A. Appendix

A.1. Pseudo Code for SVD-Cache Framework

Algorithm 1: SVD-Cache Framework

Input: Feature F_t, cached basis $(V_{\mathcal{C}}, \boldsymbol{\sigma}_{\mathcal{C}})$, previous low-rank feature $\widehat{F}_{k,t-\Delta}$, EMA coefficient β, rank k

Output: Predicted feature $\widehat{F}_{t+\Delta}$

```
// Stage 1: One-Time SVD on Reference Prompt (offline)
```

if *reference prompt (first time)* **then**

  Compute SVD: $F_{\text{ref}} = U\Sigma V^\top$;

  Cache basis: $V_{\mathcal{C}} \leftarrow V$, $\boldsymbol{\sigma}_{\mathcal{C}} \leftarrow \text{diag}(\Sigma)$;

```
// Stage 2: Low-Rank Subspace Reconstruction
```

Compute $U = F_t V_{\mathcal{C}} \,\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}})^{-1}$;

$(U_k, \boldsymbol{\sigma}_{\mathcal{C},k}, V_{\mathcal{C},k}) \leftarrow$ top-k components of $(U, \boldsymbol{\sigma}_{\mathcal{C}}, V_{\mathcal{C}})$;

$F_{k,t} = U_k \,\text{diag}(\boldsymbol{\sigma}_{\mathcal{C},k})\, V_{\mathcal{C},k}^\top$;

$R_t = F_t - F_{k,t}$;

```
// Stage 3: Hybrid Caching Strategy
// EMA prediction for low-rank component
```

$\widehat{F}_{k,t+\Delta} = \beta\, \widehat{F}_{k,t-\Delta} + (1-\beta)\, F_{k,t}$;

```
// Direct reuse for residual component
```

$\widehat{R}_{t+\Delta} = R_t$;

```
// Stage 4: Feature Reconstruction
```

$\widehat{F}_{t+\Delta} = \widehat{F}_{k,t+\Delta} + \widehat{R}_{t+\Delta}$;

return $\widehat{F}_{t+\Delta}$

A.2. Mathematical Derivation of Left Singular Matrix Reconstruction

In this section, we provide the mathematical justification for reconstructing the left singular matrix U using the cached basis $(V_{\mathcal{C}}, \boldsymbol{\sigma}_{\mathcal{C}})$ from a reference prompt.

Theoretical Foundation. Given the SVD decomposition of the reference feature matrix:

$$F_{\text{ref}} = U_{\text{ref}} \Sigma_{\text{ref}} V_{\text{ref}}^\top, \tag{12}$$

we cache the right singular vectors $V_{\mathcal{C}} = V_{\text{ref}}$ and singular values $\boldsymbol{\sigma}_{\mathcal{C}} = \text{diag}(\Sigma_{\text{ref}})$.

For a new feature matrix F_t at timestep t, if we assume that F_t can be well-approximated in the subspace spanned by $V_{\mathcal{C}}$, we can express:

$$F_t \approx U_t \,\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}})\, V_{\mathcal{C}}^\top, \tag{13}$$

where $U_t \in \mathbb{R}^{N \times r}$ represents the coordinates of F_t in the cached subspace.

Derivation of Reconstruction Formula. To solve for U_t, we multiply both sides by $V_{\mathcal{C}}$ from the right:

$$F_t V_{\mathcal{C}} = U_t \,\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}})\, V_{\mathcal{C}}^\top V_{\mathcal{C}}. \tag{14}$$

Since $V_{\mathcal{C}}$ is orthonormal ($V_{\mathcal{C}}^\top V_{\mathcal{C}} = I$), this simplifies to:

$$F_t V_{\mathcal{C}} = U_t \,\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}}). \tag{15}$$

Multiplying both sides by $\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}})^{-1}$ yields:

$$U_t = F_t V_{\mathcal{C}} \,\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}})^{-1}. \tag{16}$$

Geometric Interpretation. This reconstruction can be understood geometrically: $F_t V_{\mathcal{C}}$ projects the feature matrix onto the cached principal directions, and $\text{diag}(\boldsymbol{\sigma}_{\mathcal{C}})^{-1}$ normalizes these projections by the corresponding singular values. The resulting U_t represents the coefficients needed to express F_t as a linear combination of the cached basis weighted by the reference singular values.

Key Advantage. This approach avoids recomputing the full SVD for each new prompt, reducing computational complexity from $\mathcal{O}(ND\min(N,D))$ for full SVD to $\mathcal{O}(NDr)$ for basis projection, where $r \ll \min(N,D)$ is the rank of the cached basis. The validity of this approximation is empirically supported by our observation in Fig. 1(b) that V and $\boldsymbol{\sigma}$ remain stable across different prompts.

A.3. Detailed Experiment Settings

Model Configurations and Evaluation Protocols. Our experimental evaluation encompasses six distinct generative models across two primary tasks: text-to-image generation and text-to-video generation. For text-to-image generation, we evaluate four different models: FLUX.1-dev, FLUX.1-Schnell, FLUX.1-dev-int8 and FLUX with sparse attention. Each model is assessed using standardized benchmarks and protocols to ensure fair comparisons. For text-to-video generation, we utilize HunyuanVideo, which is evaluated using VBench framework to capture various aspects of video quality and coherence. Detailed configurations for each model, including resolution settings, prompt selections, and evaluation metrics, can be found in the subsequent sections.

A.3.1. Text-to-Image Generation

FLUX.1-dev. We evaluate FLUX.1-dev using the standard DrawBench protocol, which provides a diverse set of 200 prompts spanning multiple categories including animals, colors, conflicting, and fine-grained details. All images are generated at a resolution of 1024×1024 pixels, maintaining consistency with the model's optimal operating parameters. We assess generation quality using two metrics: ImageReward (IR) for evaluating photorealism and overall image

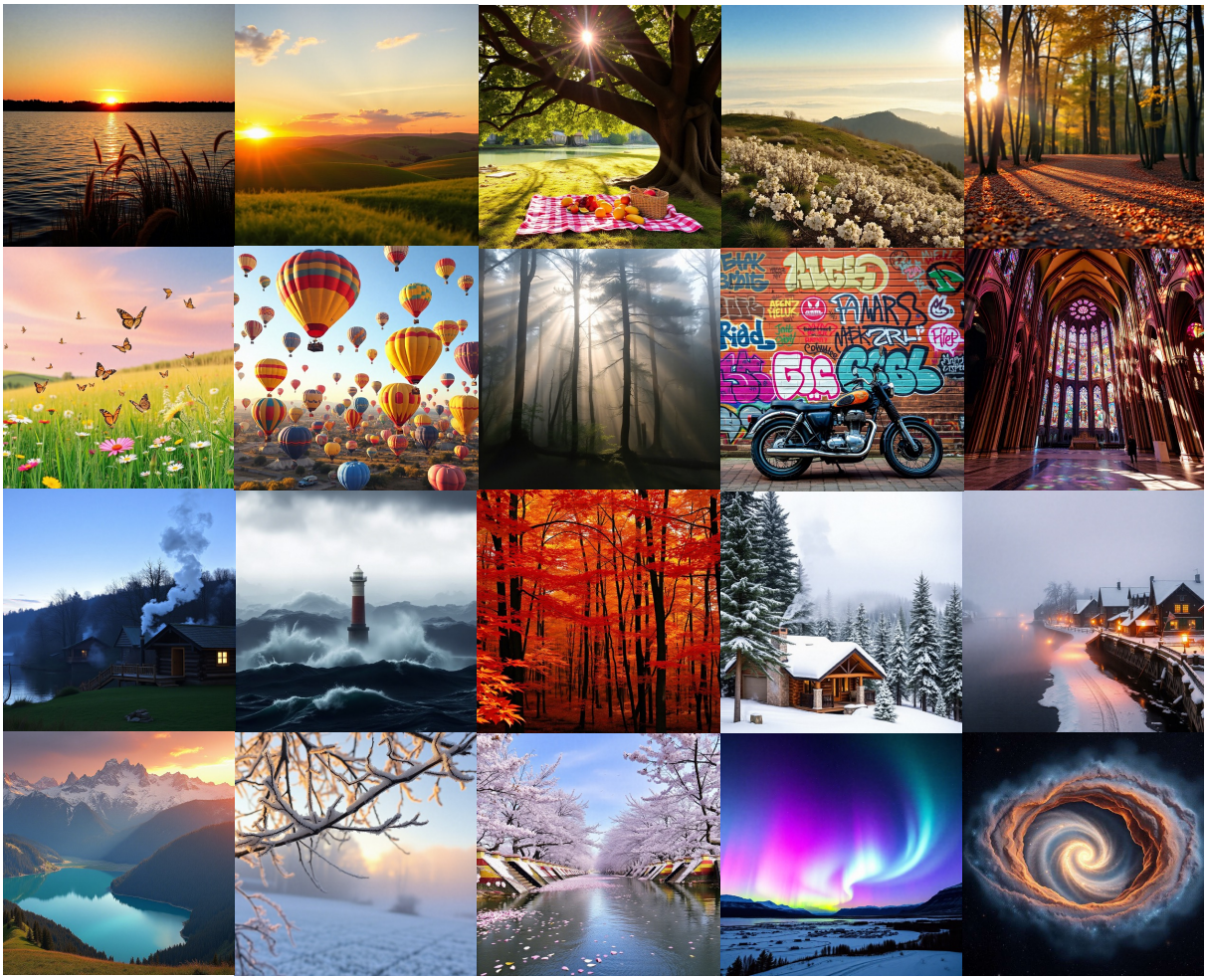

Figure 7. **Images of SVD-Cache on FLUX.1-Schnell.** SVD-Cache achieves 15.22 × acceleration while preserving image fidelity.

quality and CLIP Score for measuring text-image semantic alignment.

FLUX.1-schnell. FLUX.1-schnelll, a step-distillation model from FLUX.1-dev, is designed for rapid image generation. It is evaluated using the standard DrawBench protocol with 200 prompts at 1024×1024 resolution. Despite its focus on speed optimization, we maintain the same evaluation standards, assessing both generation quality through ImageReward and CLIP Score, as well as fidelity through PSNR, SSIM, and LPIPS metrics. This allows us to quantify the trade-offs between generation speed and output quality.

FLUX.1-dev-int8 FLUX.1-dev-int8, a quantized version of FLUX.1-dev, is designed for efficient inference on resource-constrained devices. It is evaluated using the standard DrawBench protocol with 200 prompts at 1024×1024 resolution. We assess generation quality using ImageReward, CLIP Score, PSNR, SSIM and LPIPS.

FLUX with sparse attention We utilize SpargeAttention, a universal training-free sparse attention accelerating language, image, and video models. Prompts, resolution and evaluation metrics are the same as FLUX.1-dev.

SVD-Cache on FLUX.1[dev]-int8, 7.61 ×

SVD-Cache with SparseAttention, 10.73 ×

Figure 8. **Images of SVD-Cache on FLUX.1[dev]-int8 and Sparse Attention.** SVD-Cache achieves 7.61 × and 10.73 × acceleration respectively while generating high-quality images.

A.3.2. Text-to-Video Generation

HunyuanVideo. For text-to-video generation, we evaluate HunyuanVideo using the comprehensive VBench framework, which provides multi-dimensional human-aligned assessments across 946 diverse prompts sourced from the VBench-full-info.json file. Videos are generated at 480×640 resolution with 65 frames each, providing substantial temporal content for thorough evaluation. VBench evaluates 18 critical aspects including but not limited to motion quality, visual appearance consistency, temporal coherence, semantic alignment with text prompts, and overall video quality through human-correlated metrics.