Towards Stabilized and Efficient Diffusion Transformers through Long-Skip-Connections with Spectral Constraints

Supplementary Material

6. Limitations

Despite achieving significant speedups, `Skip-DiT` inherits limitations from its DiT foundation. including high computational demands, reliance on large-scale training data, and quadratic complexity that hinders high-resolution generation. Furthermore, `Skip-DiT` introduces a marginal parameter overhead (5.5% for class-to-image, 3.5% for text-to-video) and extra GPU memory (scales with the batch size $\mathcal{N}$), requiring an additional $\mathcal{N} \times 0.37\%$ for the class-to-image model and $\mathcal{N} \times 4.55\%$ for the text-to-video model. These models remain feasible for deployment with proper $\mathcal{N}$.

7. Supplementary Experiments

Analysis of Block Selection for Caching To identify the optimal block for caching, we analyze the feature similarity across timesteps in the final three blocks of Latte-T2V (Table 7). Our analysis reveals that caching at block 27, the last DiT block, yields the best performance. This block, which is connected to block 0 via the primary Long Skip Connection (LSC), not only exhibits maximum feature similarity but also enables the highest speedup. These results demonstrate the superior caching efficiency achieved by `Skip-DiT`.

Table 7. Connection selection for caching in the text-to-video task. The best metrics are emphasized in **bold**.

LSC	Cached	Similarity(%)↑	VBench(%) ↑	PSNR ↑	LPIPS ↓	SSIM ↑
0	[1,26]	**99.88**	**75.51**	29.52	**0.06**	**0.89**
1	[2,25]	99.86	75.44	**29.65**	**0.06**	**0.89**
2	[3,24]	99.66	75.49	29.44	**0.06**	0.88

Comparison in Few-Step Generation We benchmarked `Skip-DiT` against a vanilla DiT in a few-step text-to-video generation scenario, using a 5 and 10-step DDIM scheduler instead of the standard 50 steps. As shown in Table 8, while the vanilla DiT holds a slight advantage in the SSIM metric, `Skip-DiT` achieves superior visual quality and semantic consistency. These results again validate the balanced and robust performance of our architecture, even in a highly accelerated, few-step setting.

8. Detailed Proof of Theorems

Consider an ideal denoising diffusion transformer M with L identical blocks, where the denoising capability induces the following fundamental properties:

Table 8. Text-to-video generation performance under the few-step scenario. The best metrics are emphasized in **bold**.

NFE	Model	ΔVBench(%)	PSNR ↑	LPIPS ↓	SSIM ↑
10 steps	**Latte**	↓1.73	15.96	0.47	**0.61**
	`Skip-DiT`	**↓0.94**	**16.18**	**0.43**	0.59
5 steps	**Latte**	↓9.40	12.94	0.73	**0.53**
	`Skip-DiT`	**↓8.55**	**13.17**	**0.69**	0.50

- *Noise Reduction Invariance*: Each transformer block $\mathcal{T}_l$ strictly reduces noise magnitude of input h. This directly implies the Jacobian spectral norm ($\sigma_{\max}$) constraint:

$$\sigma_{\max}\left(\frac{\partial \mathcal{T}_l}{\partial h}\right) \triangleq \gamma_l < 1 \tag{9}$$

- *Transformer Blocks Homogeneity*: Identical noise reduction ratio across layers $\gamma_l = \gamma, \quad \forall l \in \{1, ..., L\}$. Thus, the complete model satisfies

$$\sigma_{\max}(M) = \prod_{l=1}^{L} \gamma = \gamma^L \ll 1 \tag{10}$$

8.1. Theorem1

The spectral norm of the Jacobian matrix of DiT with Long-Skip-Connections is controlled tighter than that of Vanilla DiT M, making the Skip-DiT model more robust, numerically stable, and capable of converging faster.

Proof. Define layer transformations for $L/2 < l \leq L$ and their Jacobian matrices, where $\mathcal{T}$ denotes the Transformer block and $0 < \alpha < 1$:

$$h_{l+1}^{\text{vanilla}} = \mathcal{T}(h_l^{\text{vanilla}}), h_{l+1}^{\text{skip}} = (1-\alpha) \cdot \mathcal{T}(h_l^{\text{skip}}) + \alpha \cdot h_{L-l}^{\text{skip}}.$$

The corresponding Jacobian matrices J are:

$$J_l^{\text{vanilla}} = \frac{\partial \mathcal{T}(h_l)}{\partial h_l}, J_l^{\text{skip}} = (1-\alpha) \cdot \frac{\partial \mathcal{T}(h_l)}{\partial h_l} + \alpha \cdot \frac{\partial h_{L-l}}{\partial h_l}.$$

Applying subadditivity and submultiplicativity of spectral norms σ_{max}:

$$\sigma_{\max}(J_l^{\text{skip}}) \leq (1-\alpha)\gamma + \alpha\gamma^{2l-L}. \tag{11}$$

Given $\gamma < 1$ and $2l - L \geq 1$ for $l > L/2$, we establish the layer-wise bound:

$$\sigma_{\max}(J_l^{\text{skip}}) < \gamma = \sigma_{\max}(J_l^{\text{vanilla}}). \tag{12}$$

For the full model Jacobian $J = \prod_{l=1}^{L} J_l$, the spectral norms satisfy:

$$\sigma_{\max}(J^{\text{skip}}) \leq \prod_{l=0}^{L-1} \left[(1-\alpha)\gamma + \alpha\gamma^{2l-L}\right] < \prod_{l=0}^{L-1} \gamma = \gamma^L = \sigma_{\max}(J^{\text{vanilla}}). \quad (13)$$

This spectral radius reduction attenuates perturbation growth through $\|\delta h_L\| \leq (\sigma_{\max}(J))^L \|\delta h_0\|$, stabilizes gradient flow via $\|\nabla_\theta \mathcal{L}\| \leq \sigma_{\max}(J)\|\nabla_{h_L}\mathcal{L}\|$, and improves the Lipschitz constant $\text{Lip}(M_{\text{skip}}) < \text{Lip}(M_{\text{vanilla}})$. □

8.2. Theorem2

Under feature reuse with interval τ, Skip-DiT achieves tighter perturbation bounds than Vanilla DiT, enabling larger allowable reuse intervals while maintaining error tolerance $\epsilon_{\max}$

Proof. Let h_t be the reused feature with error ϵ_t, δ the perturbation bound per step, x the input, and θ the model parameters. Through the differential mean-value theorem and error propagation:

$$\|f(x;\theta+\Delta\theta) - f(x;\theta)\| \leq \sup_{\theta'} \left\| \frac{\partial f}{\partial \theta} \right\| \|\Delta\theta\| \triangleq \delta \quad (14)$$

$$\epsilon_{t+1} = \|h_{t+1} - h_t\| \leq \text{Lip}(M)\epsilon_t + \delta \quad (15)$$

From Theorem 1, the Lipschitz constants satisfy:

$$\text{Lip}(M_{\text{skip}}) = \sigma_{\max}(J^{\text{skip}}) < \gamma^L = \text{Lip}(M_{\text{vanilla}}) \quad (16)$$

The perturbation bounds inherit:

$$\delta_{\text{skip}} = C_{\text{skip}}\|\Delta\theta\| < C_{\text{vanilla}}\|\Delta\theta\| = \delta_{\text{vanilla}} \quad (17)$$

where $C.$ bounds the parameter-to-output Jacobian.

$$\left\| \frac{\partial f_{\text{skip}}}{\partial \theta} \right\| = \left\| \prod_{l=1}^{L} \frac{\partial h_l}{\partial \theta} \right\| \leq \prod_{l=1}^{L} \left\| \frac{\partial h_l}{\partial \theta} \right\| = \prod_{l=1}^{L} \left[(1-\alpha)\gamma + \alpha\gamma^{2l-L}\right] \cdot C_{\text{base}} < \prod_{l=1}^{L} \gamma \cdot C_{\text{base}} = C_{\text{vanilla}}$$

For T-step feature reuse, cumulative error develops as:

$$\epsilon_T \leq \frac{\text{Lip}(M)^T - 1}{\text{Lip}(M) - 1}\delta \quad (18)$$

Given $\sigma_{\max}(J^{\text{skip}}) < \gamma^L$ and $\delta_{\text{skip}} < \delta_{\text{vanilla}}$, the maximal interval τ satisfies:

$$\frac{\sigma_{\max}(J^{\text{skip}})^\tau - 1}{\sigma_{\max}(J^{\text{skip}}) - 1}\delta_{\text{skip}} = \frac{\gamma^{L\tau} - 1}{\gamma^L - 1}\delta_{\text{vanilla}} = \epsilon_{\max} \quad (19)$$

Solving equation 19 yields $\tau_{\text{skip}} > \tau_{\text{vanilla}}$ under identical $\epsilon_{\max}$, proving Skip-DiT permits larger reuse intervals. □

Table 9. Comparison of training efficiency between DiT-XL and `Skip-DiT` . Images are generated with a 250-step DDPM solver. The term *cfg* refers to classifier-free guidance scales, where metrics for *cfg=1.0* are computed without classifier-free guidance. The best metrics are highlighted in **bold**. `Skip-DiT` significantly exceeds DiT-XL/2 with much less training steps.

Model	Steps	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
DiT-XL/2	7000k	9.49	7.17	122.49	0.67	0.68
Skip-DiT	400k	13.46	5.83	87.45	0.67	0.63
	800k	10.13	5.87	108.28	0.68	0.65
	1600k	9.07	6.26	119.38	0.68	0.67
	2200k	8.59	6.41	124.74	0.68	0.67
	2500k	8.41	6.30	125.16	0.68	0.67
	2900k	8.37	6.50	127.63	0.68	0.68

9. Class-to-image Generation Experiments

Peebles and Xie [22] proposed the first diffusion model based on the transformer architecture, and it outperforms all prior diffusion models on the class conditional ImageNet [7] 512×512 and 256×256 benchmarks. We add skip connections to its largest model, DiT-XL, to get `Skip-DiT`. We train `Skip-DiT` on class conditional ImageNet with resolution 256×256 from scratch with completely the same experimental settings as DiT-XL, and far exceeds DiT-XL with only around 38% of its training cost.

Training of `Skip-DiT` We add long-skip-connections in DiT-XL and train `Skip-DiT` for 2.9M steps on 8 A100 GPUs, compared to 7M steps for DiT-XL, which also uses 8 A100 GPUs. The datasets and other training settings remain identical to those used for DiT-XL, and we utilize the official training code of DiT-XL[†]. The performance comparison is presented in Table 9, which demonstrates that `Skip-DiT` significantly outperforms DiT-XL while requiring only 23% of its training steps (1.6M vs. 7M), highlighting the training efficiency and effectiveness of `Skip-DiT`.

Accelerating Evaluation We evaluate `Skip-DiT` and compare its performance against two other caching methods: Δ-DiT and FORA. As shown in Table 10, `Skip-DiT` achieves a 1.46× speedup with only a minimal FID loss of 0.02 when the classifier-free guidance scale is set to 1.5, compared to the 7–8× larger losses observed with Δ-DiT and FORA. Moreover, even with a 1.9× acceleration, `Skip-DiT` performs better than the other caching methods. These findings further confirm the effectiveness of `Skip-DiT` for class-to-image tasks.

10. Evaluation Details

VBench [10] is a novel evaluation framework for video generation models. It breaks down video generation assess-

[†]https://github.com/facebookresearch/DiT

Table 10. Class-to-image generation performance. The definition of the cache step n follows that in Table 4. Images are generated with a 250-step DDPM solver. Speedups are calculated on an H100 GPU with a sample batch size of 8. $n = i$ indicates caching the high-level features in x_t for reuse during the inference of $x_{t-1}, x_{t-2}, \ldots, x_{t-n+1}$. The term *cfg* refers to classifier-free guidance scales, where metrics for *cfg=1.0* are computed without classifier-free guidance. We highlight baseline DiT models (without caching) in grey and the best metrics in **bold**.

Methods	FID ↓	sFID ↓	IS ↑	Precision%	Recall%	Speedup
cfg=1.5						
DiT-XL	2.30	4.71	276.26	82.68	57.65	1.00×
FORA	2.45	5.44	265.94	81.21	58.36	1.57×
Delta-DiT	2.47	5.61	265.33	81.05	58.83	1.45×
Faster-Diff	4.96	10.19	223.21	75.21	**59.28**	1.42×
`Skip-DiT` with Cached Inference						
`Skip-DiT`	2.29	4.58	281.81	82.88	57.53	1.00×
$n = 2$	**2.31**	**4.76**	**277.51**	**82.52**	58.06	1.46×
$n = 3$	2.40	4.98	272.05	82.14	57.86	1.73×
$n = 4$	2.54	5.31	267.34	81.60	58.31	**1.93**×
cfg=1.0						
DiT-XL	9.49	7.17	122.49	66.66	67.69	1.00×
FORA	11.72	9.27	113.01	64.46	67.69	1.53×
Delta-DiT	12.03	9.68	111.86	64.57	67.53	1.42×
Faster-Diff	22.98	18.09	80.41	56.53	67.41	1.42×
`Skip-DiT` with Cached Inference						
`Skip-DiT`	8.37	6.50	127.63	68.06	67.89	1.00×
$n = 2$	**9.25**	**7.09**	**123.57**	**67.32**	67.40	1.46×
$n = 3$	10.18	7.72	119.60	66.53	**67.84**	1.71×
$n = 4$	11.37	8.49	116.01	65.73	67.32	**1.92**×

ment to 16 dimensions from video quality and condition consistency: subject consistency, background consistency, temporal flickering, motion smoothness, dynamic degree, aesthetic quality, imaging quality, object class, multiple objects, human action, color, spatial relationship, scene, temporal style, appearance style, overall consistency.

Peak Signal-to-Noise Ratio (PSNR) measures generated visual content quality by comparing a processed version $\mathbf{v}$ to the original reference $\mathbf{v}_r$ by:

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{R^2}{\texttt{MSE}(\mathbf{v}, \mathbf{v}_r)}\right) \tag{20}$$

where R is the maximum possible pixel value, and $\texttt{MSE}(\cdot, \cdot)$ calculates the Mean Squared Error between original and processed images or videos. Higher PSNR indicates better reconstruction quality. However, PSNR does not always correlate with human perception and is sensitive to pixel-level changes.

Structural Similarity Index Measure (SSIM) is a perceptual metric that evaluates image quality by considering luminance, contrast, and structure:

$$\text{SSIM} = [l(\mathbf{v}, \mathbf{v}_r)]^\alpha \cdot [c(\mathbf{v}, \mathbf{v}_r)]^\beta \cdot [s(\mathbf{v}, \mathbf{v}_r\mathbf{v}_r)]^\gamma \tag{21}$$

where α, β, γ are weights for luminance, contrast, and structure quality, where luminance comparison is $l(x, y) = \frac{2\mu_\mathbf{v}\mu_{\mathbf{v}_r}+C_1}{\mu_\mathbf{v}^2+\mu_{\mathbf{v}_r}^2+C_1}$, contrast comparison is $c(x, y) = \frac{2\sigma_\mathbf{v}\sigma_{\mathbf{v}_r}+C_2}{\sigma_\mathbf{v}^2+\sigma_{\mathbf{v}_r}^2+C_2}$, and structure comparison is $s(x, y) = \frac{\sigma_{xy}+C_3}{\sigma_\mathbf{v}\sigma_{\mathbf{v}_r}+C_3}$, with C denoting numerical stability coefficients. SSIM scores range from -1 to 1, where 1 means identical visual content.

Learned Perceptual Image Patch Similarity (LPIPS) is a deep learning-based metric that measures perceptual similarity using L2-Norm of visual features $v \in \mathbb{R}^{H\times W\times C}$ extracted from pretrained CNN $\mathcal{F}(\cdot)$. LPIPS captures semantic similarities and is therefore more robust to small geometric transformations than PSNR and SSIM.

$$\text{LPIPS} = \frac{1}{HW}\sum_{h,w} ||\mathcal{F}(v_r) - \mathcal{F}(v)||_2^2 \tag{22}$$

Fréchet Inception Distance (FID) and Fréchet Video Distance (FVD) FID measures the quality and diversity of generated images by computing distance between feature distributions of reference $\mathcal{N}(\mu_r, \Sigma_r)$ and generated images $\mathcal{N}(\mu, \Sigma)$ using inception architecture CNNs, where μ, Σ are mean and covariance of features.

$$\text{FID} = ||\mu_r - \mu||^2 + Tr(\Sigma_r + \Sigma - 2(\Sigma_r\Sigma)^{1/2}) \tag{23}$$

FVD is a video extension of FID. Lower FID and FVD indicate higher generation quality.

Table 11. Comparison of our caching method with the faster sampler DDIM. `Skip-Cache` is evaluated with a 250-steps DDPM and compared to the DDIM sampler under similar throughput. Baseline DiT models (without caching) are highlighted in grey, and the best metrics are indicated in **bold**. Notably, DDIM outperforms the 250-step DDPM in the UCF101 task for both Latte and `Skip-DiT` . `Skip-Cache` denotes `Skip-DiT` with cached inference.

Method	UCF101		FFS		Sky		Taichi	
	FVD ↓	FID ↓	FVD ↓	FID ↓	FVD ↓	FID ↓	FVD ↓	FID ↓
Latte	165.04	23.75	28.88	5.36	49.46	11.51	166.84	11.57
`Skip-DiT`	173.70	22.95	20.62	4.32	49.22	12.05	163.03	13.55
$\texttt{Skip-DiT}_{n=2}$	165.60	**22.73**	**23.55**	**4.49**	**51.13**	**12.66**	**167.54**	**13.89**
DDIM+`Skip-DiT`	**134.22**	24.60	37.28	6.48	86.39	13.67	343.97	21.01
DDIM+`Latte`	146.78	23.06	39.10	6.47	78.38	13.73	321.97	21.86
$\texttt{Skip-DiT}_{n=3}$	169.37	**22.47**	**26.76**	**4.75**	**54.17**	**13.11**	**179.43**	**14.53**
DDIM+`Skip-DiT`	**139.52**	24.71	39.20	6.49	90.62	13.80	328.47	21.33
DDIM+`Latte`	148.46	23.41	41.00	6.54	74.39	14.20	327.22	22.96

11. Scheduler Selection for text-to-video Tasks

For the text-to-video generation task, all the videos generated for evaluation are sampled with 50 steps DDIM [33], which is the default setting used in Latte. In the class-to-video

generation tasks, vanilla Latte uses 250-step DDPM [8] as the default solver for class-to-video tasks, which we adopt for all tasks except UCF101. For UCF101, we employ 50-step DDIM [33], as it outperforms 250-step DDPM on both Latte and `Skip-DiT`. Table 11 highlights this phenomenon, showing our methods consistently outperform DDPM-250 under comparable throughput, except for UCF101, where DDIM performs better than 250 steps DDPM. In the text-to-image task, we choose 50-step DDIM for sampling, and for the class-to-image task, we choose 250-steps DDPM.

12. Implementation of Other Caching Methods

DeepCache DeepCache [17] is a training-free caching method designed for U-Net-based diffusion models, leveraging the inherent temporal redundancy in sequential denoising steps. It utilizes the skip connections of the U-Net to reuse high-level features while updating low-level features efficiently. `Skip-DiT` shares significant similarities with DeepCache but extends the method to DiT models. Specifically, we upgrade traditional DiT models to `Skip-DiT` and cache them using `Skip-DiT` . In the work of DeepCache, two key caching decisions are introduced: (1) `N`: the number of steps for reusing cached high-level features. Cached features are computed once and reused for the next `N-1` steps. (2) The layer at which caching is performed. For instance, caching at the first layer ensures that only the first and last layers of the U-Net are recomputed. In `Skip-DiT`, we adopt these two caching strategies and additionally account for the timesteps to cache, addressing the greater complexity of DiT models compared to U-Net-based diffusion models. For all tasks except the class-to-image task, caching is performed at the first layer, whereas for the class-to-image task, it is applied at the third layer.

Δ-DiT Δ-DiT [6] is a training-free caching method designed for image-generating DiT models. Instead of caching the feature maps directly, it uses the offsets of features as cache objects to preserve input information. This approach is based on the observation that the front blocks of DiT are responsible for generating the image outlines, while the rear blocks focus on finer details. A hyperparameter b is introduced to denote the boundary between the outline and detail generation stages. When $t \leq b$, Δ-Cache is applied to the rear blocks; when $t > b$, it is applied to the front blocks. The number of cached blocks is represented by N_c. While this caching method was initially designed for image generation tasks, we extend it to video generation tasks. In video generation, we observe significant degradation in performance when caching the rear blocks, so we restrict caching to the front blocks during the outline generation stage. For Hunyuan-DiT [14], we cache the middle blocks due to the U-shaped transformer architecture. Detailed configurations are provided in Table 12.

Table 12. Configuration details for Δ-Cache in different models and tasks. *t2v* denotes text-to-video, *c2v* denotes class-to-video, *t2i* denotes text-to-image, and *c2i* denotes class-to-image.

Δ-DiT	Task	Diffusion steps	b	All layers	N_c
Latte	*t2v*	50	12	28	21
Latte	*c2v*	250	60	14	10
Hunyuan	*t2i*	50	12	28	18
DiT-XL/2	*c2i*	250	60	28	21

PAB PAB (Pyramid Attention Broadcast) [53] is one of the most promising caching methods designed for real-time video generation. The method leverages the observation that attention differences during the diffusion process follow a U-shaped pattern, broadcasting attention outputs to subsequent steps in a pyramid-like manner. Different broadcast ranges are set for three types of attention—spatial, temporal, and cross-attention—based on their respective differences. $PAB_{\alpha\beta\gamma}$ denotes the broadcast ranges for spatial (α), temporal (β), and cross (γ) attentions.

In this work, we use the official implementation of PAB for text-to-video tasks on Latte and adapt the caching method to other tasks in-house. For the class-to-video task, where cross-attention is absent, $PAB_{\alpha\beta}$ refers to the broadcast ranges of spatial (α) and temporal (β) attentions. In the text-to-image task, which lacks temporal attention, $PAB_{\alpha\beta}$ instead denotes the broadcast ranges of spatial (α) and cross (β) attentions. We do not apply PAB to the class-to-image task, as it involves only spatial attention.

Table 13. Configuration details for T-GATE in different settings. *t2v* denotes text-to-video, *t2i* denotes text-to-image.

T-GATE	Task	Diffusion steps	m	k
Latte	*t2v*	50	20	2
Hunyuan-DiT	*t2i*	50	20	2

T-Gates T-Gates divide the diffusion process into two phases: (1) the Semantics-Planning Phase and (2) the Fidelity-Improving Phase. In the first phase, self-attention is computed and reused every k step. In the second phase, cross-attention is cached using a caching mechanism. The hyperparameter m determines the boundary between these two phases. For our implementation, we use the same hyperparameters as PAB [53]. Detailed configurations are provided in Table 13.

FORA FORA (Fast-Forward Caching) [30] stores and reuses intermediate outputs from attention and MLP layers across denoising steps. However, in the original FORA paper, features are cached in advance before the diffusion process. We do not adopt this approach, as it is a highly time-consuming process. Instead, in this work, we skip the "Initialization" step in FORA and calculate the features dynamically during the diffusion process.

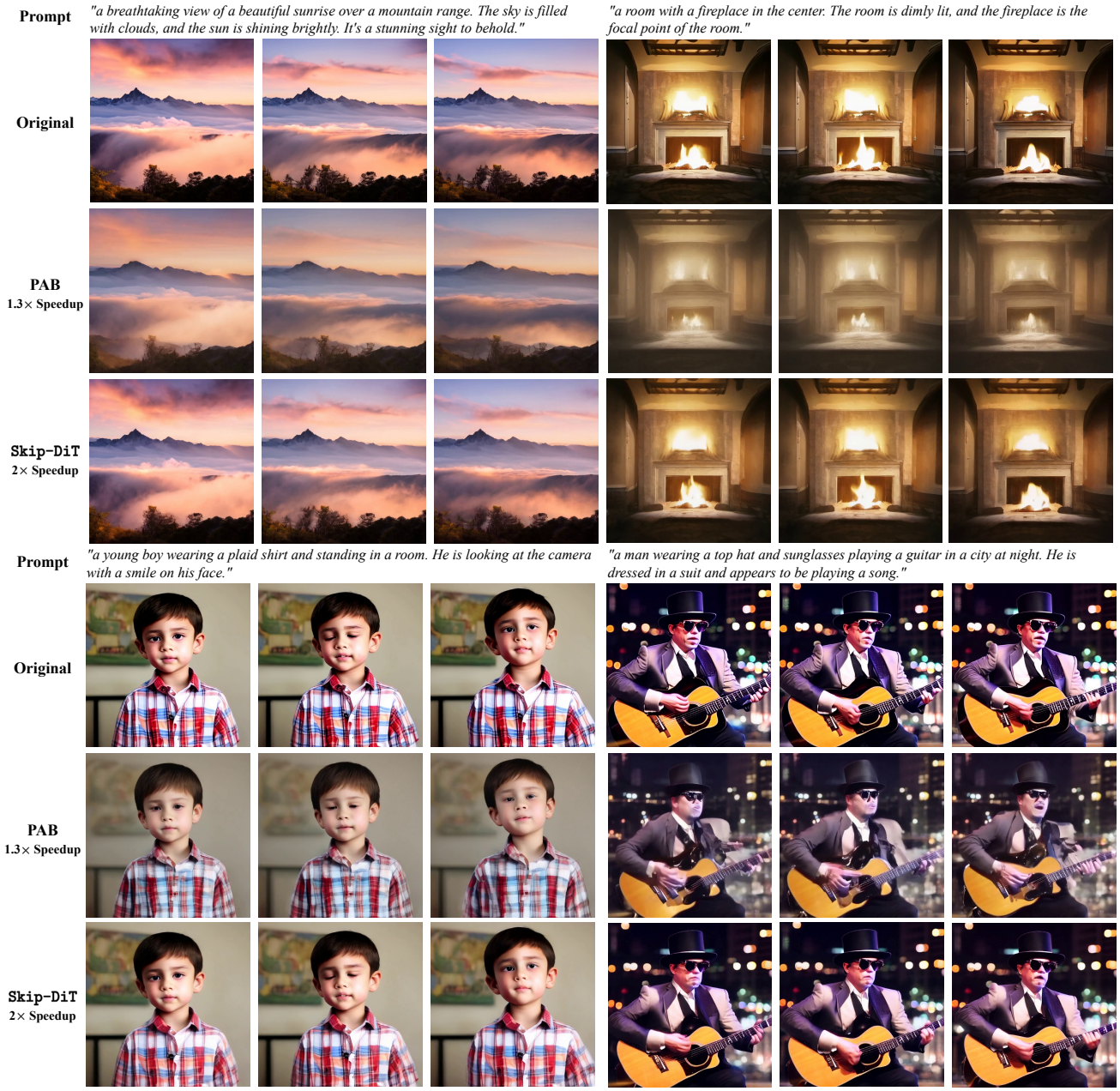

Figure 6. Qualitative results of text-to-video generation. We present `Skip-DiT` , PAB_{469}, and the original model. The frames are randomly sampled from the generated video.

AdaCache AdaCache [11] identifies the feature sample-variance in the DiT, and proposes to predict the feature similarity of the current timestep to decide whether to cache. The codebook for Latte, which records the threshold is not released. We provide a version of codebook which almost reproduce the performance in the paper. The codebook is as follows: $\{0.08 : 3, 0.10 : 2, 0.12 : 1\}$

TeaCache Same as AdaCache, TeaCache[16] also discovers the feature instability in DiT. Different from AdaCache, TeaCache finds the relationship between input and output similarity and employ high-order polynomial functions to predict the similarity of the current timestep. We use the official codebase of it and choose the slow-caching strategy to evaluate its upper-bound.

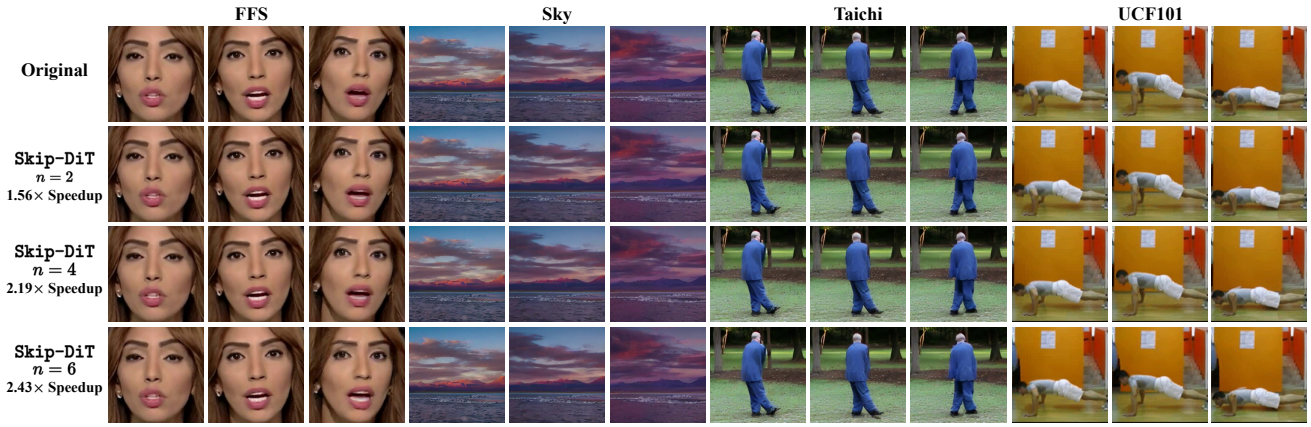

Figure 7. Qualitative results of class-to-video generation. We present the original video generation model and Skip-DiT with different caching steps n. The frames are randomly sampled from the generated video.

Figure 8. Qualitative results of text-to-image generation. We present Skip-DiT , Δ-DiT, FORA, T-GATE, and the original model.

13. Case Study

Video Generation In Figure 6, we showcase the generated video frames from text prompts with Skip-DiT , PAB, and comparing them to the original model. From generating portraits to scenery, Skip-DiT with caching consistently demonstrates better visual fidelity along with faster generation speeds. Figure 7 presents class-to-video generation examples with Skip-DiT with varying caching steps $\in \{2, 4, 6\}$. By comparing the output of Skip-DiT with cache to standard output, we see Skip-DiT maintains good generation quality across different caching steps.

Image Generation Figure 8 compares qualitative results of Skip-DiT compared to other caching-based acceleration methods (Δ-DiT, FORA, T-GATE) on Hunyuan-DiT. In Figure 9, Skip-DiT show distinct edges in higher speedup and similarity to the original generation, while other baselines exist with different degrees of change in details such as color, texture, and posture. Similarly, we present Skip-DiT with varying caching steps in Figure 9, showing that with more steps cached, it still maintains high fidelity to the original generation.

Figure 9. Qualitative results of class-to-image generation. We present the original image generation model and `Skip-DiT` with different caching steps n.