OmniCache: A Trajectory-Oriented Global Perspective on Training-Free Cache Reuse for Diffusion Transformer Models

Supplementary Material

1. Pseudocode of OmniCache

Algorithm 1 Calibration Stage of OmniCache

```
Require: Noisy input $x_T$, model $\epsilon_\theta$
Ensure: reuse set $\mathcal{S}$, coeffs $\gamma_t$
1: for $t = T : 1$ do
2:     $q \leftarrow \widetilde{\epsilon}_\theta(x_t, t) - \epsilon_\theta(x_t, t)$
3:     Save $q$ and $x_t$
4: end for
5: get trajectory/curvature through saved $x_t$
6: get coeffs $\gamma_t$ through saved $q$
7: get reuse set $\mathcal{S}$
```

Algorithm 2 Inference Stage of OmniCache

```
Require: Noisy input $x_T$, model $\epsilon_\theta$, reuse set $\mathcal{S}$, coeffs $\gamma_t$
Ensure: Denoised output $x_0$
1: for $t = T : 1$ do
2:     if $t \in \mathcal{S}$ then
3:         $q \leftarrow \widetilde{\epsilon}_\theta(x_{t+1}, t+1) - \epsilon_\theta(x_{t+1}, t+1)$ {Both are pre-stored on step $t+1$}
4:         $\epsilon_\theta(x_t, t) \leftarrow \widetilde{\epsilon}_\theta(x_t, t) - \gamma_{t-1}\,\mathrm{Filter}(q, t)$ {Filter: LowPass if early, else HighPass}
5:     end if
6:     Normal Inference Step
7: end for
8: return $x_0$
```

OmniCache operates in two stages: a calibration stage and an inference stage.

1.1. Calibration Stage

As shown in Alg. 1, we store, for various input examples, the hidden states x_t at different timesteps along with the noise introduced by cache reuse. Based on these pre-stored $\{x_t\}$, we reconstruct the diffusion model's full sampling trajectory and compute its local curvature. We then select those sampling steps whose omission has the least effect on the trajectory; at exactly these steps we perform cache reuse in order to accelerate inference. Moreover, since the noise induced by cache reuse at adjacent timesteps tends to be correlated, we use the pre-stored noise q to estimate an inter-step noise correlation coefficient γ_t, which will later guide our noise-correction procedure.

1.2. Inference Stage

During inference, we apply cache reuse at the predetermined "reuse set" $\mathcal{S}$. Our cache reuse occurs at each DIT blocks. Regarding the OmniCache inference process, as shown in Alg. 2, we perform two forward passes for each non-skipped step t: 1) One actual forward pass, 2) One forward pass with cache reuse at step t. This dual-pass approach allows us to compute the noise q_t introduced by cache reuse at step t. When we intend to skip the subsequent step t-1, we estimate q_{t-1} using q_t and the correlation γ_t, thereby deriving the corrected output for step t-1.

2. Latency

In Tab 1, we report the wall-time breakdown of OpenSora during inference. Out of 30 sampling steps, OmniCache eliminates 15 steps of computation; unlike ToCa and similar methods, we do not rely on any sparsity-based operations. The additional overhead from noise correction and high-/low-pass filtering is negligible compared to a standard sampling step.

3. Visualization on OpenSora

As shown in Fig. 1, our visual comparison on OpenSora reveals that the current SOTA method, TeaCache, produces outputs with noticeable deviations from the original model, such as the notebook in the third row and the water cup in the fifth row. In contrast, our method generates results that are essentially consistent with the original model while achieving a 2.5× acceleration.

4. Visualization on Latte

As shown in Fig. 2, we conducted a visual comparison on Latte. TeaCache exhibits many abnormal distortions—for example, the pear in the last row and the teddy bear in the fifth row. In contrast, our generated videos are more consistent with the original model, and the objects in the videos appear more natural.

Table 1. Latency Comparison in inference stage.

Latency	Original (30 Steps)	OmniCache-Slow				
		Real Inference Steps (15)	Cache Reuse Steps (15)	High/Low pass Filtering	Noise Correction	All Inference Time
OpenSora	25.72 s	12.645 s	0.15 s	$8.4e^{-3}$ s	$3e^{-3}$ s	**12.80 s**

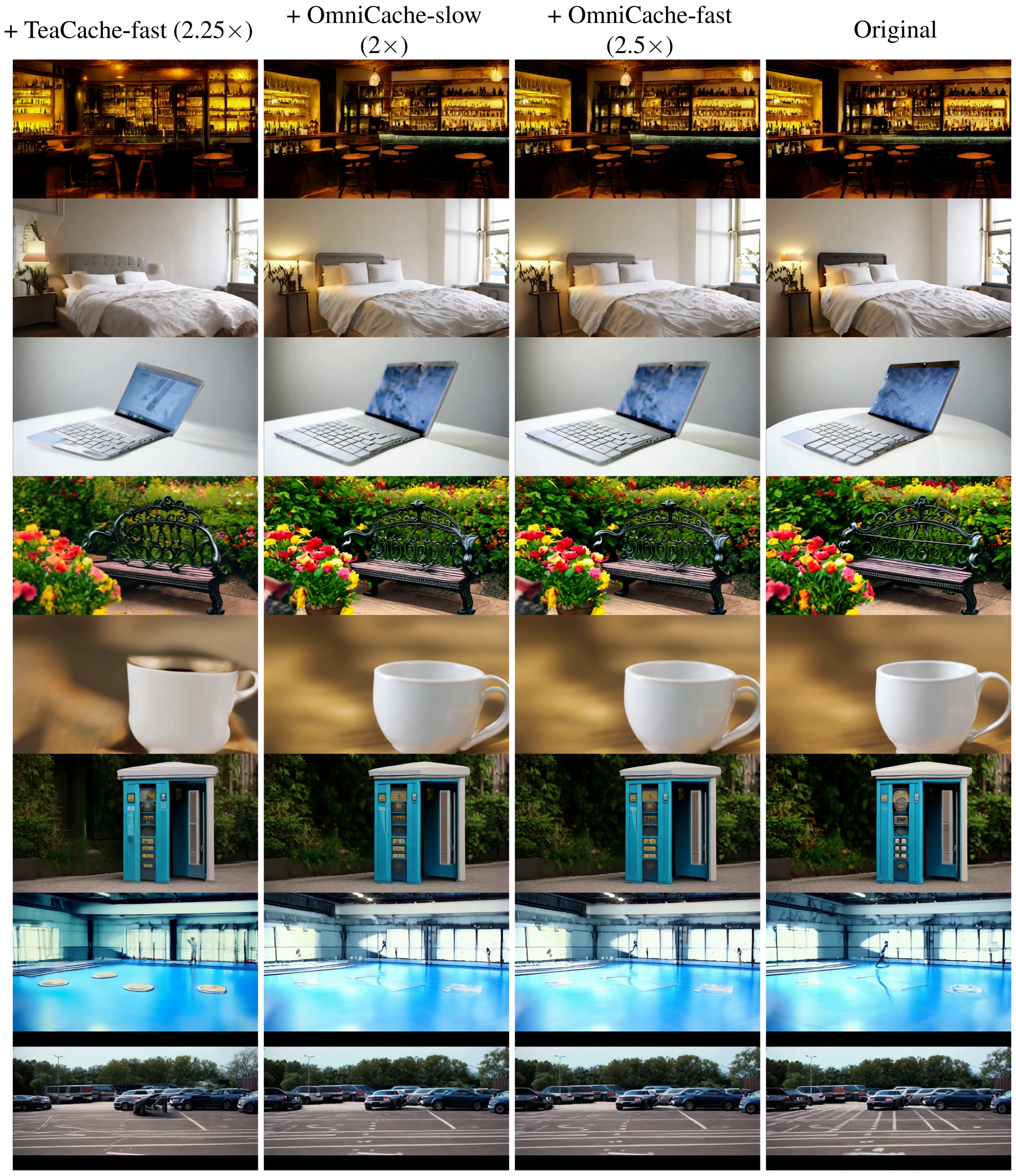

Figure 1. First-Frame Visualization of the Output Video on OpenSora V1.2 (480P, 2s at 30 Steps)

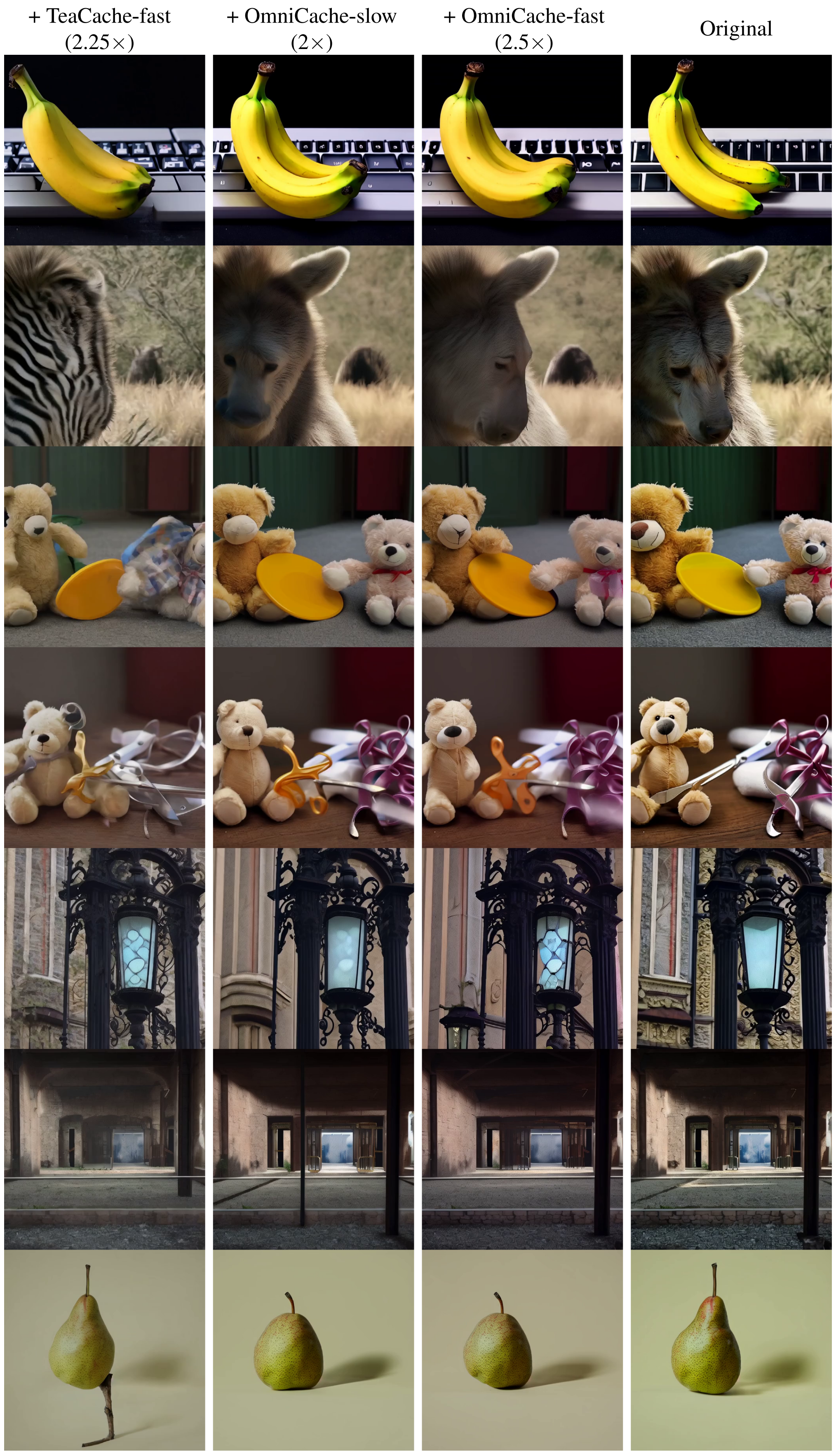

Figure 2. First-Frame Visualization of the Output Video on Latte (512×512, 2s at 50 Steps)