OTCache: Optimal Transport for Geometry-Aware Caching in Diffusion Models

Huanlin Gao[1,2†], Fang Zhao[1,2†], Qiang Hui[1,2], Fuyuan Shi[1,2], Shaoan Zhao[1,2], Yantao Li[1,2,3], Chao Tan[1,2], Ting Lu[1,2], Yuren You[2], Kai Wang[1,2*], and Shiguo Lian[1,2*]

[1] Data Science & Artificial Intelligence Research Institute, China Unicom
[2] Unicom Data Intelligence, China Unicom
[3] National Key Laboratory for Novel Software Technology, Nanjing University
{gaohl51,zhaof50,wangk115,liansg}@chinaunicom.cn
[†]Equal contribution. [*]Corresponding authors.

Abstract. We propose **OTCache**, a training-free framework for accelerating diffusion sampling via caching schedule prediction. Existing graph-based caching methods reduce redundant computation by optimizing shortest-path objectives, but rely on an additive independence assumption, which often breaks down in the low NFE regime. To address this issue, OTCache models caching schedules across inference budgets as a smooth evolution in policy space, inspired by Optimal Transport (OT). The framework consists of three stages: (1) obtaining a high-fidelity **reference schedule** using a graph-based caching method under a conservative budget; (2) performing a lightweight **anchor search** under an extreme low-budget setting via Optuna optimization with an end-to-end perceptual objective; and (3) predicting schedules for target budgets via **quantile interpolation** between the reference and anchor policies using continuous warping representations. Experiments on FLUX.1 [dev], Qwen-Image, and HunyuanVideo show that OTCache achieves 4.5×, 4.7×, and 3.66× acceleration, respectively, while consistently improving generation fidelity over state-of-the-art caching baselines. This work provides a new perspective on accelerating diffusion models through Optimal-Transport-inspired schedule modeling. **Code:** `https://github.com/UnicomAI/OTCache`

Keywords: Diffusion Models · Graph-based Caching · Optimal Transport

1 Introduction

Flow Matching (FM) [2, 22] has emerged as a cornerstone of modern generative modeling, driving significant breakthroughs in high-fidelity synthesis across image [20], video [19, 52], and multi-modal tasks [15]. By modeling continuous transport paths via instantaneous velocity fields, FM provides a mathematically principled and effective paradigm for sampling. However, in commercial-scale models such as FLUX.1 [20] and HunyuanVideo [19], the massive parameter

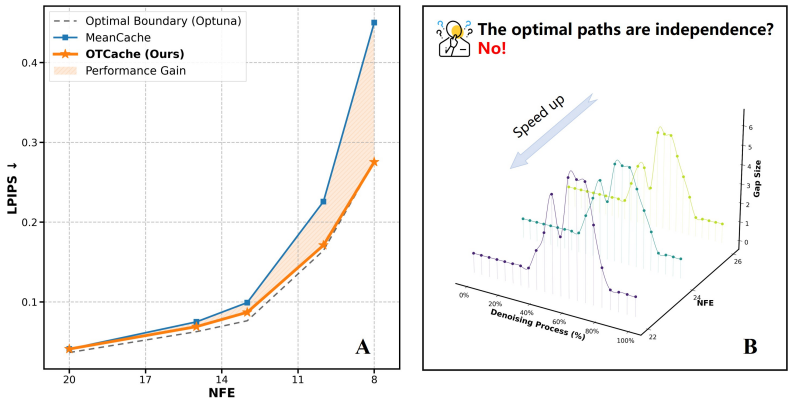

Fig. 1: Performance gain and structural insights on FLUX.1 [dev]. (A) Failure of additive surrogates: MeanCache deviates from the optimal boundary in the ultra-low NFE regime, reflecting the limitations of local error aggregation. OTCache recovers this fidelity loss, significantly narrowing the gap to the search-based optimum. **(B) Structural regularity of optimal paths:** Optimal schedules across different NFE (budgets) demonstrate strong structural relationships rather than independent patterns, suggesting a shared underlying policy trajectory that OTCache exploits via geometric interpolation.

scale of transformer-based denoisers coupled with the prolonged iterative sampling process leads to staggering computational overhead and memory footprints. This **computation-intensive** nature creates a formidable deployment barrier, significantly hindering their applicability in interactive or resource-constrained scenarios.

To alleviate this burden, established acceleration strategies such as distillation [35, 36], pruning [11], and quantization [21] have been widely explored. Nevertheless, most of these methods impose a heavy "adoption tax"—they typically necessitate intensive retraining on large-scale datasets, complex architectural modifications, or sophisticated engineering pipelines, which limits their flexibility for rapid deployment. In contrast, **caching techniques** [30, 45] have emerged as a compelling training-free alternative.

Recent advancements such as MeanCache [9] formulate cache scheduling as a constrained shortest-path problem, enabling efficient, training-free planning under a fixed NFE budget. However, this paradigm relies on an *additive independence* assumption, approximating end-to-end degradation as a sum of local edge errors. We find that this surrogate becomes increasingly unreliable in the low-NFE regime. As shown in Fig. 1A, when NFE decreases from 20 to 8, MeanCache still exhibits a clear gap to the search-based optimum (Optuna), indicating substantial room for improvement in LPIPS.

More fundamentally, Fig. 1B reveals an overlooked structural regularity: the gap profiles (first-order differences) of optimal schedules across budgets are **not** unrelated; instead, their temporal structures evolve smoothly as the NFE budget varies. This observation motivates a shift in perspective, rather than independently solving a discrete shortest-path problem for each budget, we model

budget-conditioned schedule evolution and exploit structural relationships across budgets.

Based on this insight, we propose **OTCache**, a training-free framework that predicts schedules for Low-NFEs via optimal transport interpolation. By anchoring prediction with a reliable high-budget reference and a low-budget anchor, and interpolating between them in Wasserstein space, OTCache generates robust schedules that remain stable under Low-NFE acceleration. The main contributions of this work are summarized as follows:

- **Rethinking Cache Scheduling under Low NFE.** We identify two fundamental limitations of existing graph-based schedulers: (i) surrogate misalignment caused by additive shortest-path objectives under nonlinear error propagation, and (ii) the independent-budget assumption that ignores structural relationships across optimal schedules.
- **Budget-Conditioned Schedule Evolution.** We introduce a three-stage acceleration framework that treats cache schedules as probability measures and models their evolution across NFEs as a smooth trajectory in policy space. Leveraging the geometric structure of Optimal Transport (OT), we obtain target-budget schedules through quantile interpolation between two reliable endpoints.
- **Outstanding Performance.** Experiments on FLUX.1, Qwen-Image, and HunyuanVideo demonstrate that OTCache achieves 4.5×, 4.7×, and 3.66× acceleration, respectively. In particular, on Qwen-Image, OTCache achieves an LPIPS of **0.171**.

2 Related Work

2.1 Diffusion Model Acceleration.

The remarkable success of generative models [38, 40] across diverse modalities is significantly attributed to the continuous advancements in sampling speed. Early efforts primarily focused on optimizing the iterative denoising process through principled numerical SDE/ODE solvers [5, 16, 40], such as DDIM [38], EDM [18], and DPM-Solver [28], which aim to maintain high synthesis quality with fewer discretization steps. To further compress inference trajectories, knowledge distillation [13] has been extensively explored to map multi-step denoising into compact few-step or even single-step regimes, exemplified by Progressive Distillation [35] and Consistency Models [36, 39, 43]. Complementary to these, orthogonal strategies—including quantization [21, 37], pruning [31], and system-level parallelization frameworks [6, 50]—have been investigated to enhance raw hardware throughput. More recently, the emergence of Flow Matching (FM) [2, 22] has introduced a new paradigm that learns deterministic velocity fields, inherently possessing the potential for high-fidelity generation with minimal sampling steps. Nevertheless, most existing methods still require heavy computation, large-scale data, or complex engineering, limiting their practical adoption in resource-constrained scenarios.

2.2 Cache in Diffusion Models.

As a training-free acceleration paradigm, caching strategies [29, 45] have gained prominence by reusing intermediate representations to bypass redundant computation. Early methods such as DeepCache [30] introduced architecture-specific feature reuse for UNet backbones, while T-GATE [49] and Δ-DiT [4] extended this idea to Transformer-based architectures [33]. For large-scale video generation, PAB [51] and TeaCache [23] exploit temporal correlations and error thresholds to trigger feature reuse. Recent studies further explore fine-grained caching criteria, including adaptive reuse for video diffusion transformers [17], profiling-based cache selection [32], frequency-aware caching [24], speculative feature caching [26], and search-based policy discovery [1]. The field has also evolved towards graph-based caching; LeMiCa [8] abstracts video synthesis into a Directed Acyclic Graph (DAG) for global scheduling, while MeanCache [9] draws inspiration from MeanFlow [10] and reformulates caching from an instantaneous velocity view to an average velocity perspective, stabilizing the sampling trajectory. Despite these advances, existing methods typically optimize schedules within a fixed budget or rely on local surrogate criteria. How to obtain accurate *optimal* caching policies in the extreme high-acceleration regime, namely the low-NFE setting, remains an open problem.

3 Methodology

3.1 Preliminaries

Flow Matching. Flow Matching [2,22] and Rectified Flow [27] introduce a new paradigm for diffusion-based generative modeling by constructing continuous transport paths between a noise distribution π_1 and a data distribution π_0. By defining a probability path via linear interpolation:

$$x_t = (1-t)x_0 + tx_1, \quad t \in [0,1], \tag{1}$$

these methods aim to learn a straight-line trajectory in the state space. Since the clean data x_0 is unknown during the denoising (generation) process, a learned velocity field $v_\theta(x_t, t)$ is employed to approximate the direction $(x_0 - x_1)$, thereby constructing a neural ODE model:

$$d\hat{x}_t = v_\theta(x_t, t)dt \tag{2}$$

Numerical solvers discretize this ODE into N steps to recover the data distribution. The total computational cost is primarily determined by the Number of Function Evaluations (NFE), making efficient step-wise scheduling critical for inference acceleration.

Graph-based Cache Scheduling Feature caching has emerged as a predominant training-free paradigm to accelerate inference by reusing intermediate states

across adjacent timesteps. Recent state-of-the-art methods [8, 9] formulate this as a constrained shortest-path problem on a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In this formulation, each node $v \in \mathcal{V}$ corresponds to a discrete timestep, and an edge $e = (t \to s)$ represents a caching transition where the velocity field at s is estimated using cached features from t. The fidelity loss is quantified by edge weights:

$$\mathcal{L}(t \to s) = \|v(x_s, s) - \hat{v}(x_s, s; x_t)\|_p, \tag{3}$$

where $\hat{v}$ denotes the cached estimator. The global scheduling problem aims to find an optimal path $\pi^\star$ that minimizes the cumulative error:

$$\pi^\star = \arg\min_{\pi \in \mathcal{P}} \sum_{e \in \pi} \mathcal{L}(e)^\gamma, \quad \text{s.t. } |\pi| \le \mathcal{B}, \tag{4}$$

where $\mathcal{P}$ is the set of feasible paths and $\mathcal{B}$ is the computation budget.

3.2 Rethink: Beyond Additive Shortest-Path Surrogates

Limitation. Graph-based schedulers formulate caching as a constrained shortest-path problem (Eq. 4). In this formulation, the caching cost between any two timesteps (t, s) is first estimated independently as an edge weight $\mathcal{L}(e)$, and the optimal schedule is obtained by minimizing the sum of these edge costs. This implicitly assumes *additive independence*: the end-to-end degradation of a schedule can be approximated by aggregating independently estimated edge losses.

This approximation is generally reliable in the **high-NFE regime**, where caching intervals are short and each decision has limited influence on subsequent steps. However, in the **low-NFE regime**, both the number of caching operations and the spacing between cached states increase. Consequently, the effect of earlier caching decisions propagates further along the trajectory and interacts with later ones, violating the additive independence assumption. As a result, the shortest-path surrogate becomes increasingly inaccurate for estimating the true path degradation. Consistent with this observation, Fig. 1A shows that the performance gap between MeanCache and the optimal boundary grows as NFE decreases.

Observation. Despite this limitation, we observe a clear structural regularity: near-optimal schedules across different NFE budgets are *not independent*. As shown in Fig. 1B, the temporal gap patterns of optimal schedules evolve smoothly as the budget varies. This suggests that optimal schedules under different NFEs follow a structured evolution rather than forming unrelated solutions.

Core Insight. We hypothesize that optimal schedules under different NFE budgets correspond to different-resolution observations of a shared *ideal trajectory* in policy space. Although the observation resolution (NFE) changes, the underlying trajectory remains fixed. Under this perspective, schedules across budgets form a smooth evolution on a policy manifold, which we interpret as a *policy*

geodesic. This view motivates predicting schedules through geometric interpolation in strategy space, which we instantiate using optimal transport in the following section.

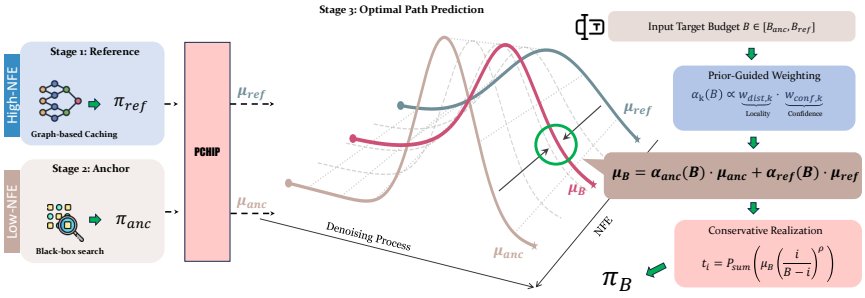

Fig. 2: Overview of OTCache. Stage 1: use graph-based caching methods to obtain a reliable high-NFE(budget) schedule as the reference policy. Stage 2: perform black-box search to find a near-optimal low-NFE schedule as the anchor policy. Stage 3: inspired by optimal transport (OT), convert both endpoint schedules into continuous warping curves via PCHIP and apply quantile interpolation to predict the target-budget schedule π_B.

3.3 Overview of OTCache

Motivated by the observations in Sec. 3.2, we propose **OTCache**, a training-free, three-stage framework for accelerating diffusion sampling via caching schedule prediction, as illustrated in Fig. 2.

Stage 1: Reference Empirically, we find that graph-based caching methods (e.g., MeanCache) may fail to recover the globally optimal schedule in the high-acceleration regime (low-NFE regime), where large integration steps can break the validity of additive shortest-path surrogates. In contrast, under a conservative budget, discretization errors are sufficiently mild such that the surrogate objective aligns well with end-to-end generation fidelity. Throughout this work, we adopt a conservative reference budget.

We obtain a reference schedule using a generic graph-based caching method under this conservative setting. Formally, we denote the resulting schedule as the reference policy:

$$\pi_{\text{ref}} \;\coloneqq\; \mathcal{C}_{\text{graph}}(B_{\text{ref}})\,. \tag{5}$$

Here $\mathcal{C}_{\text{graph}}(\cdot)$ denotes a generic graph-based caching method; in our experiments, we instantiate it with MeanCache. We treat π_{ref} as a high-fidelity reference schedule in the conservative regime, which provides a structural prior for the subsequent budget-conditioned schedule prediction.

Stage 2: Anchor In the low-NFE regime, MeanCache can be unreliable because its shortest-path formulation relies on the *additive independence* assumption, which may break down under large integration steps. As a result, the path minimizing the additive surrogate can be misaligned with true end-to-end generation fidelity. To obtain an accurate low-budget boundary condition for subsequent schedule prediction, we introduce an *anchor* schedule by directly optimizing an end-to-end perceptual objective under an extreme budget B_{anc} via black-box search.

End-to-end objective. Let $x_0(\pi)$ denote the generated sample obtained by executing a candidate schedule π, and x_0 denote the full-budget reference output obtained once by running the original (non-cached) sampling path with all steps evaluated under the same prompt and seed. We define the anchor objective as the perceptual discrepancy between the two outputs:

$$\mathcal{J}(\pi) \triangleq \ell_{\text{LPIPS}}(x_0(\pi), x_0)\,. \tag{6}$$

This objective is defined per prompt-seed pair. Unlike MeanCache, which optimizes a surrogate based on averaged velocity discrepancies, and LeMiCa, which optimizes latent-space deviations, we directly optimize a perceptual distance (LPIPS [48]) on the final generated image or video outputs, thereby capturing the fidelity gap induced by acceleration in an end-to-end manner.

Black-box search. To obtain an accurate anchor in the low budget regime, we directly minimize the end-to-end objective in Eq. (6) using a lightweight black-box optimizer (Optuna + CMA-ES). We run the search for a fixed number of trials and apply early stopping to cap the cost (i.e., terminate if no improvement is observed for a predefined patience):

$$\pi_{\text{anc}} \coloneqq \operatorname*{arg\,min}_{\pi \in \mathcal{P}_{B_{\text{anc}}}} \mathcal{J}(\pi), \tag{7}$$

where $\mathcal{P}_{B_{\text{anc}}}$ denotes the feasible set under budget B_{anc}.

To improve sample efficiency, we adopt two practical choices. **(i) Warm start.** We initialize the search from the MeanCache schedule at the same budget (e.g., $B_{\text{anc}} = 8$), leveraging a strong prior and providing a conservative fallback when no improvement is found for a given prompt-seed pair. **(ii) First-order parameterization.** We optimize the schedule in a gap (first-order difference) space rather than directly over discrete timesteps, which empirically yields a better-conditioned search landscape while preserving monotonic structure by construction.

Stage 3: Optimal Path Prediction To generalize scheduling patterns across arbitrary budgets, we model their evolution as a continuous trajectory in a functional space.

Continuous Representation via Monotonic Splines. We lift each discrete schedule $\pi_B = \{t_i\}_{i=0}^{B-1}$ to a continuous warping function over a normalized progress domain $u \in [0, 1]$ by assigning uniformly spaced progress coordinates and fitting a shape-preserving Piecewise Cubic Hermite Interpolator (PCHIP). This yields a strictly monotone time-warping curve without spurious oscillations. The curve is then uniformly sampled at a fixed resolution to obtain an equal-dimensional representation. Applying this procedure to π_{ref} and π_{anc} produces the aligned continuous embeddings $(\mu_{\text{ref}}, \mu_{\text{anc}})$, enabling stable optimal-transport interpolation across budgets.

Log-conditioned Weighted Geodesic. To predict the schedule for a target budget $B \in [B_{\text{anc}}, B_{\text{ref}}]$, we model the transition between policies as a smooth path in the space of probability measures. In one dimension, the most natural way to interpolate between two distributions is through the **Wasserstein geodesic**. A key property of this approach is that the interpolation can be performed directly as a linear combination of the **quantile functions** (i.e., our continuous warping curves $\mu(u)$).

Specifically, the predicted schedule $\mu_B(u)$ for target budget B is constructed as:

$$\mu_B(u) = \alpha_{\text{anc}}(B) \cdot \mu_{\text{anc}}(u) + \alpha_{\text{ref}}(B) \cdot \mu_{\text{ref}}(u), \tag{8}$$

where the interpolation weights $\alpha_k(B)$ are defined by:

$$\alpha_k(B) = \frac{w_{\text{dist},k} \cdot w_{\text{conf},k}}{\sum_{j \in \{\text{anc, ref}\}} w_{\text{dist},j} \cdot w_{\text{conf},j}}. \tag{9}$$

Here, the terms $w_{\text{dist},k} = (|B_k - B| + 1)^{-1}$ and $w_{\text{conf},k} = \log B_k$ represent two intuitive priors: (i) **Locality**, which anchors the prediction to the nearest known budget; and (ii) **Confidence**, which assigns higher trust to the high-budget reference since dense schedules provide a more stable structural prior of the ODE trajectory.

Conservative Realization. The discrete schedule $\pi_B = \{t_i\}_{i=0}^{B-1}$ is realized by sampling the continuous geodesic μ_B via a **power-law warping**. Motivated by the observation that early-stage caching requires more conservative updates due to high vector field volatility [8,9], we introduce an exponent $\rho \geq 1.0$ to bias the NFE density toward the start of the reverse ODE:

$$t_i = \mathcal{P}_{\text{sum}}\left(\mu_B\left(\left[\frac{i}{B-1}\right]^{\rho}\right)\right). \tag{10}$$

The projection operator $\mathcal{P}_{\text{sum}}$ rounds the samples to integers and redistributes residuals to the largest sampling gaps, ensuring the maximum timestep $T_{\max}$ is strictly satisfied. This allows OTCache to faithfully "morph" high-fidelity structural priors into accelerated schedules for any target budget.

4 Experiments

4.1 Experimental setup

Baselines and Compared Methods. To rigorously assess the effectiveness and generalizability of OTCache, we conduct experiments on three state-of-the-art generative frameworks: FLUX.1 [dev] [20] for high-fidelity image synthesis, Qwen-Image [46] for text-to-image generation, and HunyuanVideo [19] for large-scale video generation. We benchmark OTCache against a comprehensive suite of representative caching paradigms. This includes threshold-based and structural redundancy methods, such as TaylorSeer [25], DBCache [42], DiCache [3], ToCa [53], and DuCa [54]. We further compare with trajectory-aware acceleration techniques, including TeaCache [23], as well as the latest Graph-based caching methods, LeMiCa [8] and MeanCache [9].

Metrics. To rigorously assess the trade-off between computational acceleration and generation performance, we adopt a multi-faceted evaluation suite spanning efficiency, perceptual quality, and structural fidelity. Efficiency is quantified by total FLOPs and inference latency. For text-to-image (T2I) generation, we follow the DrawBench [34] protocol and report ImageReward [47] and CLIP Score [12] to evaluate high-level perceptual quality and semantic text-image alignment, respectively. For text-to-video (T2V) tasks, we employ VBench [14] to capture human-centric preferences across multiple temporal and spatial dimensions. Furthermore, to quantify the potential fidelity degradation introduced by the caching mechanism, we treat the full-step (uncompressed) inference as the reference ground truth and compute LPIPS [48], SSIM [44], and PSNR. These reconstruction metrics serve to measure the degree of perceptual similarity, structural consistency, and pixel-level precision maintained by OTCache relative to the original diffusion trajectory.

Implementation Details. All experiments are implemented in PyTorch and executed on NVIDIA H100 GPUs. To ensure a rigorous and fair evaluation, we adopt the sampling protocol established by prior works [9, 23], selecting 50 representative prompts (10 per attribute) from the T2V-CompBench [41] dataset. For all model architectures, we employ FlashAttention [7] as the default attention backend to optimize memory bandwidth and computational throughput. In Stage 1 of OTCache, the reference budget is set to $B_{\text{ref}} = 20$. In Stage 2, the anchor search is conducted using Optuna with a budget of 200 trials to identify the optimal boundary caching sequence for 8-step inference. Notably, since TaylorSeer encounters out-of-memory (OOM) issues under HunyuanVideo, we uniformly adopt the cpu-offload setting to ensure fair comparison.

4.2 Text-to-Image Generation

As demonstrated in Table 1 and Table 2, **OTCache** consistently establishes a new state-of-the-art Pareto frontier on both FLUX.1 [dev] and Qwen-Image

models. By introducing a caching mechanism from an Optimal Transport (OT) perspective, our method achieves a superior trade-off between acceleration ratio and generation fidelity compared to existing baselines.

Performance on FLUX.1 [dev]. On the FLUX.1 [dev] model (1024×1024), OTCache ($\mathcal{B} = 15$) achieves a 3.04× speedup, surpassing MeanCache's 2.91×. Simultaneously, it further enhances reconstruction accuracy, reducing LPIPS from 0.142 to 0.126 and boosting PSNR from 24.83 to 26.03. Notably, in high-acceleration regimes ($> 3.6\times$), where baseline methods such as TeaCache experience catastrophic quality collapse, OTCache demonstrates remarkable robustness. At a significant 4.50× acceleration, it maintains an ImageReward of 0.996 and an LPIPS of 0.254, outperforming MeanCache in both efficiency (4.50× vs. 4.12×) and perceptual quality.

Table 1: Quantitative comparison of acceleration methods on **FLUX.1 [dev]** (1024 × 1024). Best results are highlighted in **bold**.

Method	Acceleration		Visual Quality				
	Latency(s) ↓	Speed ↑	Img Rew. ↑	CLIP ↑	LPIPS ↓	SSIM ↑	PSNR ↑
Original: 50 steps	11.57	1.00×	1.033	31.229	–	–	–
60% steps	7.01	1.65×	0.984	31.242	0.217	0.808	20.26
30% steps	3.60	3.21×	0.880	30.832	0.399	0.682	15.80
TeaCache ($l = 0.25$)	4.62	2.50×	0.960	31.145	0.338	0.721	17.29
DiCache ($\delta = 0.8$)	4.32	2.68×	0.675	30.814	0.416	0.717	21.27
TaylorSeer ($\mathcal{N} = 6, O = 2$)	4.24	2.74×	0.971	**31.310**	0.415	0.663	16.28
TaylorSeer ($\mathcal{N} = 6, O = 1$)	4.06	2.85×	0.961	31.191	0.419	0.660	15.83
LeMiCa ($\mathcal{B} = 15$)	4.13	2.80×	0.991	31.125	0.153	0.858	24.45
MeanCache ($\mathcal{B} = 15$)	3.98	2.91×	1.010	31.244	0.142	0.870	24.83
OTCache ($\mathcal{B} = 15$)	**3.80**	**3.04×**	**1.011**	31.167	**0.126**	**0.881**	**26.03**
TeaCache ($l = 1.5$)	3.16	3.66×	0.717	30.696	0.504	0.624	15.01
LeMiCa ($\mathcal{B} = 10$)	3.21	3.60×	0.981	**31.355**	0.312	0.740	19.03
MeanCache ($\mathcal{B} = 10$)	2.81	4.12×	0.993	31.323	0.272	0.761	19.43
OTCache ($\mathcal{B} = 10$)	**2.57**	**4.50×**	**0.996**	31.269	**0.254**	**0.780**	**20.22**

As shown in Fig. 3, on FLUX.1 [dev], when the acceleration exceeds 3.6×, baseline methods exhibit more frequent content inconsistencies and object distortions (e.g., malformed legs of the red goat, unnatural scissors, and redundant piano keys). In contrast, OTCache achieves better content consistency and higher visual quality at an even larger speedup (4.50×), substantially outperforming graph-based caching baselines.

Performance on Qwen-Image. For high-resolution generation on Qwen-Image (1664×928, 16:9 aspect ratio), the efficiency gains of OTCache are even more pronounced. At $\mathcal{B} = 15$, it reaches a 3.20× speedup with a near-lossless LPIPS of 0.069, which is significantly superior to MeanCache's 0.075 at 2.85×. Even under an extreme 4.70× acceleration ($\mathcal{B} = 10$), OTCache preserves high structural fidelity with an SSIM of 0.864 and a PSNR of 21.48, whereas other

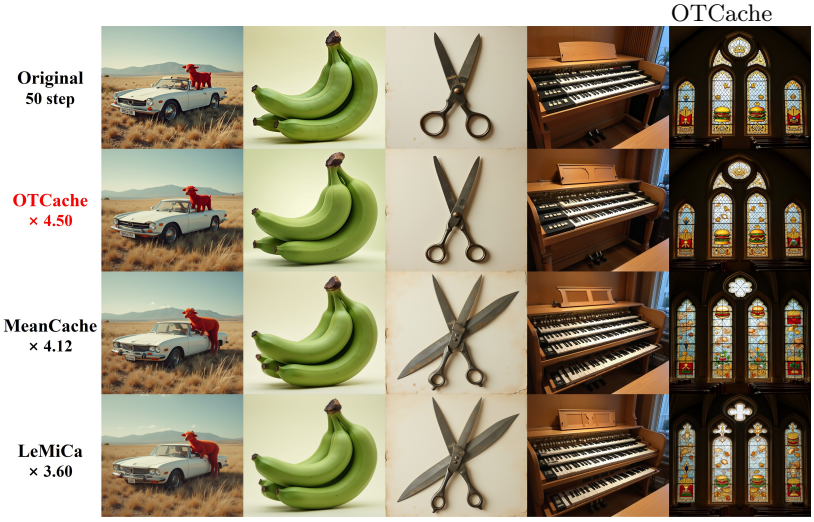

Fig. 3: Comparison of different methods at high acceleration ratios on **FLUX.1 [dev]** (1024×1024).

Table 2: Quantitative comparison of acceleration methods on **Qwen-Image** (1664 × 928). Best results are highlighted in **bold**.

Method	Acceleration		Visual Quality				
	Latency(s) ↓	Speed ↑	Img Rew. ↑	CLIP ↑	LPIPS ↓	SSIM ↑	PSNR ↑
Original: 50 steps	32.68	1.00×	1.180	33.626	–	–	–
30% steps	9.86	3.31×	1.128	33.026	0.363	0.727	15.83
TeaCache ($l = 0.6$)	18.52	1.76×	1.087	32.598	0.416	0.698	14.90
DBCache ($r = 0.6$)	11.92	2.74×	1.016	33.435	0.298	0.825	22.22
LeMiCa ($\mathcal{B} = 15$)	13.26	2.46×	1.120	33.590	0.122	0.924	26.89
MeanCache ($\mathcal{B} = 15$)	11.45	2.85×	1.159	33.636	0.075	0.938	27.66
OTCache ($\mathcal{B} = 15$)	**10.21**	**3.20×**	**1.164**	**33.652**	**0.069**	**0.943**	**28.12**
LeMiCa ($\mathcal{B} = 13$)	11.54	2.83×	1.096	33.667	0.177	0.884	24.06
MeanCache ($\mathcal{B} = 13$)	9.09	3.60×	1.147	**33.799**	0.113	0.907	24.80
OTCache ($\mathcal{B} = 13$)	**8.87**	**3.68×**	**1.150**	33.614	**0.087**	**0.930**	**26.79**
LeMiCa ($\mathcal{B} = 10$)	10.78	3.03×	1.111	**33.739**	0.253	0.816	19.09
MeanCache ($\mathcal{B} = 10$)	7.16	4.56×	1.142	33.621	0.236	0.815	18.98
OTCache ($\mathcal{B} = 10$)	**6.95**	**4.70×**	**1.147**	33.584	**0.171**	**0.864**	**21.48**

caching-based baselines either provide lower speedups or incur significantly higher reconstruction errors.

Similar to the observations on FLUX.1 [dev], the qualitative comparisons in Fig. 4 further highlight the strong generalization capability of OTCache for text-to-image models. In the 16:9 generation setting commonly used for poster-style compositions, OTCache maintains significantly better content consistency before and after acceleration. In contrast, graph-based caching baselines such as MeanCache and LeMiCa exhibit noticeable content shifts, including changes in the relative positions of objects (e.g., the cat and the octopus) and altered orientations of subjects (e.g., the giraffe). These qualitative findings are consistent with the analysis in Sec. 3.2 and the quantitative results reported in Table 2.

Fig. 4: Comparison of different methods at high acceleration ratios on **Qwen-Image** (1664×928).

4.3 Text-to-Video Generation.

As shown in Table 3, OTCache establishes a new state-of-the-art for efficient text-to-video generation on HunyuanVideo. By modeling cache scheduling as a budget-conditioned path evolution via Optimal Transport (OT), OTCache achieves a 3.21× speedup, matching TeaCache and surpassing MeanCache's 3.05×, while improving reconstruction fidelity (LPIPS 0.176→0.162).

OTCache also remains robust under extreme acceleration. At 3.66×, it achieves a VBench score of 80.37% with an LPIPS of 0.252, consistently outperforming MeanCache in both efficiency and visual quality, while prior methods such as DiCache and TeaCache exhibit noticeable temporal artifacts. As illustrated in Fig. 5, we further present qualitative comparisons on representative key frames. OTCache consistently produces more stable structures and clearer visual details, whereas competing methods show more distortions and temporal inconsistencies. Fig. 6 further provides a multi-dimensional comparison across VBench metrics.

Table 3: Quantitative comparison in text-to-video generation on **HunyuanVideo**. Best results are in **bold**.

Method	Acceleration		Visual Quality			
	Latency(s) ↓	Speed ↑	VBench ↑	LPIPS ↓	SSIM ↑	PSNR ↑
Original: 50 steps	105.92	1.00×	80.39%	–	–	–
30% steps	39.53	2.68×	79.84%	0.381	0.659	17.335
ToCa ($\mathcal{N}=5$)	36.17	2.93×	79.51%	0.454	0.590	15.765
Duca ($\mathcal{N}=5$)	34.32	3.09×	79.54%	0.454	0.595	15.807
DiCache ($\delta=0.8$)	33.76	3.11×	74.09%	0.382	0.701	22.053
TaylorSeer ($\mathcal{N}=5, O=1$)	34.95	3.03×	79.95%	0.428	0.603	16.026
Teacache ($l=0.33$)	34.06	3.11×	80.02%	0.363	0.651	17.957
MeanCache ($\mathcal{B}=12$)	33.05	3.21×	80.01%	0.176	0.809	24.002
OTCache ($\mathcal{B}=12$)	**33.01**	**3.21×**	**80.20%**	**0.162**	**0.815**	**24.356**
DiCache ($\delta=3.0$)	31.81	3.33×	70.86%	0.583	0.490	19.098
Teacache ($l=0.39$)	31.86	3.32×	79.75%	0.396	0.631	17.382
TaylorSeer ($\mathcal{N}=7, O=1$)	31.50	3.36×	79.76%	0.480	0.595	15.444
MeanCache ($\mathcal{B}=10$)	29.48	3.59×	80.08%	0.269	0.732	20.464
OTCache ($\mathcal{B}=10$)	**28.96**	**3.66×**	**80.37%**	**0.252**	**0.746**	**21.150**

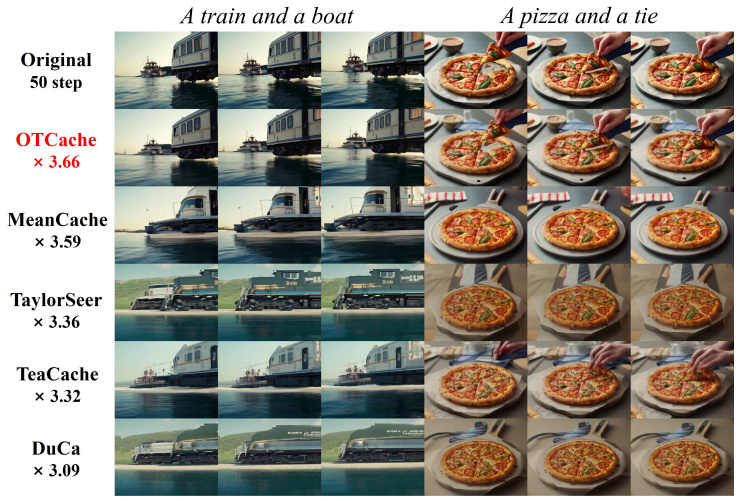

Fig. 5: Comparison of different methods at high acceleration ratios on **Hunyuan-Video**.

4.4 Ablation Study

Content Consistency Following MeanCache, we conduct a detailed evaluation of OTCache on rare-word generation, where semantic ambiguity and low-frequency usage pose significant challenges to text-to-image models. As shown in Figure 7, MeanCache outperforms TeaCache in maintaining content consistency under moderate acceleration. However, as the acceleration ratio increases, MeanCache gradually exhibits content drift: an extra round table appears at 2.43×, chairs are missing at 3.00×, and the bedside lamp disappears at 4.12×, indicating degradation of fine-grained semantic fidelity. In contrast, OTCache preserves content more faithfully across all acceleration levels. Notably, even under a more extreme 4.50× speedup (orange box), OTCache still retains critical elements such as the bedside lamp and its illumination, demonstrating superior robustness in maintaining rare-word semantics and structural details under aggressive acceleration.

Effect of ρ The parameter ρ modulates the NFE density, where higher values prioritize the early-stage reverse ODE to stabilize high-variance sampling. We evaluate $\rho \in [1.0, 1.45]$ at a fixed budget $\mathcal{B} = 15$ (Table 5). Compared to uniform sampling ($\rho = 1.0$), non-linear allocation significantly enhances reconstruction fidelity. Specifically, $\rho = 1.3$ achieves the optimal balance, yielding the highest PSNR (26.034) and lowest LPIPS (0.126). While $\rho = 1.0$ marginally leads in Image Reward, its inferior PSNR and LPIPS indicate structural instability. Conversely, $\rho = 1.45$ causes performance degradation, suggesting that excessive early-stage bias compromises late-stage refinement. Thus, we fix $\rho = 1.3$ for all subsequent experiments.

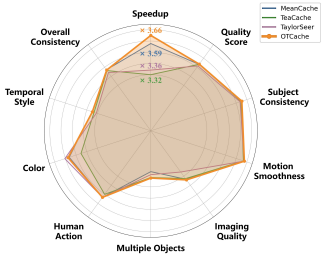

Fig. 6: VBench metrics and acceleration ratio of proposed OTCache and other methods.

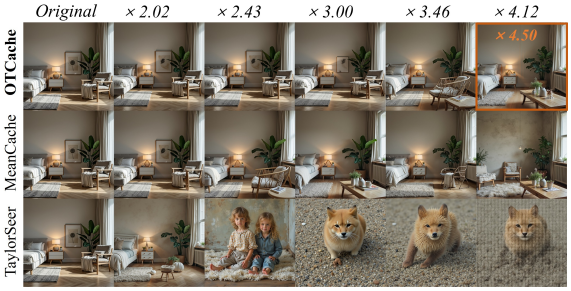

Fig. 7: Content consistency under a rare-word prompt ("Matutinal") across varying acceleration ratios

Table 4: Search accuracy on LPIPS ↓.

Rank	MC	Search	Gain (%)
Top-1	0.338	**0.261**	**25.04%**
Top-2	0.338	0.263	24.48%
Top-3	0.338	0.263	24.26%
Top-4	0.338	0.264	24.02%
Top-5	0.338	0.265	23.67%

Table 5: Impact of parameter ρ on quality metrics.

ρ	Img Rwd. ↑	PSNR ↑	SSIM ↑	LPIPS ↓
1.00	**1.022**	24.304	0.864	0.139
1.15	1.000	24.847	0.871	0.134
1.30	1.011	**26.034**	**0.880**	**0.126**
1.45	1.001	25.798	0.877	0.131

Effectiveness and Efficiency of Anchor Search We analyze the effectiveness and efficiency of the anchor search under $B_{\text{anc}} = 8$ across 100 prompt-seed pairs. Table 4 shows that the searched anchors consistently outperform the MeanCache (MC) baseline. The Top-1 schedule reduces LPIPS by **25.04%** on average, while even the Top-5 candidates maintain more than **23%** improvement over MC, indicating that directly optimizing the end-to-end perceptual objective effectively recovers the fidelity loss introduced by surrogate-based graph methods. Meanwhile, Fig. 8 demonstrates that the search remains highly efficient. The median number of trials required to identify the Top-1 optimum is around 50, and the vast majority of cases converge within 200 trials. These results show that the anchor search in Stage 2 of OTCache is both efficient and effective, providing a reliable boundary condition for schedule prediction with minimal computational overhead.

5 Conclusion

This paper introduces **OTCache**, a training-free framework for accelerating diffusion sampling through caching schedule prediction. Unlike existing graph-based caching methods that rely on additive shortest-path objectives, OTCache models the evolution of caching schedules across inference budgets as a smooth trajectory in policy space inspired by Optimal Transport. The framework consists of three stages: a high-fidelity reference schedule under a conservative budget, an anchor schedule obtained via lightweight end-to-end search in the ultra-low NFE regime, and a quantile-interpolation strategy that predicts schedules

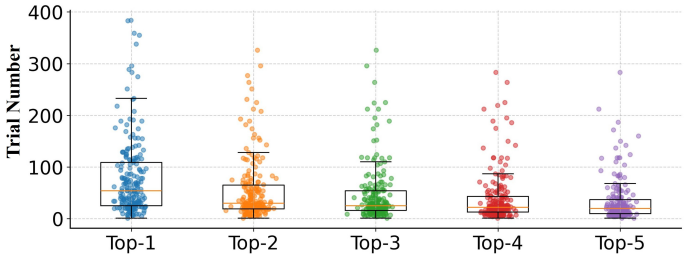

Fig. 8: Search efficiency. Number of trials required to identify Top-K schedules across 50 prompt-seed pairs at budget $B = 8$. The median convergence for the Top-1 optimum is around 50 trials, with most near-optimal schedules discovered within 100 trials.

for arbitrary budgets through continuous warping representations. More broadly, OTCache provides a new perspective on diffusion acceleration through geometry-aware schedule modeling.

Acknowledgements

This work was supported by the National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund Project under Grant U24B20181.

References

1. Aggarwal, A., Shrivastava, A., Gwilliam, M.: Evolutionary caching to accelerate your off-the-shelf diffusion model. arXiv preprint arXiv:2506.15682 (2025)
2. Albergo, M.S., Vanden-Eijnden, E.: Building normalizing flows with stochastic interpolants. arXiv preprint arXiv:2209.15571 (2022)
3. Bu, J., Ling, P., Zhou, Y., Wang, Y., Zang, Y., Wu, T., Lin, D., Wang, J.: Dicache: Let diffusion model determine its own cache. arXiv preprint arXiv:2508.17356 (2025)
4. Chen, P., Shen, M., Ye, P., Cao, J., Tu, C., Bouganis, C.S., Zhao, Y., Chen, T.: Delta dit: A training-free acceleration method tailored for diffusion transformers. arXiv preprint arXiv:2406.01125 (2024)
5. Chen, P., Zhang, X., Liu, Z., Hu, H., Liu, X., Wang, K., Wang, M., Qian, Y., Lian, S.: Optimizing for the shortest path in denoising diffusion model. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
6. Chen, Z., Ma, X., Fang, G., Tan, Z., Wang, X.: Asyncdiff: Parallelizing diffusion models by asynchronous denoising. arXiv preprint arXiv:2406.06911 (2024)
7. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems **35**, 16344–16359 (2022)
8. Gao, H., Chen, P., Shi, F., Tan, C., Liu, Z., Zhao, F., Wang, K., Lian, S.: Lemica: Lexicographic minimax path caching for efficient diffusion-based video generation. arXiv preprint arXiv:2511.00090 (2025)
9. Gao, H., Chen, P., Shi, F., Wu, R., YanTao, L., Hui, Q., You, Y., Lu, T., Tan, C., Zhao, S., et al.: Meancache: From instantaneous to average velocity for accelerating flow matching inference. arXiv preprint arXiv:2601.19961 (2026)
10. Geng, Z., Deng, M., Bai, X., Kolter, J.Z., He, K.: Mean flows for one-step generative modeling. arXiv preprint arXiv:2505.13447 (2025)
11. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
12. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718 (2021)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21807–21818 (2024)
15. Hung, C.Y., Majumder, N., Kong, Z., Mehrish, A., Zadeh, A., Li, C., Valle, R., Catanzaro, B., Poria, S.: Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization (2024), `https://arxiv.org/abs/2412.21037`
16. Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., Mitliagkas, I.: Gotta go fast when generating data with score-based models. arXiv preprint arXiv:2105.14080 (2021)
17. Kahatapitiya, K., Liu, H., He, S., Liu, D., Jia, M., Zhang, C., Ryoo, M.S., Xie, T.: Adaptive caching for faster video generation with diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15240–15252 (2025)

18. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems **35**, 26565–26577 (2022)
19. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
20. Labs, B.F.: Flux. `https://github.com/black-forest-labs/flux` (2024)
21. Li, Y., Xu, S., Cao, X., Sun, X., Zhang, B.: Q-dm: An efficient low-bit quantized diffusion model. Advances in neural information processing systems **36**, 76680–76691 (2023)
22. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
23. Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., Wan, F.: Timestep embedding tells: It's time to cache for video diffusion model. CoRR **abs/2411.19108** (2024). `https://doi.org/10.48550/ARXIV.2411.19108`, `https://doi.org/10.48550/arXiv.2411.19108`
24. Liu, J., Cai, P., Zhou, Q., Lin, Y., Kong, D., Huang, B., Pan, Y., Xu, H., Zou, C., Tang, J., et al.: Freqca: Accelerating diffusion models via frequency-aware caching. arXiv preprint arXiv:2510.08669 (2025)
25. Liu, J., Zou, C., Lyu, Y., Chen, J., Zhang, L.: From reusing to forecasting: Accelerating diffusion models with taylorseers. arXiv preprint arXiv:2503.06923 (2025)
26. Liu, J., Zou, C., Lyu, Y., Ren, F., Wang, S., Li, K., Zhang, L.: Speca: Accelerating diffusion transformers with speculative feature caching. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 10024–10033 (2025)
27. Liu, X., Gong, C., qiang liu: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
28. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. arXiv preprint arXiv:2211.01095 (2022)
29. Ma, X., Fang, G., Bi Mi, M., Wang, X.: Learning-to-cache: Accelerating diffusion transformer via layer caching. Advances in Neural Information Processing Systems **37**, 133282–133304 (2024)
30. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free. arXiv preprint arXiv:2312.00858 (2023)
31. Ma, X., Fang, G., Wang, X.: Llm-pruner: On the structural pruning of large language models. Advances in neural information processing systems **36**, 21702–21720 (2023)
32. Ma, X., Liu, Y., Liu, Y., Wu, X., Zheng, M., Wang, Z., Lim, S.N., Yang, H.: Model reveals what to cache: Profiling-based feature reuse for video diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17150–17159 (2025)
33. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)
35. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)

36. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. In: European Conference on Computer Vision. pp. 87–103. Springer (2024)
37. Shang, Y., Yuan, Z., Xie, B., Wu, B., Yan, Y.: Post-training quantization on diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1972–1981 (2023)
38. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
39. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models (2023)
40. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
41. Sun, K., Huang, K., Liu, X., Wu, Y., Xu, Z., Li, Z., Liu, X.: T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. arXiv preprint arXiv:2407.14505 (2024)
42. vipshop.com: cache-dit: A unified and training-free cache acceleration toolbox for diffusion transformers (2025), `https://github.com/vipshop/cache-dit.git`, open-source software available at https://github.com/vipshop/cache-dit.git
43. Wang, C., Guo, Z., Duan, Y., Li, H., Chen, N., Tang, X., Hu, Y.: Target-driven distillation: Consistency distillation with target timestep selection and decoupled guidance. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 7619–7627 (2025)
44. Wang, Z., Bovik, A.C.: A universal image quality index. IEEE signal processing letters **9**(3), 81–84 (2002)
45. Wimbauer, F., Wu, B., Schoenfeld, E., Dai, X., Hou, J., He, Z., Sanakoyeu, A., Zhang, P., Tsai, S., Kohler, J., et al.: Cache me if you can: Accelerating diffusion models through block caching. arXiv preprint arXiv:2312.03209 (2023)
46. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), `https://arxiv.org/abs/2508.02324`
47. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems **36**, 15903–15935 (2023)
48. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
49. Zhang, W., Liu, H., Xie, J., Faccio, F., Shou, M.Z., Schmidhuber, J.: Cross-attention makes inference cumbersome in text-to-image diffusion models. arXiv preprint arXiv:2404.02747 (2024)
50. Zhao, X., Cheng, S., Chen, C., Zheng, Z., Liu, Z., Yang, Z., You, Y.: Dsp: Dynamic sequence parallelism for multi-dimensional transformers. arXiv preprint arXiv:2403.10266 (2024)
51. Zhao, X., Jin, X., Wang, K., You, Y.: Real-time video generation with pyramid attention broadcast. arXiv preprint arXiv:2408.12588 (2024)
52. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all (2024), https://github.com/hpcaitech/Open-Sora
53. Zou, C., Liu, X., Liu, T., Huang, S., Zhang, L.: Accelerating diffusion transformers with token-wise feature caching. arXiv preprint arXiv:2410.05317 (2024)

54. Zou, C., Zhang, E., Guo, R., Xu, H., He, C., Hu, X., Zhang, L.: Accelerating diffusion transformers with dual feature caching. arXiv preprint arXiv:2412.18911 (2024)