | 58.57 |
| HiCache | 0.346 | 0.472 | 0.712 | 0.204 | 0.069 | 1.004 | 26.51 | 2.09× | 58.57 |
| TeaCache | 0.341 | 0.496 | 0.724 | 0.183 | 0.068 | 0.797 | 25.85 | 2.14× | 50.20 |
| EasyCache | 0.390 | 0.479 | 0.725 | 0.183 | 0.069 | 1.061 | 27.76 | 2.00× | 50.20 |
| HERO | 0.347 | 0.490 | 0.716 | 0.181 | 0.071 | 0.861 | 27.44 | 1.96× | 61.56 |
| Worldcache | 0.341 | 0.508 | 0.741 | 0.184 | 0.068 | **0.796** | 21.20 | 2.61× | 50.20 |
| **SpectralCache (Ours)** | **0.331** | **0.520** | **0.773** | **0.179** | **0.060** | 0.800 | **16.06** | **3.45**× | **50.20** |

Table 2: Quantitative comparison on 3D reconstruction on Aether. **Bold** indicates the best result.

**3D Reconstruction Results.** Table 2 reports the 3D reconstruction performance on Aether-5B, including both video depth and camera pose estimation. Compared with WorldCache, our method achieves the best overall depth estimation performance among accelerated methods, reducing Abs Rel from 0.341 to 0.331 while improving the accuracies under $\delta < 1.25$ and $\delta < 1.25^2$ from 0.508 and 0.741 to 0.520 and 0.773, respectively. For camera pose estimation, our method further reduces ATE from $0.184$ to $0.179$ and RPE trans. from 0.068 to 0.060 compared with WorldCache, while maintaining comparable RPE rot. (0.800 vs. 0.796). More importantly, these improvements are achieved with a $3.45\times$ end-to-end speedup, approximately $1.3\times$ higher than the $2.61\times$ speedup of the strongest baseline. These results demonstrate that our method effectively preserves geometric and spatial information required for 3D world reconstruction while substantially reducing inference cost.

### 5.3 Ablation Studies

**Effects of Different Number of Ranks.** Table 3 studies the effect of the retained rank on generation quality and inference efficiency. Increasing the rank generally improves generation quality at a modest cost in acceleration. We adopt rank 128 as the default setting, which achieves a favorable trade-off with a PSNR of $25.06$, an SSIM of $0.818$, and a $5.22\times$ speedup. Further increasing the rank to 256 yields only marginal quality improvements while slightly reducing the speedup.

**Effects of Different Maximum Cache Intervals.** As shown in Table 4, increasing the maximum cache interval $N$ improves acceleration but gradually degrades generation quality due to larger approximation errors. We choose $N = 4$ as the default setting, which achieves a favorable trade-off between efficiency and quality, with a PSNR of $25.06$, an SSIM of $0.818$, and a $5.22\times$ speedup. Increasing $N$ to 5 or 6 further improves the speedup, but leads to noticeable quality degradation.

| Rank | PSNR↑ | SSIM↑ | Speedup↑ |
|---|---|---|---|
| Voyager | $\infty$ | 1.000 | 1.00× |
| Rank=32 | 24.67 | 0.806 | 5.28× |
| Rank=64 | 24.88 | 0.813 | 5.26× |
| Rank=128 | 25.06 | 0.818 | 5.22× |
| Rank=256 | 25.12 | 0.820 | 5.15× |

Table 3: Ablation study on effects of ranks.

| Interval | PSNR↑ | SSIM↑ | Speedup↑ |
|---|---|---|---|
| Voyager | $\infty$ | 1.000 | 1.00× |
| $N = 3$ | 25.33 | 0.822 | 4.87× |
| $N = 4$ | 25.06 | 0.818 | 5.22× |
| $N = 5$ | 24.72 | 0.810 | 5.44× |
| $N = 6$ | 24.63 | 0.806 | 5.78× |

Table 4: Ablation study on cache step intervals.

| Methods | PSNR↑ | SSIM↑ | WS-Static↑ | WS-Dynamic↑ | Latency↓ | Memory (GB)↓ |
|---|---|---|---|---|---|---|
| Voyager | $\infty$ | 1.000 | 66.28 | 46.40 | 1.00 × | 50.44 |
| HERO | 17.71 | 0.601 | 62.37 | 43.67 | 0.96× | 173.42 |
| WorldCache | 23.49 | 0.770 | 64.89 | 45.43 | 3.65 × | 50.58 |
| SpectralCache (Ours)* | **26.09** | **0.832** | **66.20** | **46.31** | 3.84 × | 50.44 |
| **SpectralCache (Ours)** | 25.06 | 0.818 | 65.90 | 46.13 | **5.22** × | **50.44** |

Table 5: Ablation study on the effect of singular value scaling for full computation. * denotes our method with singular value scaling disabled at triggered full computation steps.

**Effect of Singular Value Scaling for Full Computation.** We further ablate the contribution of singular value scaling for full computation. As shown in Table 5, when singular value scaling is disabled at triggered steps, our variant with triggered full computation achieves a even stronger quality but with a limited speedup of $3.84\times$, compared with our $5.22\times$ speedup by enabling singular value scaling. These results demonstrate that singular value scaling effectively reduces expensive full backbone evaluations while maintaining high generation quality.

**Effects of Different Singular Value Scaling Factors.** As discussed in Section 3, cache-triggered full-computation steps exhibit stable singular subspaces and an approximately constant singular-value scaling factor of $1.1$ relative to the preceding full-computation step. We therefore ablate the scaling factor $\gamma$ in Figure 5. The best performance is achieved at $\gamma = 1.1$, with a PSNR of $25.06$ and an SSIM of $0.818$, consistent with our empirical observation.

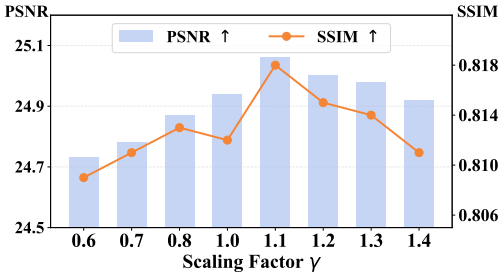



Figure 5: Ablation study on scaling factors $\gamma$.

| Operation | Avg. Latency Overhead (ms) |
|---|---|
| Full DiT Computation | 17942.06 |
| SVD Decomposition | 82.16 |
| Extrapolation Cache Step | 16.40 |
| Scaling Approximation Step | 45.57 |
| Error Computation | 0.33 |

Table 6: Average latency overhead of different operations in SpectralCache pipeline for HunyuanVoyager-13B.

### 5.4 Cost Analysis

We further analyze the additional computational and memory overhead introduced by our spectral caching pipeline. Our method only needs to retain the low-rank SVD states from two most recent full computation steps, resulting in approximately $0.1$ GB additional cache memory, which accounts for only about $0.2\%$ of the full model with 50.44 GB. We compare the computational costs of the SVD and approximation operations with that of a single full DiT computation, as shown in Table 6. The additional memory and computational costs introduced by SVD, spectral approximation, and error estimation are minor compared with the expensive full DiT computation, while enabling substantial reductions in the number of full backbone evaluations.

## 6 Conclusion

In this paper, we present **SpectralCache**, a training-free spectral feature caching framework for accelerating diffusion-based world models. Unlike existing caching methods that primarily exploit temporal redundancy at the feature or token level, SpectralCache leverages the underlying spectral structure of diffusion features. We reveal that world-model features exhibit stable singular subspaces across nearby denoising steps, while their singular values follow predictable evolution patterns. Based on these observations, SpectralCache reuses stable singular subspaces and efficiently estimates singular values through linear extrapolation for consecutive cache steps and singular value scaling for cache-triggered full computation steps, substantially reducing expensive backbone evaluations. Extensive experiments on representative world models demonstrate that SpectralCache consistently improves both generation quality and inference efficiency over existing training-free caching methods, achieving up to $5.22\times$ acceleration while preserving high generation quality.

## AI USE STATEMENT

In this work, large language models (LLMs) were used solely as a general-purpose writing assistant. Their role was limited to correcting grammar, fixing typographical errors, and polishing the language for clarity and readability.

## REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801. IEEE, 2025.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pp. 611–625. Springer, 2012.

Guantao Chen, Shikang Zheng, Yuqi Lin, and Linfeng Zhang. Forecast the principal, stabilize the residual: Subspace-aware feature caching for efficient diffusion transformers. *arXiv preprint arXiv:2601.07396*, 2026.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9): 10850–10869, 2023.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 27713–27724, 2025.

Liang Feng, Shikang Zheng, Jiacheng Liu, Yuqi Lin, Qinming Zhou, Peiliang Cai, Xinyu Wang, Junjie Chen, Chang Zou, Yue Ma, et al. Hicache: A plug-in scaled-hermite upgrade for taylor-style cache-then-forecast diffusion acceleration. In *International Conference on Learning Representations*, volume 2026, pp. 112361–112386, 2026a.

Weilun Feng, Haotong Qin, Chuanguang Yang, Xiangqi Li, Han Yang, Yuqi Li, Zhulin An, Libo Huang, Michele Magno, and Yongjun Xu. S2q-vdit: Accurate quantized video diffusion transformer with salient data and sparse token distillation. *arXiv preprint arXiv:2508.04016*, 2025a.

Weilun Feng, Chuanguang Yang, Haotong Qin, Xiangqi Li, Yu Wang, Zhulin An, Libo Huang, Boyu Diao, Zixiang Zhao, Yongjun Xu, et al. Q-vdit: Towards accurate quantization and distillation of video-generation diffusion transformers. *arXiv preprint arXiv:2505.22167*, 2025b.

Weilun Feng, Guoxin Fan, Haotong Qin, Mingqiang Wu, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Dingrui Wang, Longlong Liao, et al. Worldcache: Accelerating world models for free via heterogeneous token caching. *arXiv preprint arXiv:2603.06331*, 2026b.

Bowei Guo, Shengkun Tang, Cong Zeng, and Zhiqiang Shen. Mosaicdiff: Training-free structural pruning for diffusion model acceleration reflecting pretraining dynamics. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1655–1664. IEEE, 2025.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *Advances in neural information processing systems*, 37:90736–90770, 2024.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (TOG)*, 44(6): 1–15, 2025.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15240–15252. IEEE, 2025.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363. IEEE, 2025a.

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *Advances in Neural Information Processing Systems*, 37:115377–115408, 2024.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772. IEEE, 2024.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1972–1981. IEEE, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Quanjian Song, Xinyu Wang, Donghao Zhou, Jingyu Lin, Cunjian Chen, and Yue Ma. Hero: Hierarchical extrapolation and refresh for efficient world models. *arXiv preprint arXiv:2508.17588*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Junyi Wu, Zhiteng Li, Zheng Hui, Yulun Zhang, Linghe Kong, and Xiaokang Yang. Quantcache: Adaptive importance-guided quantization with hierarchical latent and layer caching for video generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15035–15044. IEEE, 2025.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pp. 2226–2240. PMLR, 2023.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.

Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. *Advances in Neural Information Processing Systems*, 37:78630–78652, 2024.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

Jintao Zhang, Pengle Zhang, Jun Zhu, Jianfei Chen, et al. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations*, volume 2025, pp. 71566–71585, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Yang Zhang, Er Jin, Yanfei Dong, Ashkan Khakzar, Philip Torr, Johannes Stegmaier, and Kenji Kawaguchi. Effortless efficiency: Low-cost pruning of diffusion models. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2024.

Zhixin Zheng, Xinyu Wang, Chang Zou, Shaobo Wang, and Linfeng Zhang. Compute only 16 tokens in one timestep: Accelerating diffusion transformers with cluster-driven feature caching. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10181–10189, 2025.

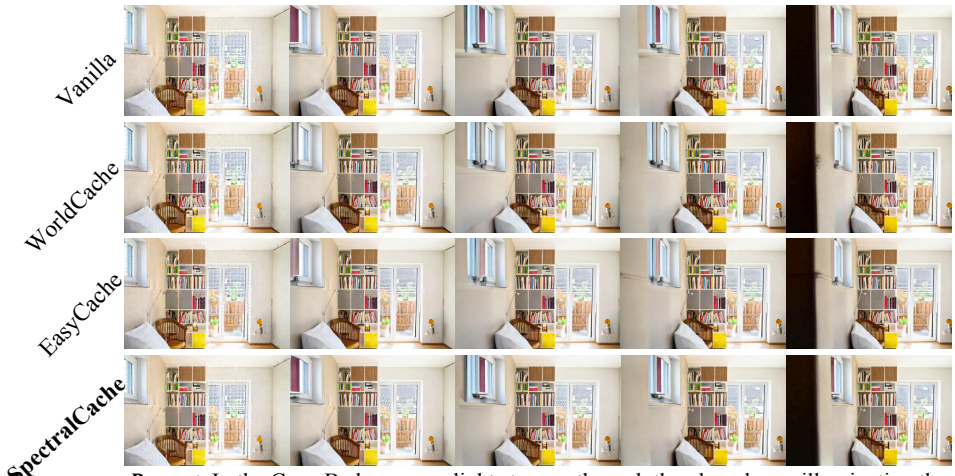
Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8535–8546. IEEE, 2025.

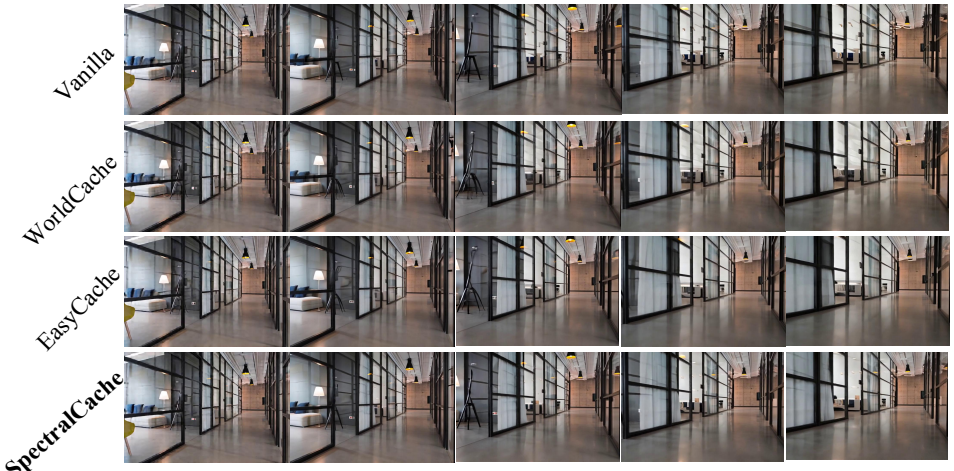
Chang Zou, Evelyn Zhang, Runlin Guo, Haohang Xu, Conghui He, Xuming Hu, and Linfeng Zhang. Accelerating diffusion transformers with dual feature caching, 2024. *URL https://arxiv.org/abs/2412.18911*.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

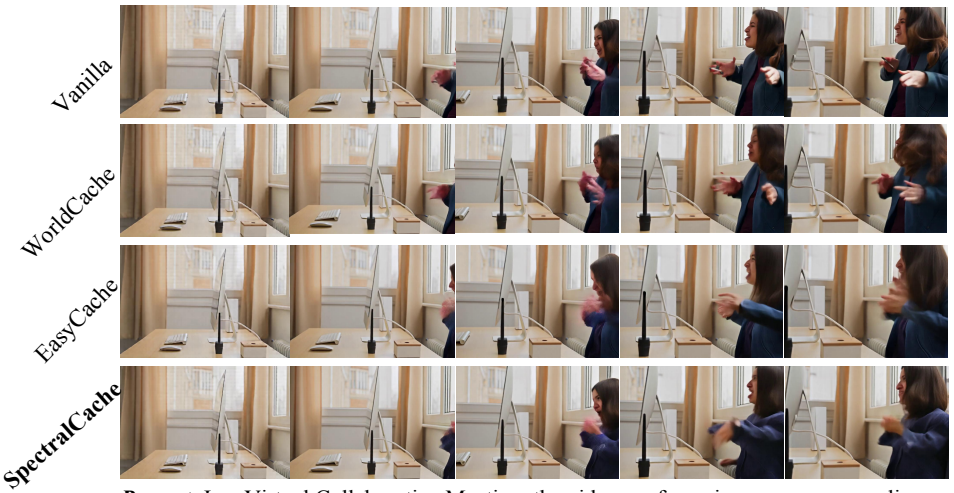
# A APPENDIX




***Prompt:*** In the Cozy Bedroom, sunlight streams through the glass doors, illuminating the bookshelf filled with colorful books and highlighting the warm wooden armchair.




***Prompt:*** In the Modern Office Hallway, sleek glass partitions reflect the glow of overhead lights, creating a sophisticated and spacious atmosphere.




***Prompt:*** In a Virtual Collaborative Meeting, the video conferencing screen comes alive with dynamic interactions as team members' avatars exchange ideas and express emotions.

Figure 6: More visual comparison on HunyuanVoyager-13B between SpectralCache and existing methods.

### A.1 More Visualization Results.

Here, we provide more visual comparisons to demonstrate the effectiveness of our proposed SpectralCache. The results are shown in Figure 6 and Figure 7.




***Prompt:*** In the Cozy Caravan Bedroom, sunlight streams through the window, illuminating the plush bed adorned with pillows and floral bedding, while a scenic view stretches beyond.




***Prompt:*** In the Modern Living Space, soft natural light bathes the sleek couch and dining table, creating an inviting and sophisticated ambiance.

Figure 7: More visual comparison on Aether-5B between SpectralCache and existing methods.

### A.2 Overall Framework

We consider a diffusion-based world model that generates video and geometric modalities through a sequence of denoising steps. Let $\mathbf{z}_t$ denote the noisy latent at step $t$. Conditioned on text, camera trajectories, and other control signals $\mathbf{c}$, a diffusion Transformer evaluates

$$\mathbf{F}_t = \mathcal{G}_\theta(\mathbf{z}_t, t, \mathbf{c}), \qquad \mathbf{y}_t = \mathcal{H}_\theta(\mathbf{F}_t), \tag{9}$$

where $\mathbf{F}_t$ denotes the final-layer visual features, $\mathcal{G}_\theta$ represents the expensive Transformer computation, and $\mathcal{H}_\theta$ projects the feature into denoised $\mathbf{y}_t$. The scheduler then updates the latent as

$$\mathbf{z}_{t+1} = \mathcal{S}(\mathbf{z}_t, \mathbf{y}_t, t). \tag{10}$$

$\mathcal{G}_\theta$ dominates the overall inference time cost due to repeated denoising precess. Following the description and notations in main paper, the overall framework of our algorithm can be shown as:

**Algorithm 1** Spectral Feature Caching for Diffusion-Based World Models

**Require:** Initial latent $\mathbf{z}_T$, backbone $\mathcal{G}_\theta$, output head $\mathcal{H}_\theta$, scheduler $\mathcal{S}$, rank $r$, error threshold $\tau$, maximum cache interval $N$, scaling factor $\gamma$

Initialize spectral buffer $\mathcal{B} \leftarrow \varnothing$
$E_{\text{acc}} \leftarrow 0$, $k \leftarrow 0$
last_refresh_true $\leftarrow$ False
**for** $t = T, T-1, \ldots, 1$ **do**
  **if** $|\mathcal{B}| < 2$ **then**
    $\mathbf{X}_t \leftarrow \mathcal{G}_\theta(\mathbf{z}_t, t, \mathbf{c})$, Compute $(\mathbf{U}_t^r, \mathbf{\Sigma}_t^r, \mathbf{V}_t^r, \mathbf{R}_t)$
    $\mathcal{Q}_t \leftarrow \{\mathbf{U}_t^r, \mathbf{\Sigma}_t^r, \mathbf{V}_t^r, \mathbf{R}_t, t\}$
    Append $\mathcal{Q}_t$ to $\mathcal{B}$ and keep the latest two states
    $E_{\text{acc}} \leftarrow 0$, $k \leftarrow 0$
    last_refresh_true $\leftarrow$ True
  **else**
    Let $\mathcal{Q}_m, \mathcal{Q}_n$ be the two latest true full-computation states in $\mathcal{B}$
    **if** $E_{\text{acc}} < \tau$ **and** $k < N$ **then**
      $\beta_t \leftarrow \dfrac{t-n}{n-m}$, $\widehat{\mathbf{\Sigma}}_t^r \leftarrow \mathbf{\Sigma}_n^r + \beta_t\,(\mathbf{\Sigma}_n^r - \mathbf{\Sigma}_m^r)$, $\widehat{\mathbf{X}}_t \leftarrow \mathbf{U}_n^r \widehat{\mathbf{\Sigma}}_t^r (\mathbf{V}_n^r)^\top + \mathbf{R}_n$
      Compute chaotic-token error $E(t)$:
      $E_{\text{acc}} \leftarrow E_{\text{acc}} + E(t)$, $k \leftarrow k+1$, $\mathbf{X}_t \leftarrow \widehat{\mathbf{X}}_t$
    **else**
      **if** last_refresh_true $=$ True **then**
        $\widehat{\mathbf{\Sigma}}_t^r \leftarrow \gamma \mathbf{\Sigma}_n^r$, $\widehat{\mathbf{X}}_t \leftarrow \mathbf{U}_n^r \widehat{\mathbf{\Sigma}}_t^r (\mathbf{V}_n^r)^\top + \mathbf{R}_n$, $\mathbf{X}_t \leftarrow \widehat{\mathbf{X}}_t$
        $E_{\text{acc}} \leftarrow 0$, $k \leftarrow 0$
        last_refresh_true $\leftarrow$ False
      **else**
        $\mathbf{X}_t \leftarrow \mathcal{G}_\theta(\mathbf{z}_t, t, \mathbf{c})$, Compute $(\mathbf{U}_t^r, \mathbf{\Sigma}_t^r, \mathbf{V}_t^r, \mathbf{R}_t)$
        $\mathcal{Q}_t \leftarrow \{\mathbf{U}_t^r, \mathbf{\Sigma}_t^r, \mathbf{V}_t^r, \mathbf{R}_t, t\}$
        Append $\mathcal{Q}_t$ to $\mathcal{B}$ and keep the latest two states
        $E_{\text{acc}} \leftarrow 0$, $k \leftarrow 0$
        last_refresh_true $\leftarrow$ True
      **end if**
    **end if**
  **end if**
  $\mathbf{y}_t \leftarrow \mathcal{H}_\theta(\mathbf{X}_t)$
  $\mathbf{z}_{t-1} \leftarrow \mathcal{S}(\mathbf{z}_t, \mathbf{y}_t, t)$
**end for**

Algorithm 1 summarizes the complete inference pipeline of SpectralCache. Given the initial latent, diffusion backbone, output head, and scheduler, we first perform two full backbone computations to initialize the spectral buffer $\mathcal{B}$. For each full computation feature, we apply low-rank decomposition and store its singular subspaces, singular values, and residual, while retaining only the two most recent spectral states. These states provide the spectral information required for subsequent feature approximation without repeatedly evaluating the expensive diffusion backbone. After initialization, SpectralCache dynamically switches among spectral extrapolation, singular-value scaling, and true full computation according to the accumulated approximation error and cache interval. When the accumulated error remains below the threshold $\tau$ and the number of consecutive cache steps $k$ is smaller than the maximum interval $N$, the current feature is approximated through spectral extrapolation. Specifically, we reuse the stable singular subspaces from the latest full computation state and linearly extrapolate its singular values from the two most recent full computation states. The corresponding residual is then added to reconstruct the current feature. After each approximation, the error of chaotic tokens is evaluated and accumulated to determine whether spectral extrapolation can continue. Once the accumulated error exceeds $\tau$ or the cache interval reaches $N$, a refresh is triggered to prevent excessive approximation error. Rather than directly performing an expensive backbone evaluation at every triggered step, SpectralCache further exploits the spectral consistency between neighboring full computation features. If the preceding refresh is a true full computation, the current triggered step is approximated by reusing its singular subspaces and residual while scaling the

singular values with factor $\gamma$, thereby skipping another expensive backbone evaluation. Otherwise, a true full computation is performed, and its newly decomposed spectral state is inserted into $\mathcal{B}$ to refresh the extrapolation basis. The approximation error and cache counter are then reset. Finally, the resulting feature is passed through the output head and diffusion scheduler to obtain the latent for the next denoising step.

### A.3 More details for Experiments

**Detailed Experimental Settings.** Following WorldCache Feng et al. (2026b), we evaluate on two multi-modal diffusion world models, HunyuanVoyager-13B and Aether-5B. For HunyuanVoyager-13B, we use the standard 50-step inference protocol to generate 49-frame videos at $512 \times 768$ resolution. For Aether-5B, we generate 41-frame videos at $480 \times 720$ resolution with 50 denoising steps, while using 30 steps for 3D reconstruction. Following the default configuration of Feng et al. (2026b), we set the token-grouping percentiles to static token $p_s = 0.3$ and chaotic token $p_c = 0.7$, and use an accumulated-error threshold of $\tau = 1.0$ for chaotic tokens. We evaluate generation quality using WorldScore and perceptual consistency against the no-cache baseline using PSNR and SSIM. All methods use identical conditioning inputs and sampling schedules for fair comparison. For WorldScore benchmark Duan et al. (2025), we follow its standard evaluation protocol, which formulates world generation as sequential next-scene generation under explicit scene and camera-trajectory specifications. WorldScore evaluates three aspects: controllability, generation quality, and dynamics. We report WorldScore-Static, which aggregates controllability and quality metrics, and WorldScore-Dynamic, which further incorporates motion accuracy, magnitude, and smoothness. For generation results evaluation, we follow the standard benchmark protocol Feng et al. (2026b); Duan et al. (2025); Song et al. (2025) across diverse indoor/outdoor scenarios and realistic/stylized conditions. We report both WorldScore-Static and WorldScore-Dynamic.

**Effect of Cache Error Threshold.** Table 7 studies the effect of the accumulated chaotic-token error threshold $\tau$, which controls when the accumulated approximation error becomes sufficiently large to terminate the current cache sequence and trigger a refresh computation. In our main experiments, we directly adopt $\tau = 1.0$, the default setting of WorldCache, without additional tuning for SpectralCache. A smaller $\tau$ imposes a stricter error constraint and therefore triggers full computations more frequently, resulting in higher generation quality but lower acceleration. For example, $\tau = 0.5$ achieves the highest PSNR of 25.39, but reduces the speedup to $4.42\times$. As $\tau$ increases, SpectralCache tolerates larger accumulated errors and performs more aggressive approximation, increasing the speedup from $4.83\times$ at $\tau = 0.7$ to $5.36\times$ at $\tau = 1.5$. However, overly large thresholds gradually degrade generation quality, with PSNR decreasing to 24.71 at $\tau = 1.5$. The inherited default $\tau = 1.0$ achieves a PSNR of 25.06 with a $5.22\times$ speedup, providing a favorable balance between generation fidelity and inference efficiency. These results also demonstrate that SpectralCache performs robustly across a reasonable range of error thresholds without requiring method-specific tuning of $\tau$.

Table 7: Ablation study on different cache error threshold $\tau$ .

| $\tau$ | PSNR↑ | Latency (rel. to base)↓ |
|---|---|---|
| EasyCache | 21.76 | 3.58 × |
| 0.5 | 25.39 | 4.42× |
| 0.7 | **25.06** | 4.83× |
| 1.0 | 25.06 | 5.22× |
| 1.3 | 24.80 | 5.32× |
| 1.5 | 24.71 | 5.36× |

**Token Grouping Percentiles.** We further study the sensitivity of SpectralCache to the token grouping percentiles $p_s$ and $p_c$ adopted from Feng et al. (2026b), which control the partition of tokens into static, regular, and chaotic groups according to their temporal variations. Specifically, $p_s$ determines the fraction of tokens regarded as static, while $p_c$ specifies the percentile threshold above which tokens are treated as chaotic. For example, $\{p_s, p_c\} = \{0.3, 0.8\}$ assigns the lowest 30% of tokens to the static group and the highest 20% to the chaotic group, with the remaining tokens treated as regular. As shown in Table 8, SpectralCache consistently achieves strong perceptual quality across

different grouping configurations, indicating that our spectral caching framework is relatively robust to the choice of token grouping percentiles. Among them, $\{p_s, p_c\} = \{0.3, 0.7\}$ achieves the best overall performance, reaching a PSNR of $25.06$ and an SSIM of $0.818$. Changing either threshold leads to a modest quality degradation; for instance, increasing $p_c$ from $0.7$ to $0.8$ decreases PSNR from $25.06$ to $24.84$, while reducing $p_s$ from $0.3$ to $0.2$ results in lower PSNR and SSIM. Therefore, consistent with the default grouping strategy in Feng et al. (2026b), we adopt $p_s = 0.3$ and $p_c = 0.7$ for SpectralCache in our main experiments.

Table 8: Ablation study on token grouping percentiles.

| $\{p_s, p_c\}$ | PSNR↑ | SSIM↑ |
|---|---|---|
| EasyCache | 21.76 | 0.737 |
| $\{0.3, 0.8\}$ | 24.84 | 0.813 |
| $\{0.3, 0.7\}$ | **25.06** | **0.818** |
| $\{0.3, 0.6\}$ | 24.98 | 0.814 |
| $\{0.2, 0.8\}$ | 24.81 | 0.808 |
| $\{0.2, 0.7\}$ | 24.94 | 0.811 |