

TS-DP: Reinforcement Speculative Decoding For Temporal Adaptive Diffusion Policy Acceleration

Ye Li, Jiahe Feng, Yuan Meng[†], *Member, IEEE*, Kangye Ji, Chen Tang, Jiajun Fan, Xinwan Wen, Jiawei Li, Wenwu Zhu, *Fellow, IEEE*, Shu-Tao Xia[†], *Senior Member, IEEE*, Xin Wang[†], *Member, IEEE*, Zhi Wang[†], *Senior Member, IEEE*

Abstract—Diffusion Policy (DP) excels in embodied control but suffers from high inference latency and computational cost due to multiple iterative denoising steps. Existing diffusion model (DMs) acceleration methods improve inference speed by reducing computational overhead, as DMs for image generation are computation-bound. In this work, we reveal that DP are *I/O bound*, and thus propose using speculative decoding, a parallel computing framework, to accelerate DP for the first time. However, it is non-trivial to address the following challenges: (1) How to preserve the denoising quality of the base model at reduced computational cost under time-varying task difficulty in embodied settings; and (2) How to dynamically allocate computation according to task difficulty in such environments. Therefore, we propose **Temporal-aware Reinforcement-based Speculative Diffusion Policy (TS-DP)**, the first framework that enables speculative decoding for DP with temporal adaptivity. First, to handle dynamic environments where task difficulty varies over time, we distill a Transformer-based drafter to imitate the base model and replace its costly denoising calls. Second, an RL-based scheduler further adapts to the time-varying task difficulty by adjusting speculative parameters to maintain accuracy while improving efficiency. Extensive experiments show that TS-DP achieves up to $4.17\times$ faster inference with a draft acceptance rate of over 85% in simulation. Real-world experiments further demonstrate a $3.8\times$ speedup while improving success rates on real-time tasks (e.g., a 23% gain on the conveyor-belt task), reaching an inference frequency of 26 Hz and enabling real-time control while maintaining comparable task performance. The code is available at <https://github.com/ChildTang/TS-DP>.

Index Terms—Diffusion Policy, Speculative Decoding, Acceleration, Embodied Intelligence, Multimodal Model

I. INTRODUCTION

DIFFUSION Policy (DP) [1] has been widely adopted in Vision-Language-Action (VLA) models [2]–[8] thanks to its strong ability to model multi-modal action distributions

Ye Li, Yuan Meng, Kangye Ji, Xinwan Wen, Jiawei Li, Wenwu Zhu, Shu-Tao Xia, Xin Wang, and Zhi Wang are with Tsinghua University, Beijing, China. Ye Li, Kangye Ji, Shu-Tao Xia, and Zhi Wang are also with Tsinghua Shenzhen International Graduate School (SIGS), Tsinghua University. Yuan Meng, Wenwu Zhu, and Xin Wang are also with the Key Laboratory of Pervasive Computing (Ministry of Education) and the Beijing National Research Center for Information Science and Technology (BNRIST), Department of Computer Science and Technology, Tsinghua University. Jiahe Feng is with the College of Artificial Intelligence, Tianjin University, Tianjin, China. Chen Tang is with MMLab, The Chinese University of Hong Kong, Hong Kong SAR, China. Jiawei Li is also with Joy Future Academy, JD Group, Beijing, China. Jiajun Fan is with the Siebel School of Computing and Data Science, The Grainger College of Engineering, University of Illinois Urbana-Champaign, Urbana, IL, USA.

Email: {wangzhi, xiast}@sz.tsinghua.edu.cn, {yuanmeng, xin_wang}@tsinghua.edu.cn.

[†]Corresponding author.

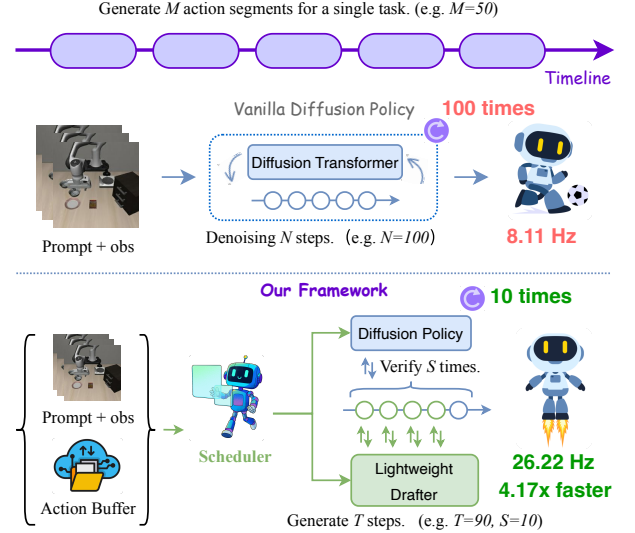


Fig. 1. **Vanilla Diffusion Policy (DP) vs. Our TS-DP.** For embodied tasks, DP completes a task through multiple interactions with the environment, generating several segments of actions. Each segment typically requires hundreds of model calls for denoising, making the process highly time-consuming. TS-DP introduces a customized speculative decoding framework for temporal-complexity-aware accelerating DP. A lightweight drafter generates multiple denoising results, which are verified in parallel by the DP for near-lossless acceleration. A scheduler further adjusts key speculative decoding parameters—such as the number of drafts and the acceptance rate—based on the complexity of different task phases, achieving adaptive and efficient acceleration.

and generate coherent action trajectories. However, its large backbone and iterative denoising process impose substantial computational overhead, resulting in low action-generation frequency and limiting its applicability in latency-sensitive real-world settings. Consequently, improving the inference efficiency of DP has therefore emerged as a critical challenge. Recent research has increasingly focused on accelerating diffusion models (DMs) to enable faster inference. General model compression techniques, including pruning and quantization [9]–[12], reduce computational cost by simplifying the diffusion backbone and have shown effectiveness in static generation tasks. Building on these advances, a growing body of work explores specific acceleration techniques for DP, where cache-based methods [13], [14] reuse intermediate features from previous denoising steps to eliminate redundant computation during action generation. While effective to some extent, these approaches inevitably alter internal feature representations, resulting in *lossy acceleration* and constrained

overall speedup. Furthermore, they remain largely static and fail to account for the temporal correlations and evolving task complexity inherent in embodied control.

Meanwhile, speculative decoding has emerged as a loss-less and complexity-adaptive acceleration paradigm, where a lightweight drafter proposes multiple candidates that are verified in parallel by a stronger model. This strategy has achieved notable success in LLMs [15]–[18], delivering substantial speedups without compromising fidelity, and has also been adapted to DMs by accelerating the denoising process [19], [20]. However, these approaches primarily target compute-bound DMs for image and video generation, where GPU computation dominates and the verification stage offsets most potential savings, resulting in limited acceleration. In contrast, **DP is inherently I/O-bound** as it generates low-dimensional action vectors instead of high-resolution visual data. This property makes speculative decoding particularly well-suited for DP inference. Yet DP differs fundamentally from conventional DMs: it must produce **sequential** action segments while continuously interacting with an evolving environment, where task difficulty and dynamics change over time, making temporal adaptivity essential for stable and efficient acceleration. As a result, temporal adaptivity becomes crucial for maintaining both stability and efficiency. This motivates a temporal-complexity-aware speculative decoding framework that preserving behavioral consistency and robust control performance.

Therefore, we propose TS-DP, the first temporal-complexity-aware speculative decoding framework designed to accelerate DP, as illustrated in Fig. 1. This presents two key challenges: (1) how to match the DP’s denoising quality at lower cost under time-varying task difficulty, and (2) how to dynamically adjust computation based on task difficulty to ensure stable and efficient control.

For the first challenge, prior studies leverage either the differences between consecutive time steps as drafts or the exchangeability of trajectories to enable self-speculation and parallel verification. However, embodied tasks are highly diverse, and their multi-segment action sequences often exhibit significant distributional differences, making these approaches difficult to generalize and largely limited to image generation. To address this, we adopt a single Transformer block as the drafter and train it via knowledge distillation, where DP serves as the teacher and the drafter is optimized to minimize both its deviation from the ground truth and its discrepancy from the teacher’s outputs. The drafter shares the same input–output interface as DP, performs multiple denoising steps in its place, and has its results verified in parallel by DP. During each speculative decoding round, accepted denoising steps are retained, while the first rejected one is corrected via Reflection-Maximal Coupling to better align with DP’s denoising distribution. This process repeats iteratively until the denoising is complete.

For the second challenge, the temporal characteristics of embodied tasks cause task difficulty to vary dynamically over time. As shown in Fig. 7, empirical analysis reveals an inverse relationship between end-effector velocity and the number of accepted drafts—faster, coarse-grained motions lead to fewer accepted drafts, while slower, fine-grained actions yield more.

This indicates that using fixed parameters for speculative decoding is suboptimal. Existing approaches, such as SP-VLA [21], distinguish task difficulty by velocity, which limits their generalization across diverse behaviors. To address this, we introduce a Reinforcement Learning (RL)–based scheduler that dynamically adjusts speculative decoding parameters according to task phase, enabling adaptive and efficient acceleration. Extensive experiments demonstrate that TS-DP achieves a draft acceptance rate of **over 85%**, increasing the inference frequency from **8 Hz** to **26 Hz** while reducing the Number of Function Evaluations (NFE) by $4.17\times$, all while maintaining comparable task performance. Real-world results further demonstrate up to a **$3.8\times$** NFE reduction and improved success rates on real-time tasks (e.g., a **23%** increase on the conveyor-belt task), confirming its effectiveness for real-time control. The main contributions are as follows:

- (1) We propose TS-DP, a temporal-complexity-aware speculative decoding framework for DP, enabling frequency-adaptive and accuracy-preserving acceleration. To the best of our knowledge, this is the first work that brings speculative decoding to DP with temporal adaptivity.
- (2) We propose a speculative decoding framework tailored for DP. To match the base model’s denoising quality at lower cost under time-varying task difficulty, we distill a Transformer-based drafter to imitate the base model and replace its costly multi-step denoising calls, while DP verifies its outputs in parallel.
- (3) We propose a temporal-complexity-aware speculative decoding scheduler. To dynamically adjust computation based on task difficulty, we train a PPO-based scheduler that formulates DP execution as a Markov process and adaptively tunes speculative parameters over time.

II. RELATED WORK

A. Diffusion Policy Acceleration

Benefiting from the strong modeling capacity and widespread success of DMs [22]–[24], DP has emerged as a leading framework for visuomotor policy learning. It leverages the generative capability of DMs to produce continuous and multimodal robotic actions conditioned on visual observations and language instructions [1], [25]. Despite their strong performance, DP suffer from high inference latency caused by iterative denoising, which limits their applicability in real-time, high-frequency robotic control.

To tackle this challenge, extensive research has focused on accelerating diffusion-based generative models and their extensions to embodied policies [21], [26]–[31]. A variety of techniques are applied to accelerate diffusion models in the field of image and video generation, such as quantization [9], [12], [32]–[36], feature cache [37]–[42]. In contrast, EfficientVLA [43] and BAC [13] are specifically designed for DP acceleration, where the latter introduces block-wise adaptive caching to selectively refresh upstream features and mitigate error propagation. However, most existing methods modify intermediate representations during inference, resulting in lossy acceleration, and few explicitly account for the temporal redundancy inherent in DP.

B. Speculative Decoding

Speculative decoding has recently emerged as a principled paradigm for accelerating autoregressive generation. Originally developed for LLMs [44]–[48], it employs a lightweight *drafter* to propose multiple future tokens that are verified in parallel by a stronger *target* model within a single forward pass, achieving provable lossless acceleration. Representative approaches such as the Medusa and Eagle series [15]–[18] further enhance efficiency through multi-head drafting and hierarchical verification mechanisms.

Several studies have extended speculative decoding to continuous generative processes. [19] introduced the frozen target draft model, which accelerates diffusion denoising by using stepwise differences as drafts and applies reflection-maximal coupling to preserve the original sampling distribution. Building on this, [20] proposed autospeculative decoding, eliminating the need for a separate drafter by exploiting trajectory exchangeability for self-speculation and parallel verification. While these methods achieve lossless acceleration in DDPMs and LDMs, they are primarily designed for compute-bound diffusion models in image and video generation, and the verification overhead largely cancels out the benefit—resulting in the counterintuitive phenomenon that even using 8 GPUs for verification yields only about a $1.8\times$ speedup.

In contrast, DP is inherently I/O-bound and therefore better suited for speculative decoding. The drafter operates at a much lower computational cost than the base policy, allowing parallel verification to be substantially more efficient than in compute-bound DMs. However, unlike standard DMs, DP must generate sequential action segments through interactions with the environment, where early prediction errors can accumulate and degrade overall task performance. To fix them, we introduce **TS-DP**, a temporal-complexity-aware speculative decoding framework that enables adaptive, near-lossless acceleration for embodied tasks.

III. METHOD

A. Preliminaries

Speculative decoding. Speculative decoding leverages a lightweight draft model p to generate a short trajectory of length L in a single forward pass, while the large-scale target model q verifies the generated drafts in parallel. This approach reduces the number of target model invocations, thereby improving inference efficiency. The joint distribution of the drafted latents is formulated as:

$$\tilde{\mathbf{x}}_{t+1:t+L} \sim p(\tilde{\mathbf{x}}_{t+1:t+L} \mid \mathbf{x}_t) = \prod_{j=1}^L p(\tilde{\mathbf{x}}_{t+j} \mid \tilde{\mathbf{x}}_{t+1:t+j-1}, \mathbf{x}_t), \quad (1)$$

where $\mathbf{x}_t$ denotes the current diffusion latent at timestep t , and $\tilde{\mathbf{x}}_{t+j}$ denotes the j -th drafted latent. Each drafted latent is accepted according to the standard likelihood-ratio test:

$$\alpha_j(\tilde{\mathbf{x}}_{t+j}) = \min\left(1, \frac{q(\tilde{\mathbf{x}}_{t+j} \mid \tilde{\mathbf{x}}_{t+1:t+j-1}, \mathbf{x}_t)}{p(\tilde{\mathbf{x}}_{t+j} \mid \tilde{\mathbf{x}}_{t+1:t+j-1}, \mathbf{x}_t)}\right). \quad (2)$$

The first rejection index is defined as:

$$j^* = \min\{j \in \{1, \dots, L\} : U_j > \alpha_j(\tilde{\mathbf{x}}_{t+j})\}, \quad (3)$$

where $U_j \sim \text{Unif}(0, 1)$ are independent uniform random variables. All drafted latents after j^* are discarded, and a corrective coupling is applied at step $t + j^* - 1$ to obtain a latent consistent with the target distribution.

The window length L and the closeness of p to q jointly control the trade-off between parallelism and the probability of encountering a rejection within the window.

Reflection-maximal coupling. For continuous, approximately Gaussian one-step conditionals, we apply *reflection-maximal coupling* to correct the first rejected draft sample through a single deterministic mapping. Let

$$r(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{m}_r, \Sigma), \quad s(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{m}_s, \Sigma), \quad (4)$$

where $r(\mathbf{x})$ and $s(\mathbf{x})$ denote the draft and target Gaussian densities with means $\mathbf{m}_r$ and $\mathbf{m}_s$ and shared covariance Σ . Draw $\tilde{\mathbf{x}} \sim r$ and perform the acceptance test:

$$A = \mathbf{1}\left(U \leq \min\left(1, \frac{s(\tilde{\mathbf{x}})}{r(\tilde{\mathbf{x}})}\right)\right), \quad U \sim \text{Unif}(0, 1). \quad (5)$$

If $A = 1$, accept $\mathbf{x} = \tilde{\mathbf{x}}$. If $A = 0$, obtain a corrected sample by reflecting $\tilde{\mathbf{x}}$ across the hyperplane orthogonal to $\Delta := \mathbf{m}_r - \mathbf{m}_s$. In the isotropic case $\Sigma = \sigma^2 \mathbf{I}$, the reflection takes the form:

$$\mathbf{x} = \mathbf{m}_s + (\mathbf{I} - 2\mathbf{e}\mathbf{e}^\top)(\tilde{\mathbf{x}} - \mathbf{m}_r), \quad \mathbf{e} := \frac{\Delta}{\|\Delta\|_2}. \quad (6)$$

Equations 4–6 depend only on the means $\mathbf{m}_r$, $\mathbf{m}_s$, the shared covariance Σ , and a single draw $\tilde{\mathbf{x}} \sim r$. The mapping in 6 generates a sample whose marginal distribution follows s , while minimizing the discrepancy between the rejected draft and its corrected counterpart.

B. Speculative Drafting and Verification Framework

Draft Model Training. The draft model is a lightweight approximator of the target model, designed for fast speculative rollouts while maintaining behavioral alignment. Since a lightweight architecture is sufficient for approximate denoising, we employ a single-layer Transformer block as the draft model, which shares the same encoder and DDPM or DDIM scheduler with the target model M_ϕ .

Specifically, we train the draft model $\hat{M}_\theta$ via knowledge distillation with two objectives. (1) The first component is a prediction-level distillation loss that aligns the draft model output $\hat{\mathbf{m}}_t^\theta$ with the target model output $\mathbf{m}_t^\phi$:

$$\mathcal{L}_{\text{pred}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left\| \hat{\mathbf{m}}_t^\theta - \mathbf{m}_t^\phi \right\|_2^2, \quad (7)$$

where t is the diffusion timestep, $\mathbf{x}_0$ is the clean action, and ϵ is the Gaussian noise used to construct the noisy latent. (2) The second component is a scheduler-aware normalized loss that aligns the draft model with the data-consistent denoising behavior of the target model:

$$\mathcal{L}_{\text{norm}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left\| \frac{\hat{\boldsymbol{\mu}}_t^\theta - \boldsymbol{\mu}_t^\phi}{\sigma_t} \right\|_2^2, \quad (8)$$

where $\boldsymbol{\mu}_t^\phi$ and $\hat{\boldsymbol{\mu}}_t^\theta$ are the data-aligned DDPM means computed from the target and draft predictions, and σ_t is the DDPM standard deviation at timestep t .

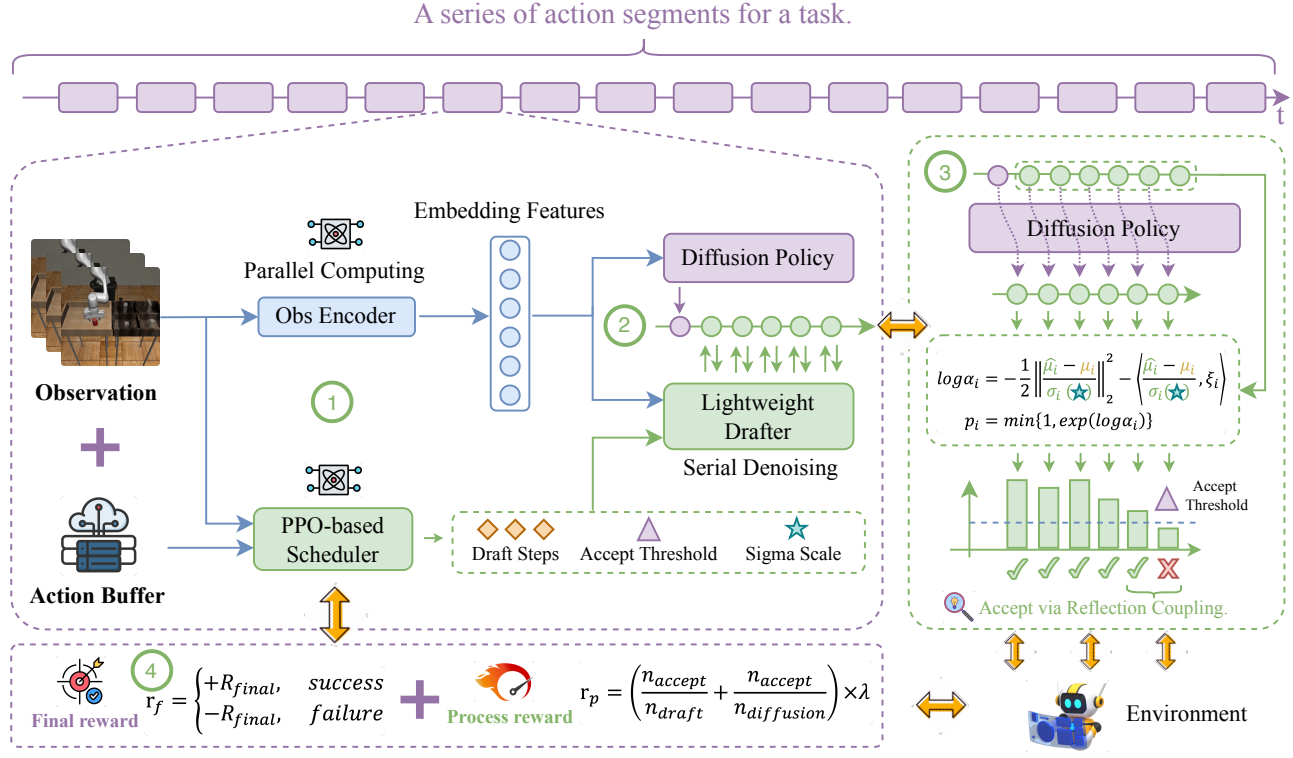


Fig. 2. **The framework of TS-DP.** TS-DP is the first temporal-aware speculative decoding framework designed to accelerate Diffusion Policy. ① **Decision stage:** A PPO-based scheduler evaluates time-varying task difficulty using past observations and actions, and produces adaptive speculative parameters. It operates in parallel with the observation encoder, adding no extra inference latency. ② **Denoising stage:** Guided by the scheduler, a lightweight drafter executes multiple denoising steps sequentially, effectively replacing expensive DP calls and reducing computational load. ③ **Verification stage:** DP verifies all drafted steps in parallel, accepting those that pass validation and correcting the first rejected draft via reflection-maximal coupling to better align it with the target distribution. ④ **Reward generation:** Based on task progress, the framework provides process rewards that encourage the scheduler to produce more valid drafts, and a final success-driven reward upon task completion, jointly guiding the scheduler toward efficient and successful task execution.

The full objective combines both terms:

$$\mathcal{L}(\theta) = \lambda_1 \mathcal{L}_{\text{pred}}(\theta) + \lambda_2 \mathcal{L}_{\text{norm}}(\theta), \quad (9)$$

with $\lambda_1, \lambda_2 > 0$. The target model parameters ϕ are kept frozen throughout training, while only the draft parameters θ are updated.

Draft Generation Procedure. During the inference phase, speculative drafts are generated through a short and efficient forward rollout using the lightweight draft model, conditioned on the current denoised estimate produced by the target model. Let M denote the target model and $\mathcal{S}$ the DDPM scheduler. At each time step t , the target model predicts the noise estimate $\mathbf{m}_t = M(\mathbf{x}_t; t)$, and the scheduler $\mathcal{S}$ produces the immediate one-step denoised candidate $\mathbf{x}_{t-1}^{(0)} = \mathcal{S}(\mathbf{m}_t, t, \mathbf{x}_t)$.

Starting from $\mathbf{x}_{t-1}^{(0)}$, a lightweight draft model $\hat{M}$ performs an autoregressive-style multi-step rollout for up to K steps. For each draft step $k = 1, \dots, K$, we compute the draft prediction at time $t - k$ and apply the same DDPM scheduler to obtain the next draft sample:

$$\begin{aligned} \hat{\mathbf{m}}_{t-k} &= \hat{M}(\mathbf{x}_{t-k+1}^{(\text{draft})}; t - k), \\ \mathbf{x}_{t-k}^{(\text{draft})} &= \mathcal{S}(\hat{\mathbf{m}}_{t-k}, t - k, \mathbf{x}_{t-k+1}^{(\text{draft})}). \end{aligned} \quad (3)$$

This rollout continues until either the maximum draft horizon K is reached or the diffusion process arrives at $t = 0$.

As shown in Fig. 6a, the acceptance probability displays clear phase-dependent trends: it is low during both the early high-noise and late low-noise stages, while remaining high and stable throughout the intermediate denoising phase. To balance acceleration and sample fidelity, TS-DP treats the draft horizon K as a timestep-dependent parameter, using smaller K in high- and low-noise stages and larger K in intermediate phases where predictions are more reliable. The scheduler dynamically adjusts K to maintain high draft acceptance and maximize speedup without degrading perceptual quality.

Verification of Speculative Drafts. Given the one-step candidate $\mathbf{x}_{t-1}^{(0)}$ from the target model and a sequence of K draft proposals $\{\mathbf{x}_{t-1}^{(1)}, \mathbf{x}_{t-2}^{(2)}, \dots, \mathbf{x}_{t-K}^{(K)}\}$ generated by the draft model, we form the candidate set $\mathcal{C}$. Verification is performed by evaluating the frozen target model M_ϕ on all elements of $\mathcal{C}$ in a single batched forward pass. For each candidate, we obtain the conditional mean μ_i and the effective standard deviation σ_i from the scheduler. As shown in Fig. 6b, the standard deviation is critical for draft acceptance: without adjustment, acceptance collapses in later denoising stages due to numerical overconfidence. We therefore treat σ_i as a tunable parameter and adapt it via the scheduler to maintain stable acceptance and consistent speculative acceleration across the trajectory. Assuming a Gaussian conditional distribution, the Metropolis-Hastings log acceptance ratio for the i -th draft proposal

TABLE I
BENCHMARK ON PROFICIENT HUMAN (PH) DEMONSTRATION DATA.

Method	Success Rate (% , $\uparrow$)						AVG (%, $\uparrow$)	NFE (%, $\downarrow$)	Speed $\times$
	Lift	Can	Square	Transport	Tool	Push-T			
Diffusion Policy [1]	100 / 100	95 / 97	82 / 88	78 / 81	43 / 53	59 / 64	76 / 80	100 / 100	–
Frozen Target Draft [19]	90 / 92	82 / 84	70 / 70	50 / 64	26 / 34	16 / 17	56 / 60	33 / 32	3.03 / 3.13
SpeCa [39]	100 / 100	92 / 96	88 / 88	68 / 72	38 / 40	67 / 69	76 / 78	34 / 35	2.94 / 2.86
BAC [13]	100 / 100	94 / 97	82 / 89	77 / 82	49 / 55	59 / 62	77 / 81	– / –	3.40 / 3.40
TS-DP	100 / 100	95 / 95	86 / 88	87 / 84	61 / 52	79 / 63	85 / 80	24 / 24	4.17 / 4.17

Accuracy, NFE, and acceleration performance are reported; TS-DP achieves a 4.17 $\times$ acceleration on such tasks while also improving accuracy by 9%. On certain tasks, such as Transport and Tool, TS-DP even surpasses the baseline. This improvement can be attributed to the implicit filtering effect of TS-DP, which enhances the smoothness and stability of the manipulator during task execution. Consistent with the original DP paper [1], each cell reports results obtained from different checkpoints.

TABLE II
BENCHMARK ON MIXED HUMAN (MH) DEMONSTRATION DATA.

Method	Success Rate (% , $\uparrow$)				AVG (%, $\uparrow$)	NFE (%, $\downarrow$)	Speed $\times$
	Lift	Can	Square	Transport			
Diffusion Policy [1]	99 / 100	92 / 97	76 / 79	35 / 46	76 / 81	100 / 100	–
Frozen Target Draft [19]	66 / 62	60 / 54	50 / 62	20 / 42	49 / 55	34 / 33	2.94 / 3.03
SpeCa [39]	96 / 100	86 / 90	76 / 68	34 / 40	73 / 75	35 / 37	2.86 / 2.70
BAC [13]	99 / 98	95 / 97	77 / 79	30 / 46	75 / 80	– / –	3.41 / 3.41
TS-DP	99 / 99	92 / 97	74 / 78	34 / 62	75 / 84	26 / 26	3.84 / 3.84

Accuracy, NFE, and acceleration performance are reported; TS-DP maintains comparable empirical accuracy on this task while delivering a 3.84 $\times$ acceleration. Due to the inherent complexity of this type of data, the training of drafter is affected to some extent; nevertheless, we still achieve near-lossless performance.

TABLE III
BENCHMARK ON MULTI-STAGE TASK (KITCHEN & BP).

Method	Success Rate (% , $\uparrow$)						AVG (%, $\uparrow$)	NFE (%, $\downarrow$)	Speed $\times$
	BP _{p1}	BP _{p2}	Kit _{p1}	Kit _{p2}	Kit _{p3}	Kit _{p4}			
Diffusion Policy [1]	98 / 98	98 / 96	100 / 100	100 / 100	100 / 100	100 / 100	99 / 99	100 / 100	–
Frozen Target Draft [19]	85 / 86	1 / 2	100 / 100	100 / 100	100 / 100	100 / 100	81 / 81	33 / 33	3.03 / 3.03
SpeCa [39]	97 / 97	85 / 85	100 / 100	100 / 100	100 / 100	100 / 100	97 / 97	37 / 37	2.70 / 2.70
BAC [13]	100 / 99	97 / 95	100 / 99	100 / 99	100 / 99	94 / 97	99 / 98	– / –	3.60 / 3.60
TS-DP	99 / 100	97 / 97	100 / 100	100 / 100	100 / 100	100 / 100	99 / 99	27 / 27	3.70 / 3.70

In the Kitchen task, p_x denotes the frequency of interacting with x or more objects (e.g., microwave, burners, oven switch, kettle); in the BP task, p_x represents the success rate under progressively harder control phases; TS-DP maintains comparable empirical accuracy on this task while delivering a 3.7 $\times$ acceleration.

$\mathbf{x} = \hat{\boldsymbol{\mu}}_i + \sigma_i \boldsymbol{\xi}_i$, where $\boldsymbol{\xi}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, is:

$$\log \alpha_i = -\frac{1}{2} \left\| \frac{\hat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i}{\sigma_i} \right\|_2^2 - \left\langle \frac{\hat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i}{\sigma_i}, \boldsymbol{\xi}_i \right\rangle. \quad (10)$$

The corresponding acceptance probability is computed as:

$$p_i = \min\{1, \exp(\log \alpha_i)\}. \quad (11)$$

We accept the draft if $p_i \geq \lambda$, where $\lambda \in (0, 1]$ is a tunable acceptance threshold; otherwise, it is rejected. As shown in Fig. 7, the number of accepted drafts is generally inversely related to the end-effector velocity—more drafts are accepted during slower motions, while fewer are accepted at higher speeds. This observation aligns with the typical behavior of embodied agents, which perform coarse, high-speed actions

such as locomotion and fine, low-speed actions such as grasping [21]. However, the non-strict correlation between velocity and acceptance rate indicates that task complexity also plays a critical role. Hence, using fixed speculative decoding parameters is suboptimal for embodied control. To address this, we treat λ as an adaptive hyperparameter tuned by the scheduler based on task complexity and velocity, enabling efficient and stable speculative decoding across task phases. Finally, we scan the candidates in chronological order, committing all consecutive drafts that are accepted until the first rejection occurs. For the first rejected draft, we apply the reflection coupling algorithm to construct a corrected sample that preserves the original stochasticity while better aligning with the target distribution. Unlike standard speculative decoding

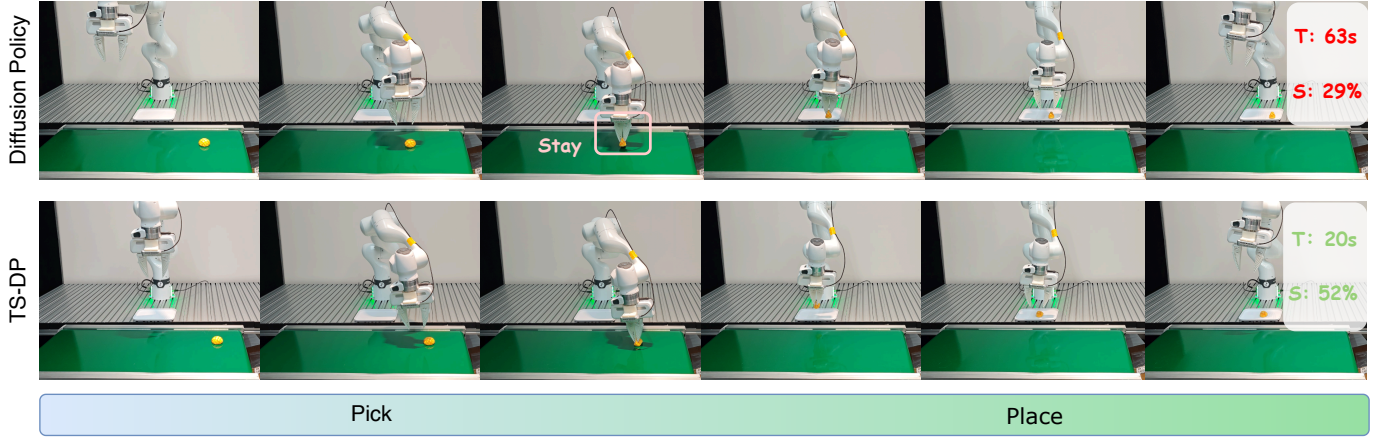


Fig. 3. **Conveyor-Belt Pick-and-Place Task.** This figure shows a Franka robotic arm picking pineapple buns from a moving conveyor belt and placing them onto a plate. TS-DP finishes in 20s, compared to 63s for DP, yielding a 3.2 $\times$ speedup. This improved real-time performance also increases the success rate from 29% to 52%. Here, T denotes time in seconds (s), and S denotes success rate in percentage (%).

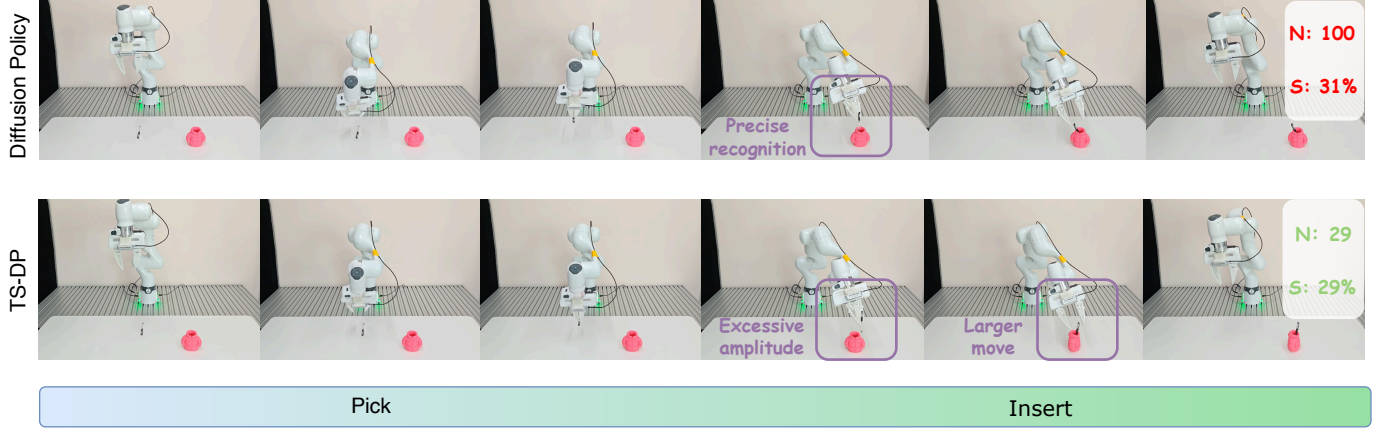


Fig. 4. **Pen-to-Holder Insertion Task.** This task is designed to examine the impact of TS-DP on high-difficulty, high-precision manipulation. As shown in the figure, DP exhibits greater precision at critical moments, such as more accurate alignment with the pen holder, resulting in smaller displacements. Although TS-DP shows minor inaccuracies in these cases, it is able to quickly correct them and successfully complete the task. It ultimately maintains a 3.4 $\times$ NFE speedup, with the success rate decreasing by only 2%. Here, N denotes NFE, and S denotes success rate(%).

under fixed proposal and target distributions, TS-DP adaptively adjusts the effective variance, acceptance threshold, and draft budget; therefore, it targets comparable task-level performance rather than exact distributional equivalence with the original DP.

C. Temporal-Complexity-Aware Speculative Decoding Scheduler

Markov Modeling for Speculative Decoding. To enable dynamic decision-making of speculative decoding parameters at each time step and train the RL model, we first formulate the speculative decoding process as a Markov chain with a fixed number of steps. Specifically, we integrate the DP, drafter, and embodied simulation environment into a unified interactive system. (1) **The observation space** comprises three components: the object states returned by the embodied environment (either images or state vectors), the embodied actions generated by DP at each time step, and the current task progress. (2) **The action space** includes the sigma scale,

acceptance threshold, and draft step configurations for the three distinct denoising stages.

We adopt a RL-based scheduler implemented with Proximal Policy Optimization (PPO) [49]–[51], as it provides stable policy updates and efficient adaptation to continuous task variations. In terms of network design, we separately encode the observation, action, and process streams to prevent interference caused by their differing dimensionalities. The extracted features are then aggregated using an MLP. When the observation is an image, a CNN is employed for feature extraction; otherwise, an MLP is used.

Reward function. The reward function aims to balance accuracy and efficiency, and is therefore divided into two components: the final reward and the intermediate reward. The final reward serves as the primary objective of the scheduler, aiming to ensure task accuracy. It is categorized into two types: discrete outcomes and continuous outcomes:

$$r_{\text{final_dis}} = \begin{cases} +R_{\text{final}}, & \text{success,} \\ -R_{\text{final}}, & \text{failure.} \end{cases} \quad (12)$$

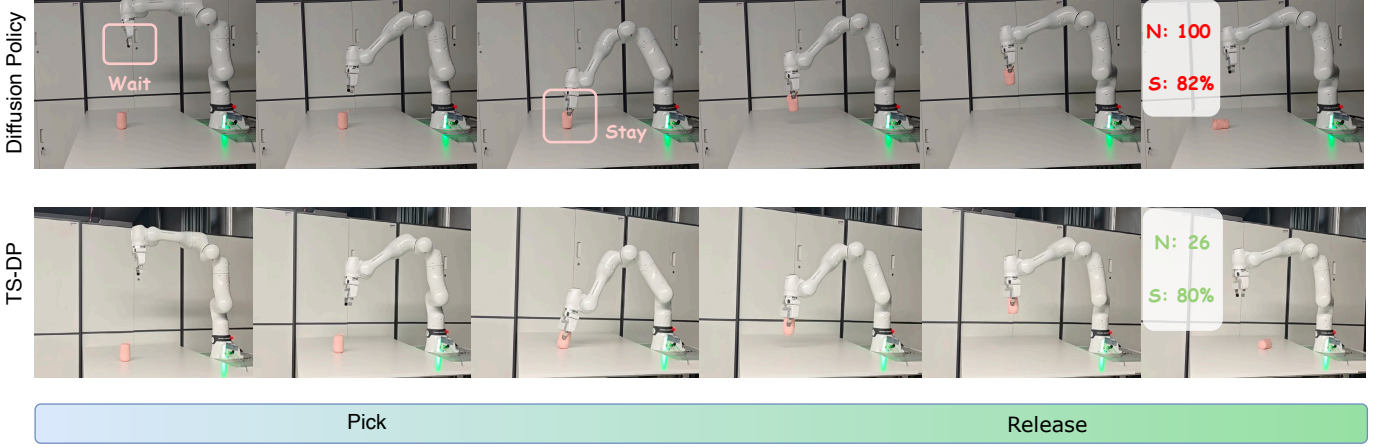


Fig. 5. **Pick-and-Release Task.** As shown in the figure, DP exhibits prolonged pauses both at the start and after grasping, whereas TS-DP effectively mitigates this issue, achieving a $3.8\times$ NFE speedup with only a 2% accuracy drop. The task closely matches the difficulty of the official simulation and real-world benchmarks, providing a clear setting to showcase the acceleration achieved by our method.

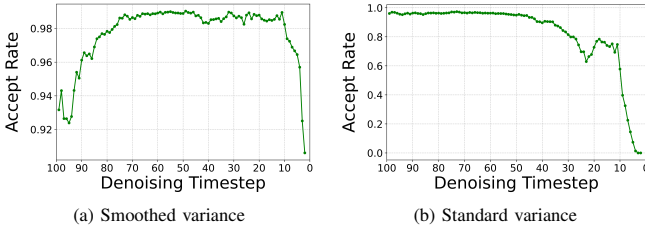


Fig. 6. **Impact of Drafter Parameters on Draft Acceptance Rate.** Evolution of the Metropolis-Hastings acceptance probability throughout the 100-step denoising process.

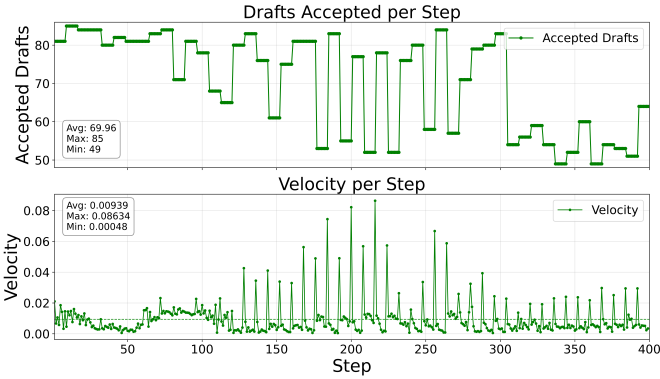


Fig. 7. **Effect of End-Effector Velocity on the Number of Accepted Drafts.** Variation of Accepted Drafts and Velocity During the Can-ph Task: (Top) Accepted Drafts per Step; (Bottom) Velocity per Step.

$$r_{\text{final}_{\text{con}}} = 2 \times R_{\text{final}} \times r_{\text{max}} - R_{\text{final}}, \quad (13)$$

where R_{final} is a hyperparameter. This distinction enables us to differentiate between completion-based tasks and binary success-failure tasks.

To transform the sparse rewards into dense ones, we design the intermediate reward as an efficiency metric:

$$r_{\text{process}}^{(i)} = \left(\frac{n_{\text{accept}}}{n_{\text{draft}}} + \frac{n_{\text{accept}}}{n_{\text{diffusion}}} \right) \times \lambda, \quad (14)$$

where n_{accept} , n_{draft} and $n_{\text{diffusion}}$ denote the number of accepted drafts, the total number of drafts, and the number of diffusion steps per denoising cycle, respectively. λ is a scaling factor that constrains the accumulated process reward to one-fourth of the final reward, encouraging the scheduler to pursue efficiency while maintaining accuracy:

$$\lambda = \frac{R_{\text{final}}/4}{N_{\text{expected}}}, \quad N_{\text{expected}} = \left\lceil \frac{T_{\text{max}}}{\Delta t} \right\rceil, \quad (15)$$

where T_{max} is the max task steps, Δt denotes the parameter update frequency.

Furthermore, to reduce computational overhead, the scheduler is executed in parallel with the encoder. As the encoder is significantly larger in scale, the scheduler incurs no additional time cost.

IV. EXPERIMENTS

A. Experimental Settings

Simulation protocol. We evaluate TS-DP on Robomimic, Push-T, Multimodal Block Pushing, and Kitchen. These benchmarks cover a broad range of manipulation skills: the Robomimic suite contains Lift (raising a cube above a height threshold), Can (pick-and-place of a soda can into a target bin), Square (inserting a square nut onto a peg with precise alignment), Transport (a bimanual task that hands an object between two arms and places it in a target bin), and Tool Hang (high-precision frame assembly and tool hanging); Push-T pushes a T-shaped block to a fixed target pose; Multimodal Block Pushing pushes two blocks into two target squares; and Franka Kitchen requires completing a sequence of household subtasks, testing long-horizon multi-stage control. Following the official Diffusion Policy protocol, we use the official checkpoints and training seeds, and evaluate each checkpoint with 50 environment initializations. Success rate is used for most tasks, while target-area coverage is used for Push-T and Block Pushing. In the tables, “ x/y ” denotes results from two official checkpoints.

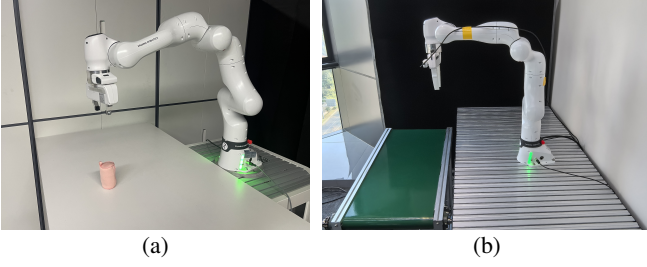


Fig. 8. **Real-world experimental platforms.** The Pick-and-Release task is conducted on platform (a), while the Conveyor-Belt Pick-and-Place and Pen-to-Holder Insertion tasks are conducted on platform (b).

TABLE IV
RESULTS ON REAL TASKS.

Method	Success Rate (% , $\uparrow$) / NFE ($\downarrow$)			AVG
	Pick	Conveyor Belt	Pen Holder	
DP [1]	82 / 100	29 / 100	31 / 100	47 / 100
TS-DP	80 / 26	52 / 27	29 / 29	54 / 27

As task difficulty increases, the acceleration gain of TS-DP decreases; moreover, the speed improvement is particularly beneficial for real-time conveyor-belt tasks, leading to a 23% increase in success rate.

Baselines and implementation. We compare against DP [1], Frozen Target Draft [19], SpeCa [39], and BAC [13] using the same target backbone, checkpoints, denoising steps, hardware, and evaluation conditions. Simulation experiments run on an NVIDIA A100 GPU with 40 GB memory. The scheduler takes observations, recent actions, and task progress as input, and adjusts the sigma scale, acceptance threshold λ , and three-stage draft budget K . The static ablation uses $K \in \{10, 25, 40\}$.

Real-world setup. We use a Franka Research 3 arm and an Intel RealSense D435i camera in an eye-to-hand setting. Polymetis executes low-level control on a local NVIDIA RTX 4090 workstation, while the remote Diffusion Policy communicates through a socket interface. Each task is evaluated over 100 trials with randomized object placement within the reachable workspace. Failures mainly arise near workspace boundaries or under illumination changes. The experimental platforms are shown in Fig. 8.

Real-world tasks. We design three tasks of increasing difficulty. *Pick-and-Release* requires the arm to detect a pink cylinder, grasp it, lift it to a preset height, and release it; its difficulty is comparable to the official simulation and real-robot benchmarks. *Conveyor-Belt Pick-and-Place* requires grasping a bun from a moving conveyor belt without stopping the belt and placing it onto a plate; this task is highly latency-sensitive and directly stresses inference speed. *Pen-to-Holder Insertion* requires picking a pen from the table and inserting it vertically into a narrow-mouth holder that is not fixed and can be displaced by contact, a fine-manipulation task that challenges precision.

B. Main Results

Here, we measure efficiency using Number of Function Evaluations (NFE) and inference speedup, where each drafter evaluation counts as 1/8 NFE and each target-model evaluation

counts as 1 NFE. NFE is a widely adopted efficiency metric in diffusion acceleration research [52], [53]. Because DP inference is memory-bandwidth-bound, its latency is dominated by repeated sequential model calls rather than raw computation; TS-DP reduces these sequential target calls through speculative generation and parallel verification. We therefore use NFE to characterize sequential sampling cost and do not interpret its reduction as a proportional reduction in FLOPs or MACs; the corresponding wall-clock gains are reported separately in the Latency Analysis. Here, “near-lossless” refers only to empirically comparable success rates or target-area coverage relative to the original DP, rather than exact distributional equivalence. Tables I and II present the acceleration performance of TS-DP across different benchmarks. On the PH dataset, TS-DP achieves a $4.17\times$ inference speedup with 76% reduction in NFE while achieving higher success rates of 85 / 80 compared to the baseline DP’s 76 / 80. On the more challenging MH dataset, our method achieves a $3.84\times$ speedup with 76% NFE reduction while achieving success rates of 75 / 84 compared to the baseline’s 76 / 81. For complex multi-stage tasks, Table III shows consistent acceleration benefits. On the Kitchen benchmark and Block Pushing tasks, TS-DP achieves a $3.70\times$ speedup with 73% NFE reduction while maintaining high performance, with an average success rate of 99 / 99 across subtasks.

Besides, we conduct real-world experiments under 3 settings: static pick-and-place, conveyor-belt pick-and-place, and placing a pen into a narrow-mouth holder. The quantitative results are reported in Table IV, while qualitative visualizations are provided in Figs. 3, 4, and 5. On the simplest Pick-and-Release task, we achieve a $3.8\times$ NFE speedup. For the real-time-critical conveyor-belt picking task, TS-DP improves the success rate by 23% while maintaining a $3.7\times$ NFE speedup, demonstrating that improved real-time performance is crucial for dynamic tasks. Even on the most challenging fine-manipulation task, TS-DP attains a $3.4\times$ NFE speedup, indicating its ability to adapt to task difficulty while preserving acceleration gains.

Action Fidelity. To further characterize the near-lossless behavior beyond task success, we measure action-level fidelity between DP and TS-DP. Under the same observation, official checkpoint, initial noise, and random seed, we generate paired action chunks and compute the normalized root mean squared error (NRMSE) and cosine similarity in the normalized action space. As shown in Table VI, NRMSE stays below 0.05 and cosine similarity above 0.93 on all six tasks, with averages of 0.0318 and 0.9574, respectively. This confirms that TS-DP closely reproduces the target policy’s actions in practice, supporting empirical fidelity rather than exact distributional equivalence.

C. Ablation Study

Latency Analysis. We first empirically justify the block-equivalent NFE definition. The target backbone contains eight Transformer blocks whereas the drafter contains one block of the same width; profiling both under identical inputs (Table VIII) yields average latencies of 9.37 ms and 1.17 ms,

TABLE V
FREQUENCY AND LATENCY EVALUATION OF TS-DP. EACH ENTRY REPORTS FREQUENCY (Hz, $\uparrow$) / LATENCY (s, $\downarrow$), WITH LATENCY GIVEN AS MEAN $\pm$ STANDARD DEVIATION OVER REPEATED TRIALS.

Method	Frequency (Hz, $\uparrow$) / Latency (s, $\downarrow$)			AVG
	Lift	Can	Square	
DP [1]	7.75/0.129 $\pm$ 0.004	7.52/0.133 $\pm$ 0.003	7.52/0.133 $\pm$ 0.004	DP: 8.15/0.124
TS-DP	24.39/0.041 $\pm$ 0.002	24.39/0.041 $\pm$ 0.002	23.81/0.042 $\pm$ 0.003	
Method	Frequency (Hz, $\uparrow$) / Latency (s, $\downarrow$)			TS-DP: 26.28/0.039
	Transport	Tool	Push-T	
DP [1]	7.69/0.130 $\pm$ 0.005	8.13/0.123 $\pm$ 0.003	10.31/0.097 $\pm$ 0.003	
TS-DP	24.39/0.041 $\pm$ 0.002	25.00/0.040 $\pm$ 0.002	35.71/0.028 $\pm$ 0.001	

Compared with DP, TS-DP increases the average control frequency by $3.2\times$, from 8.15 Hz to 26.28 Hz, while reducing the average latency from 0.124 s to 0.039 s. On the simplest Push-T task, the control frequency further reaches 35.71 Hz, well above the requirement for real-time control.

TABLE VI
EMPIRICAL ACTION-LEVEL FIDELITY BETWEEN DP AND TS-DP, REPORTED AS MEAN $\pm$ STANDARD DEVIATION.

Task	NRMSE $\downarrow$	Cosine Similarity $\uparrow$
Can	0.0324 $\pm$ 0.0015	0.9527 $\pm$ 0.0082
Lift	0.0384 $\pm$ 0.0018	0.9455 $\pm$ 0.0091
Square	0.0434 $\pm$ 0.0020	0.9387 $\pm$ 0.0095
Transport	0.0454 $\pm$ 0.0021	0.9304 $\pm$ 0.0103
Tool	0.0101 $\pm$ 0.0005	0.9958 $\pm$ 0.0021
Push-T	0.0213 $\pm$ 0.0010	0.9814 $\pm$ 0.0020
Average	0.0318	0.9574

Fidelity is slightly lower on contact-rich tasks such as Square and Transport, mirroring the small success-rate differences above; the consistently low NRMSE and high cosine similarity indicate that TS-DP reproduces the target policy’s actions rather than merely matching final task outcomes.

TABLE VII
DRAFTER-ONLY RESULTS ON THE MIXED HUMAN (MH) DEMONSTRATION DATASET.

Method	Success Rate (% , $\uparrow$)				AVG
	Lift	Can	Square	Transport	
DP [1]	99 / 100	92 / 97	76 / 79	35 / 46	75.5 / 80.5
Drafter Only	36 / 40	26 / 20	0 / 0	0 / 0	15.5 / 15.0
TS-DP	99 / 99	92 / 97	74 / 78	34 / 62	74.75 / 84

Using only the Drafter causes severe performance degradation, especially on harder tasks, whereas TS-DP maintains high accuracy through speculative decoding and the PPO scheduler.

an $8.01\times$ ratio that supports counting one drafter call as $1/8$ target-equivalent NFE. This is an architecture-normalized measure of sampling cost and is reported separately from the wall-clock results below.

We further evaluate the practical inference latency of TS-DP, measuring the wall-clock time required for single action prediction across different robotic manipulation tasks. As reported in Table V, the baseline DP requires 0.124s per action prediction, limiting its applicability in high-frequency control scenarios. Our TS-DP framework reduces this latency to 0.039s, achieving a $3.2\times$ speedup while maintaining identical or better task performance. Notably, TS-DP achieves an

TABLE VIII
BLOCK-LEVEL LATENCY (MS) OF THE TARGET BACKBONE AND THE DRAFTER, AVERAGED OVER REPEATED RUNS.

Model	Lift	Can	Square	Transport	Tool	Push-T	Avg
DP	10.03	9.57	10.14	10.46	7.88	8.15	9.37
Drafter	1.22	1.19	1.19	1.25	1.15	1.02	1.17
Ratio	8.22$\times$	8.04$\times$	8.52$\times$	8.37$\times$	6.85$\times$	7.99$\times$	8.01$\times$

average prediction frequency of 26.28 Hz compared to DP’s 8.15 Hz, well exceeding the minimum requirement for real-time robotic control. The gain is also consistent across tasks of different difficulty: on simpler and more predictable tasks such as Push-T, the control frequency can be increased to 35.71 Hz, while on more challenging manipulation settings the system still maintains roughly 24 Hz operation. This indicates that TS-DP does not merely provide a fixed acceleration factor, but instead adapts its speculative behavior to the temporal complexity of each task phase, allocating more aggressive parallel decoding when the denoising trajectory is stable and becoming more conservative near precision-critical moments. Overall, this latency profile enables TS-DP to meet the timing requirements of real-time robotic control without compromising decision quality, making it suitable for practical deployment on edge devices with limited computational resources.

Drafter Analysis. We further evaluate a Drafter-only baseline to determine whether the gains of TS-DP originate solely from replacing DP with a lightweight model. As shown in Table VII, using only the Drafter leads to a severe performance drop, and the degradation becomes more pronounced as task difficulty increases. Even on the relatively simple Lift task, the success rate falls to around 40%, while on more challenging tasks such as Square and Transport it collapses to 0%. These results indicate that the Drafter alone cannot provide reliable embodied control due to its limited representational capacity and weaker robustness at critical manipulation stages. This behavior is expected in embodied control, where task success often depends on a few key moments such as grasp closure, object alignment, or contact-rich transitions; once model capacity is reduced, these fine-grained decisions become much less reliable. As task difficulty increases, the weaker generalization

TABLE IX
PERFORMANCE COMPARISON WITH AND WITHOUT THE SCHEDULER
ON THE MIXED HUMAN (MH) DEMONSTRATION DATASET.

Method	Success Rate (% , $\uparrow$)				AVG	Speed $\times$
	Lift	Can	Square	Transport		
$K = 10$	100	94	76	66	84	2.45
$K = 25$	88	88	62	60	75	3.47
$K = 40$	74	90	64	60	72	3.92
TS-DP	100	96	84	66	87	3.80

Fixed draft counts expose a clear speed–accuracy trade-off, whereas the adaptive scheduler achieves the best overall success rate while retaining high acceleration.

ability of the compact model further amplifies this issue, causing errors that cannot be corrected once execution enters a critical stage. In contrast, TS-DP preserves the accuracy of the full policy, showing that its effectiveness comes from the combination of the Drafter, speculative verification, and adaptive scheduling rather than from model shrinking alone. **Effectiveness of Adaptive Scheduling.** To validate the effectiveness of our scheduler, we compare TS-DP with static speculative decoding settings that keep the draft step count K fixed throughout denoising. As shown in Table IX, static configurations inevitably produce a trade-off between task success and acceleration: conservative decoding ($K = 10$) yields stronger accuracy but limited speedup, while aggressive decoding ($K = 40$) accelerates more but significantly hurts performance. In contrast, our adaptive scheduler achieves the best average success rate of 87% while still reaching a $3.80\times$ speedup. Compared with the best fixed setting, the adaptive policy improves the average success rate by 3 points over $K = 10$ while preserving nearly the same acceleration regime as the more aggressive settings. This result suggests that a single static draft budget is fundamentally mismatched to embodied manipulation: stable phases can benefit from larger speculative batches, whereas contact-rich or alignment-sensitive phases require smaller budgets and stricter verification. Building on this observation, temporally varying embodied tasks cannot be effectively handled with a fixed speculative configuration. In contrast, our RL-based scheduler dynamically adjusts the decoding behavior to balance efficiency and reliability across different task phases. Fig. 9 visualizes this behavior on the Square task, showing how the scheduler varies the effective variance (Sigma), the acceptance threshold, and the high-, intermediate-, and low-noise draft budgets over execution steps. It enlarges the intermediate-stage budget during stable phases and tightens the acceptance threshold while shrinking the high- and low-noise budgets during interaction-critical phases, confirming that the PPO scheduler adapts to task phase rather than using a fixed configuration.

Results Using Speed Threshold. We further replace the PPO scheduler with a velocity-threshold heuristic following SP-VLA to test whether a hand-crafted rule is sufficient for adaptive control. As reported in Table X, this heuristic lowers the average success rate to 68%, which is 11 points below TS-DP, and also yields consistently weaker draft acceptance across tasks. This result indicates that using velocity alone

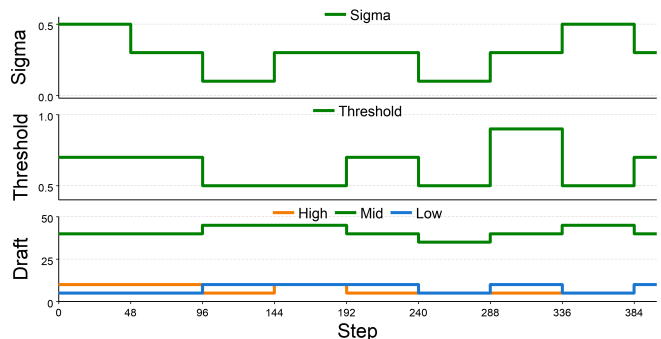


Fig. 9. **Temporal scheduling dynamics on the Square task.** The scheduler adjusts the effective variance (Sigma), acceptance threshold, and the high-, intermediate-, and low-noise draft budgets over execution steps.

TABLE X
RESULTS USING A SPEED THRESHOLD ON THE MIXED HUMAN (MH)
DEMONSTRATION DATASET.

Method	Success Rate / Draft Accept Rate (% , $\uparrow$)				AVG	NFE ($\downarrow$)
	Lift	Can	Square	Transport		
DP [1]	99.5 / –	94.5 / –	77.5 / –	40.5 / –	78.0 / –	100
VT [21]	90 / 80	86 / 80	61 / 79	35 / 81	68 / 80	36
TS-DP	99 / 84.2	94.5 / 87.9	76 / 86.0	48 / 89.6	79 / 86.9	26

VT means using a velocity threshold heuristic for scheduling [21]. Replacing the PPO scheduler with a velocity-threshold heuristic reduces both average task success and acceptance quality, despite still lowering NFE.

captures only coarse task progress and cannot reliably identify critical moments that require higher precision. In practice, motion speed may already be high when the robot is about to enter a delicate alignment or contact stage, so a heuristic that keeps acceleration aggressive at that moment can easily induce unrecoverable errors. By contrast, the PPO scheduler in TS-DP leverages richer multi-modal observations to infer task phase more accurately, enabling stronger end-to-end acceleration without sacrificing task completion.

Visualizations. Figs. 3, 4, and 5 visualize the three real-world tasks considered in our evaluation. For example, Fig. 3 illustrates the conveyor-belt pick-and-place task, where objects are grasped from a moving conveyor belt and placed onto a plate. In this case, DP operates significantly slower than TS-DP and exhibits prolonged pauses after grasping, resulting in a total completion time of 63s, whereas TS-DP completes the task in 20s. Moreover, faster inference and more responsive interaction with the environment enable TS-DP to increase the success rate by 23%. As shown in Fig. 10, TS-DP yields significantly higher draft acceptance rates and more efficient parallel computation compared to static methods. Table XI further summarizes these statistics: as task complexity grows, the scheduler reduces the draft count and NFE rises from 24.0 to 27.2 while the acceptance rate stays above 85%, indicating that TS-DP adjusts its speculative intensity to task difficulty rather than using a fixed configuration. The red curves (with scheduling) consistently maintain higher acceptance while generating more drafts, demonstrating optimal trade-off between acceleration and accuracy. More importantly, the visualization reveals a clear phase-dependent pattern: during stable

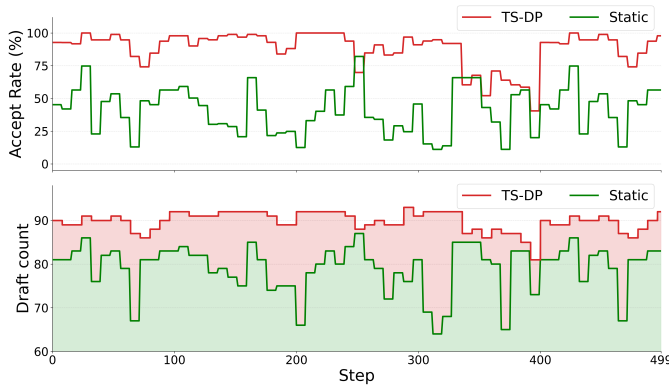


Fig. 10. **Effect of TS-DP on Acceptance Rate and Draft Count.** (Top) Acceptance Rate; (Bottom) Draft Count. With the integration of TS-DP, more drafts are accepted while maintaining task accuracy.

TABLE XI

SPECULATIVE DECODING STATISTICS ACROSS BENCHMARKS. WE REPORT NFE, SPEEDUP, DRAFT COUNT, AND ACCEPTANCE RATE ON THE PH, MH, AND KITCHEN/BP BENCHMARKS.

Benchmark	NFE	Speed	Draft Count	Accept. Rate
PH	24.0	4.17×	93.9	85.2%
MH	26.3	3.83×	92.5	87.1%
Kitchen/BP	27.2	3.77×	91.8	89.8%

execution segments, the scheduler increases the draft budget to maximize parallelism, while in interaction-critical periods it reduces the number of drafts and tightens the decision criterion to avoid error accumulation. This confirms that fixed-parameter speculative decoding is limited in embodied tasks with time-varying complexity, while our framework achieves superior end-to-end performance through intelligent hyperparameter adaptation.

D. Training Cost and Reusability

TS-DP directly uses a frozen official DP checkpoint and trains a corresponding drafter and PPO scheduler for each checkpoint. The drafter is target-checkpoint specific, while the scheduler is target-policy-and-environment specific; changing the checkpoint, task distribution, environment, or robot embodiment therefore requires retraining or fine-tuning. Since the base DP itself does not support cross-task or cross-embodiment transfer, we make no additional generalization claim beyond the capability of the base model, and the real-robot experiments likewise use acceleration modules trained for the corresponding real-robot checkpoint.

All acceleration modules are trained offline on a single NVIDIA A40 GPU. Training the lightweight drafter uses 2,728 MiB of GPU memory and about 75 s per epoch for 150 epochs, totaling roughly 3.13 hours. Training the PPO scheduler over 4,032 environment timesteps takes about 2.75 hours and uses 12,992 MiB of GPU memory. Training both modules sequentially therefore requires approximately 5.88 hours with a peak memory of 12,992 MiB. This one-time offline cost excludes training of the official DP checkpoint and is not included in the reported online inference latency, so

it is amortized over repeated deployments of the accelerated policy.

V. CONCLUSION

In this work, we present TS-DP, a temporal-aware speculative decoding framework that accelerates Diffusion Policy by combining efficient draft generation with adaptive scheduling. A distilled Transformer-based drafter replaces expensive base-model denoising calls across multiple steps, while an RL-based scheduler dynamically adjusts speculative parameters based on task difficulty to maintain accuracy and maximize efficiency. Extensive experiments across 4 manipulation benchmarks demonstrate up to **4.17×** faster inference, a **76%** NFE reduction, and real-time operation at **26 Hz** while maintaining comparable task performance in simulation. Real-world experiments further show up to a **3.8×** NFE reduction and improved success rates on real-time tasks (e.g., a **23%** gain on the conveyor-belt task).

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant No. 92467204, 62472249, 62571298, and 62402264), and in part by the Shenzhen Science and Technology Program (Grant No. KJZD20240903102300001).

REFERENCES

- [1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *Int. J. Robotics Res.*, vol. 44, pp. 1684–1704, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:257378658>
- [2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, “ π_0 : A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [3] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, “ $\pi_{0.5}$: A vision-language-action model with open-world generalization,” *arXiv preprint arXiv:2504.16054*, 2025.
- [4] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, “Octo: An open-source generalist robot policy,” *arXiv preprint arXiv:2405.12213*, 2024.
- [5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.
- [6] H. Chen, J. Liu, C. Gu, Z. Liu, R. Zhang, X. Li, X. He, Y. Guo, C.-W. Fu, S. Zhang *et al.*, “Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning,” *arXiv preprint arXiv:2506.01953*, 2025.
- [7] Y. Liu, Y. Liu, Y. Meng, J. Zhang, Y. Zhou, Y. Li, J. Jiang, K. Ji, S. Ge, Z. Wang *et al.*, “Spatial policy: Guiding visuomotor robotic manipulation with spatial-aware modeling and reasoning,” *arXiv preprint arXiv:2508.15874*, 2025.
- [8] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, “Rdt-1b: a diffusion foundation model for bimanual manipulation,” *arXiv preprint arXiv:2410.07864*, 2024.
- [9] L. Chen, Y. Meng, C. Tang, X. Ma, J. Jiang, X. Wang, Z. Wang, and W. Zhu, “Q-dit: Accurate post-training quantization for diffusion transformers,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 28 306–28 315.
- [10] B. Wan, T. Zheng, Z. Chen, Y. Wang, and J. Wang, “Pruning for sparse diffusion models based on gradient flow,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

- [11] D. Zhang, S. Li, C. Chen, Q. Xie, and H. Lu, "Laptop-diff: Layer pruning and normalized distillation for compressing diffusion models," *arXiv preprint arXiv:2404.11098*, 2024.
- [12] X. Li, Y. Liu, L. Lian, H. Yang, Z. Dong, D. Kang, S. Zhang, and K. Keutzer, "Q-diffusion: Quantizing diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 535–17 545.
- [13] K. Ji, Y. Meng, H. Cui, Y. Li, S. Hua, L. Chen, and Z. Wang, "Block-wise adaptive caching for accelerating diffusion policy," *ArXiv*, vol. abs/2506.13456, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:279402462>
- [14] Y. Yang, Y. Wang, Z. Wen, L. Zhongwei, C. Zou, Z. Zhang, C. Wen, and L. Zhang, "Efficientvla: Training-free acceleration and compression for vision-language-action models," *arXiv preprint arXiv:2506.10100*, 2025.
- [15] T. Cai, Y. Li, Z. Geng, H. Peng, and T. Dao, "Medusa: Simple framework for accelerating llm generation with multiple decoding heads," 2023.
- [16] Y. Li, F. Wei, C. Zhang, and H. Zhang, "Eagle: Speculative sampling requires rethinking feature uncertainty," *arXiv preprint arXiv:2401.15077*, 2024.
- [17] —, "Eagle-2: Faster inference of language models with dynamic draft trees," *arXiv preprint arXiv:2406.16858*, 2024.
- [18] —, "Eagle-3: Scaling up inference acceleration of large language models via training-time test," *arXiv preprint arXiv:2503.01840*, 2025.
- [19] V. D. Bortoli, A. Galashov, A. Gretton, and A. Doucet, "Accelerated diffusion models via speculative sampling," *ArXiv*, vol. abs/2501.05370, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:275405770>
- [20] H. Hu, A. Das, D. Sadigh, and N. Anari, "Diffusion models are secretly exchangeable: Parallelizing ddpm via autospeculation," *ArXiv*, vol. abs/2505.03983, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:278368005>
- [21] Y. Li, Y. Meng, Z. Sun, K. Ji, C. Tang, J. Fan, X. Ma, S. Xia, Z. Wang, and W. Zhu, "Sp-vla: A joint model scheduling and token pruning approach for vla model acceleration," *arXiv preprint arXiv:2506.12723*, 2025.
- [22] S. Weng, J. Zhang, T. Zhu, L. Yu, and C. Zhang, "Dcm-net: A diffusion model-based detection network integrating the characteristics of copy-move forgery," *IEEE Transactions on Multimedia*, vol. 27, pp. 503–514, 2024.
- [23] Y. Jiang, Q. Liu, D. Chen, L. Yuan, and Y. Fu, "Animediff: Customized image generation of anime characters using diffusion model," *IEEE Transactions on Multimedia*, vol. 26, pp. 10 559–10 572, 2024.
- [24] Y. Xu, X. Xu, H. Gao, and F. Xiao, "Sgdm: An adaptive style-guided diffusion model for personalized text to image generation," *IEEE Transactions on Multimedia*, vol. 26, pp. 9804–9813, 2024.
- [25] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos, "Active policy learning for robot planning and exploration under uncertainty," in *Robotics: Science and Systems*, 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2077953>
- [26] Y. Li, C. Tang, Y. Meng, J. Fan, Z. Chai, X. Ma, Z. Wang, and W. Zhu, "Prance: Joint token-optimization and structural channel-pruning for adaptive vit inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–17, 2025.
- [27] X. Pei, Y. Chen, S. Xu, Y. Wang, Y. Shi, and C. Xu, "Action-aware dynamic pruning for efficient vision-language-action manipulation," *arXiv preprint arXiv:2509.22093*, 2025.
- [28] Z. Yu, B. Wang, P. Zeng, H. Zhang, J. Zhang, L. Gao, J. Song, N. Sebe, and H. T. Shen, "A survey on efficient vision-language-action models," *arXiv preprint arXiv:2510.24795*, 2025.
- [29] W. Guan, Q. Hu, A. Li, and J. Cheng, "Efficient vision-language-action models for embodied manipulation: A systematic survey," *arXiv preprint arXiv:2510.17111*, 2025.
- [30] Y. Li, Z. Liu, G. Lan, M. Sader, and Z. Chen, "A ddpg-based solution for optimal consensus of continuous-time linear multi-agent systems," *Science China Technological Sciences*, vol. 66, no. 8, pp. 2441–2453, 2023.
- [31] Z. Liu, Y. Li, G. Lan, and Z. Chen, "A novel data-driven model-free synchronization protocol for discrete-time multi-agent systems via td3 based algorithm," *Knowledge-Based Systems*, vol. 287, p. 111430, 2024.
- [32] Q. Zeng, C. Hu, M. Song, and J. Song, "Diffusion model quantization: A review," *arXiv preprint arXiv:2505.05215*, 2025.
- [33] H. Chang, H. Shen, Y. Cai, X. Ye, Z. Xu, W. Cheng, K. Lv, W. Zhang, Y. Lu, and H. Guo, "Effective quantization for diffusion models on cpus," *arXiv preprint arXiv:2311.16133*, 2023.
- [34] Y. He, L. Liu, J. Liu, W. Wu, H. Zhou, and B. Zhuang, "Ptqd: Accurate post-training quantization for diffusion models," *arXiv preprint arXiv:2305.10657*, 2023.
- [35] J. Ye, Z. Wang, and L. Jiang, "Pqd: Post-training quantization for efficient diffusion models," in *Proceedings of the Winter Conference on Applications of Computer Vision*, 2025, pp. 150–156.
- [36] Y. Shang, Z. Yuan, B. Xie, B. Wu, and Y. Yan, "Post-training quantization on diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 1972–1981.
- [37] H. Zhou, D. Blessing, G. Li, O. Celik, X. Jia, G. Neumann, and R. Lioutikov, "Variational distillation of diffusion policies into mixture of experts," *ArXiv*, vol. abs/2406.12538, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:270562714>
- [38] F. Liu, S. Zhang, X. Wang, Y. Wei, H. Qiu, Y. Zhao, Y. Zhang, Q. Ye, and F. Wan, "Timestep embedding tells: It's time to cache for video diffusion model," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 7353–7363.
- [39] J. Liu, C. Zou, Y. Lyu, F. Ren, S. Wang, K. Li, and L. Zhang, "Speca: Accelerating diffusion transformers with speculative feature caching," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 10 024–10 033.
- [40] C. Zou, X. Liu, T. Liu, S. Huang, and L. Zhang, "Accelerating diffusion transformers with token-wise feature caching," *arXiv preprint arXiv:2410.05317*, 2024.
- [41] C. Zou, E. Zhang, R. Guo, H. Xu, C. He, X. Hu, and L. Zhang, "Accelerating diffusion transformers with dual feature caching," *arXiv preprint arXiv:2412.18911*, 2024.
- [42] S. Zheng, G. Chen, Q. Zhou, Y. Lin, L. He, C. Zou, P. Cai, J. Liu, and L. Zhang, "Let features decide their own solvers: Hybrid feature caching for diffusion transformers," *arXiv preprint arXiv:2510.04188*, 2025.
- [43] Y. Yang, Y. Wang, Z. Wen, Z. Luo, C. Zou, Z. Zhang, C. Wen, and L. Zhang, "Efficientvla: Training-free acceleration and compression for vision-language-action models," *ArXiv*, vol. abs/2506.10100, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:279318363>
- [44] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *International Conference on Machine Learning*, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:254096365>
- [45] C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. M. Jumper, "Accelerating large language model decoding with speculative sampling," *ArXiv*, vol. abs/2302.01318, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:256503945>
- [46] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *International Conference on Machine Learning*. PMLR, 2023, pp. 19 274–19 286.
- [47] C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. Jumper, "Accelerating large language model decoding with speculative sampling," *arXiv preprint arXiv:2302.01318*, 2023.
- [48] S. Kim, K. Mangalam, S. Moon, J. Malik, M. W. Mahoney, A. Gholami, and K. Keutzer, "Speculative decoding with big little decoder," *Advances in Neural Information Processing Systems*, vol. 36, pp. 39 236–39 256, 2023.
- [49] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [50] R. Furuta, N. Inoue, and T. Yamasaki, "Pixelrl: Fully convolutional network with reinforcement learning for image processing," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1704–1719, 2019.
- [51] W. Xu, J. Yu, Z. Miao, L. Wan, Y. Tian, and Q. Ji, "Deep reinforcement polishing network for video captioning," *IEEE Transactions on Multimedia*, vol. 23, pp. 1772–1784, 2020.
- [52] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 5775–5787.
- [53] Z. Wang, Z. Li, A. Mandlekar, Z. Xu, J. Fan, Y. S. Narang, L. Fan, Y. Zhu, Y. Balaji, M. Zhou, M.-Y. Liu, and Y. Zeng, "One-step diffusion policy: Fast visuomotor policies via diffusion distillation," *ArXiv*, vol. abs/2410.21257, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:273654186>