Fast-dVLA: Accelerating Discrete Diffusion VLA to Real-Time Performance

Wenxuan Song[1*], Jiayi Chen[1*], Shuai Chen[3,4*], Jingbo Wang[1], Pengxiang Ding[5,6], Han Zhao[5,6], Yikai Qin[1], Xinhu Zheng[1], Donglin Wang[5], Yan Wang[2(✉)], and Haoang Li[1(✉)]

[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] AIR, Tsinghua University
[3] ShanghaiTech University
[4] Shanghai Institute of Technical Physics, CAS
[5] Westlake University
[6] Zhejiang University
haoangli@hkust-gz.edu.cn

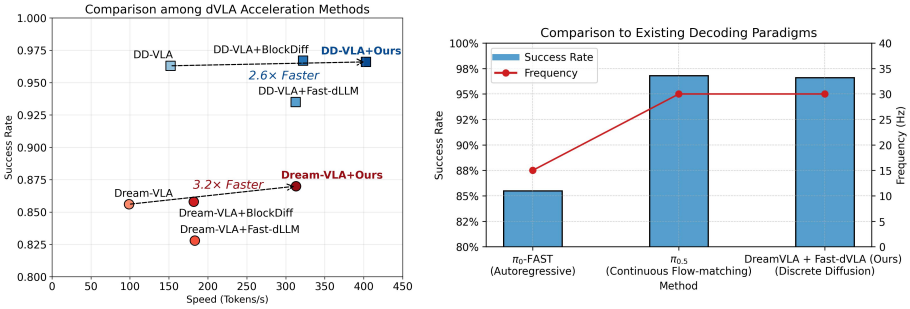


Fig. 1: Speed/Success Rate trade-off. Left (Intra-comparison): Compared to other acceleration strategies for discrete diffusion VLAs (dVLAs) methods – DD-VLA [15] and Dream-VLA [43], our Fast-dVLA achieves a favorable success rate and speed. Here, BlockDiff denotes block diffusion [1]. **Right (Inter-comparison):** Our Fast-dVLA surpasses autoregressive methods, *i.e.*, π_0-FAST [23]. It also reaches parallel performance and inference frequency with state-of-the-art (SOTA) continuous flow-matching methods, *i.e.*, $\pi_{0.5}$ [8], while maintaining several inherent advantages of dVLAs. We report the metrics on LIBERO [16].

Abstract. Vision–Language–Action (VLA) models have become a compelling route toward robotic foundation models by mapping multimodal perception and language instructions directly into executable actions. Beyond the prevalent flow-matching VLAs, discrete diffusion VLA (dVLA) models further unify perception and control in a single discrete space, offering inherent advantages in unified multimodal alignment and understanding, while better preserving the prior knowledge of vision-language models. Yet this strength comes with a practical obstacle that inference is far too slow for real-time control, as the parallel decoding prevents the utilization of Key-Value (KV) cache. In this paper, we aim to tackle this

* Equal contribution. ✉ Corresponding author.

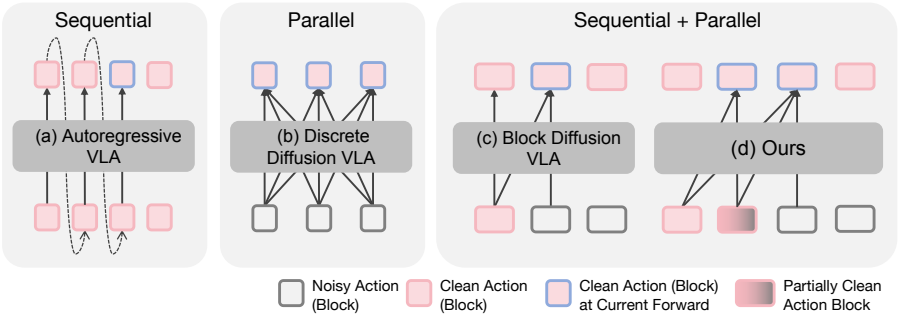
challenge by leveraging an intriguing observation that the dVLA with bidirectional attention still adheres to a block-wise left-to-right order, which motivates the application of block diffusion. However, directly applying block diffusion precludes inter-block parallelism, a key acceleration factor. To address the limitation, we introduce Fast-dVLA, an block-wise diffusion acceleration strategy. Fast-dVLA takes the full action token sequence at each timestep as an action block and denoises them together. It then decodes different blocks in an autoregressive manner to allow KV cache reusing, while allowing inter-block parallel decoding in a diffusion-forcing manner. For efficient training, we directly conduct an asymmetric distillation on a finetuned dVLA. Extensive evaluations on Discrete Diffusion VLA, Dream-VLA, and UD-VLA across diverse simulated benchmarks validate that our Fast-dVLA achieves 2.8×-4.1× speedup and maintains state-of-the-art performance. Moreover, the results on real-world high-dynamics tasks demonstrate the potential of real-time deployment and application of our Fast-dVLA.

Keywords: Vision-language-action Models · Block Diffusion · Decoding Acceleration

1 Introduction

Vision–Language–Action (VLA) models are typically trained on large-scale robotic datasets to map multimodal perception into executable robotic control, exhibiting language following and visual generalization capabilities, and have become a dominant paradigm in current research on robotic foundation models. Representative VLAs [2, 3, 18] adopt a flow-matching architecture, where the VLM performs multimodal understanding and the flow matching action head takes the processed representations as input and outputs continuous control signals. Recently, discrete diffusion VLAs (dVLAs) based on the diffusion Large Language Models (dLLMs) [6, 15, 35, 37, 44] have emerged as a promising challenger to existing VLA architectures. These models output actions in a parallel, iterative denoising manner, while not relying on the flow-matching head. Therefore, compared with flow-matching architectures, they demonstrate inherent advantages in unified multimodal alignment and understanding [6, 35], while better preserving the pretrained knowledge of VLMs [15].

However, current dVLAs still suffer from a fundamental limitation. Their inference speed is slow, with an execution frequency that is far below the real-time requirements of physical robotic systems (typically around 30 Hz). This large gap substantially limits their practical applicability in real-world settings. As illustrated in Figure 2, although dVLA significantly reduces the number of forward passes required to generate a complete action sequence compared to discrete autoregressive (AR) VLA (Figure 2 (a)) by enabling parallel decoding, its bidirectional attention mechanism prevents the reuse of key–value (KV) caches from previously generated tokens, resulting in a very low per-pass forward efficiency (Figure 2 (b)).



Model	(a) Autoregressive VLA	(b) Discrete Diffusion VLA	(c) Block Diffusion VLA	(d) Ours
Forwards per Sequence	High	Low	Moderate	Low
Forward Speed	Fast	Low	Fast	Fast
Speed per Sequence (Inference Speed)	Slow	Slow	Moderate	Fast

Fig. 2: Comparison among discrete decoding paradigms. Here, *Forward per Sequence* denotes the needed forward numbers for a full sequence output, *Forward Speed* denotes the decoding speed for each forward, and *Speed per Sequence (i.e., Inference Speed)* denotes the decoding speed for the full sequence output. **Our Fast-dVLA requires significantly fewer forward passes and executes each pass efficiently, resulting in substantially faster inference.**

To explore the feasibility of leveraging KV cache, we investigate the action decoding order in dVLAs (Figure 3). We observe that, although with bidirectional attention, dVLAs still follow a left-to-right decoding pattern. This block-wise decoding behavior suggests that a promising direction is to apply block diffusion [1] (Figure 2 (c)), which natively trains dVLAs with block-wise attention, decodes a block of action tokens in parallel, caches the corresponding KV states after completing the block, and then proceeds to the next block in an AR manner. This design achieves a moderate inference speed by balancing partial KV cache reuse with intra-block parallel decoding. However, it inherently precludes inter-block parallelism, which is a crucial factor for achieving high-throughput and low-latency inference.

This paper proposes Fast-dVLA, a novel block-wise diffusion strategy that achieves the first breakthrough in accelerating dVLAs to a real-time regime. Conceptually, we exploit block-wise sequential generation for KV cache utilization, while removing the requirement that later blocks wait for earlier ones to finish denoising. Concretely, we treat the full action token sequence at each timestep (*i.e.*, the dimensionality of the actions) and its multiples as an action block. Then, Fast-dVLA learns to denoise a sequence of blocks with monotonically increasing mask ratios in parallel. Naturally, preceding blocks can finish before subsequent ones, allowing their KV states to be cached for subsequent computations. Note that we constrain the attention to be block-wise causal to ensure the KV cache remains unchanged. For training efficiency, inspired by [33], we distill Fast-dVLA from finetuned dVLAs with bidirectional attention using an asymmetric distillation

loss. During inference, we design a pipelined parallel decoding algorithm that enables inter-block parallelism with varying noise levels across blocks.

We conduct extensive experiments on representative dVLA models, including Dream-VLA [43], Discrete Diffusion VLA (DD-VLA) [15], and UD-VLA [6] across CALVIN [20], LIBERO [16], and SIMPLER [14] benchmarks. Figure 1 shows that our method consistently achieves 2.8×–4.1× speedup while preserving action performance, which is superior to other dVLA paradigms. Lastly, diverse real-world tasks demonstrate the dynamic capability and working efficiency in the application.

Our contributions are summarized as follows:

- We reveal an implicit block-wise AR decoding tendency in the fully bidirectional dVLA, motivating an AR–diffusion hybrid denoising process.
- We propose Fast-dVLA, which leverages block-wise diffusion with a corresponding attention pattern to allow KV cache reuse, while allowing inter-block parallelism through diffusion forcing.
- Following the observation, we apply an asymmetric distillation for efficient training, and a pipelined parallel decoding for real-time inference.
- Extensive experiments on CALVIN, LIBERO, and SIMPLER demonstrate up to 4.1× acceleration over existing dVLA models, while maintaining SOTA-level success rates. Moreover, the results in diverse real-world tasks demonstrate the dynamic capability and working efficiency in the application.

2 Related Works

Discrete Diffusion VLA (dVLA). In this paper, we extend the notion of dLLM [46] to the embodied domain and define dVLA, a VLA model that enables parallel decoding of multiple action tokens (and optionally tokens from other modalities) through an iterative denoising-based inference procedure. PD-VLA [30] first adopts Jacobi decoding to enable AR VLA to predict in parallel action tokens without training. Then, DD-VLA [15] and LLADA-VLA [37] follow the BERT-style [7] masked prediction strategy, where selected action tokens are replaced with a special mask token, and the model directly learns to predict the original tokens at these masked positions. Dream-VLA [43] performs a large-scale robotic pretraining on a diffusion vision-language model to inject embodied capabilities. UD-VLA [6] and dVLA [35] integrate visual CoT or textual CoT into discrete diffusion–based VLA models and jointly diffuse future frames, textual reasoning traces, and actions within a single unified framework. However, they overlook the bottleneck in inference speed, thereby leaving a gap in real-world applications.

Acceleration of VLA. Recent efforts of efficient VLA focus on pruning for redundancy reduction. MoLe-VLA [49] dynamically activates layers via a Mixture-of-Layers design. EfficientVLA [42] proposes a training-free acceleration framework combining layer pruning, token selection, and diffusion caching. ADP [22] and LightVLA [9] introduce action-aware and differentiable token pruning strategies,

respectively, to reduce visual redundancy while maintaining performance. Beyond pruning, early-exit strategies have also been explored to reduce cost. DeeR-VLA [47] adaptively adjusts the effective model depth, while CEED-VLA [29] enables early termination over iterative steps during inference. Another effective method is caching. VLA-Cache [40], for example, improves efficiency by caching static tokens and recomputing only task-dependent components. Similarly, CronusVLA [12] proposes feature-level token caching via a FIFO queue, decoupling expensive single-frame perception from lightweight multi-frame reasoning. Researchers are also exploring novel architectures and optimization techniques. RoboMamba [19] integrates Mamba state-space modeling into the VLA framework to achieve efficient robotic reasoning. In parallel, quantization techniques, such as those proposed by BitVLA [32] and QVLA [41], enable the deployment of VLA models on hardware with limited resources by using low-bit representations. Finally, the development of lightweight backbones from the ground up offers a direct path to efficiency. TinyVLA [36] pursues this by designing compact architectures from scratch. Flower [27] proposes an efficient 950M-parameter diffusion-based VLA that uses intermediate-modality fusion and action-specific conditioning. Meanwhile, SmolVLA [28] combines application-level pruning with spatial compaction through pixel rearrangement. However, these works do not specifically address the acceleration of dVLA. In contrast, our work systematically investigates acceleration strategies tailored to the unique characteristics of dVLA, thereby filling this gap.

3 Preliminary: Discrete Diffusion VLA (dVLA)

Discrete Diffusion VLA (*e.g.*, **Dream-VLA** [43] and **DD-VLA** [15]) output discrete action tokens, obtained either by uniform bins [11] or by quantized tokenizers [23], instead of operating directly on continuous controls. Actions are represented as a length-L discrete token sequence $\mathbf{a}_0 = (a_0^i, \ldots, a_0^L)$, where each token a_0^i corresponds to discrete low-level robot actions and a special mask token M is added to the vocabulary to enable diffusion-style corruption.

The forward diffusion process randomly replaces a subset of action tokens with M according to a time-dependent mask ratio, independently across positions. The reverse process learns to recover masked tokens conditioned on the unmasked context and multimodal inputs $\mathbf{c}$ (*e.g.*, language and visual observations). At each denoising step, unmasked tokens are copied unchanged, while masked positions are predicted from a categorical distribution parameterized by the model.

During training, a mask ratio $\gamma_t \in (0, 1]$ is sampled, and the corresponding action tokens are replaced by M to obtain a corrupted sequence $\tilde{\mathbf{a}}_t$. The model is then trained to reconstruct the original tokens using cross-entropy loss computed only on masked positions:

$$\mathcal{L}_{\text{act}}(\theta) = -\sum_{i \in \mathcal{M}_{\gamma_t}} \log p_\theta\big(a_0^i \mid \tilde{\mathbf{a}}_t, \mathbf{c}\big)\,, \tag{1}$$

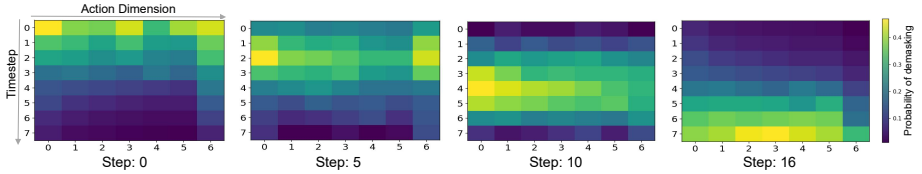


Fig. 3: Visualization of the decoding tendency of action tokens at different positions in Dream-VLA [43]. Brighter regions indicate higher decoding probability. Despite using bidirectional attention, the model exhibits a clear left-to-right decoding tendency such that action tokens at earlier temporal positions are typically decoded in earlier diffusion iterations. Overall, the decoding process reveals an implicit block-wise AR pattern.

where $\mathcal{M}_{\gamma_t}$ denotes the set of masked positions. This objective preserves the core corruption–denoising principle of discrete diffusion while enabling efficient training with standard discrete VLA architectures.

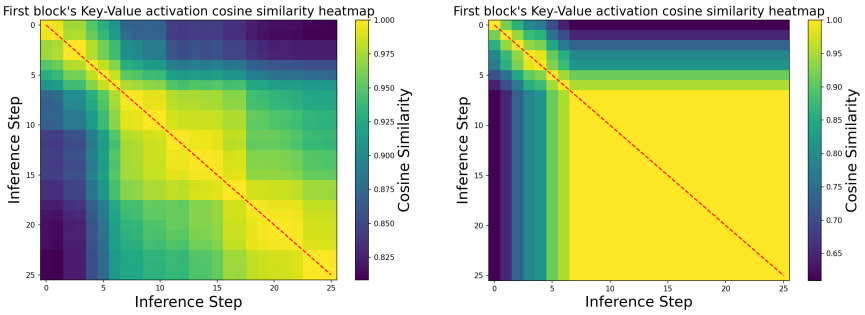
UD-VLA [6] extends this framework to unified VLA models [34] by incorporating future visual prediction. Specifically, future image observations are encoded into a discrete token sequence $\mathbf{v}_0 = (v_0^1, \ldots, v_0^L)$ using a VQ-VAE [51] encoder, and concatenated with the action tokens to form a unified sequence. The diffusion process is then applied jointly over visual and action tokens, allowing future visual reasoning and action generation to be learned in a unified manner.

4 Method

In this section, we first introduce an intriguing observation (see Section 4.1). Then, we propose Fast-dVLA to accelerate dVLA in a block-wise decoding manner. Our method is built to support two key features: (1) a block-wise attention mechanism that enables the reuse of KV cache across denoising iterations (see Section 4.2) and (2) a diffusion forcing denoising process that supports simultaneous decoding of blocks with different noising levels. To efficiently train such models, we design an asymmetric distillation that starts from a pretrained bidirectional dVLA (see Section 4.3). During inference, we design an inter-block parallel decoding schedule that balances inference speed and decoding reliability (see Section 4.4).

4.1 Motivation

As shown in Figure 3, we record and visualize the decoding frequency at different positions during the denoising process of a representative dVLA (*i.e.,* Dream-VLA). Interestingly, even though the dVLA employs bidirectional attention, the model still exhibits a strong left-to-right decoding pattern at a global level. In particular, action blocks that occur earlier in the temporal dimension tend to be decoded in earlier denoising iterations. This can be attributed to: 1) The backbone [44] of existing dVLAs is typically initialized from an AR VLM and trained in a discrete diffusion manner, thereby retaining certain autoregressive characteristics. 2) Actions at different timesteps exhibit inherent temporal dependencies. This block-wise AR decoding behavior suggests that *a finetuned*



(a) Bidirectional Attention **(b)** Block-wise Attention

Fig. 4: KV cache similarity across diffusion iterations under block-diffusion decoding. We visualize the similarity of attention key–value (KV) states for the first action block across different denoising steps. **(a):** In native dVLA with bidirectional attention, the KV representations evolve across iterations, preventing effective reuse of cached states. **(b):** In contrast, after adapting dVLA to a block-wise attention architecture via asymmetric distillation, once all tokens in the first block are unmasked, the corresponding KV states remain fixed, enabling efficient KV cache reuse and substantially reducing the computational overhead in subsequent iterations.

bidirectional dVLA can be directly forced to follow a block-diffusion decoding manner.

4.2 Critical Designs of Target Models

Block-Wise Attention for Inter-block KV Cache Reusing. As shown in Figure 4a, current dVLA [6,15,43] models generate either partial or full sequences using bidirectional attention, which causes the Key-Value (KV) representations to vary at every denoising iteration. As a result, the conventional KV cache mechanism used in AR models cannot be directly reused to accelerate inference. To address this limitation, we adopt a block diffusion decoding strategy (see Figure 2d) with block-wise attention (see Figure 5), which bridges autoregressive decoding and discrete diffusion by interpolating between sequential dependency and parallel generation. Within each block, the KV representations are influenced only by the prefix tokens and the tokens inside the current block. As shown in Figure 4b, once the decoding of a block is completed, the KV values of that block remain unchanged in subsequent steps, enabling effective cache reuse for the following decoding process.

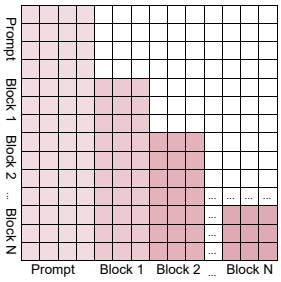


Fig. 5: Block-wise Attention.

Diffusion Forcing for Inter-block Parallel Decoding. Motivated by observations in mimic-video [21] that action tokens need not attend to the clean tokens from previous timesteps, we construct a progressively decaying noise sequence similar to diffusion forcing [5,13,45]. Let the index

set of the i-th block be defined as $B_i = \{(i-1)k, \ldots, ik-1\}$, and denote by Y_{B_i} the corresponding token subsequence.

During the forward diffusion process, we assign progressively increasing noise levels to different blocks according to a monotonic schedule $t_1 < t_2 < \cdots < t_N$. Formally, the noise sequence can be represented as $Y^{t_{1:N}} = \{Y_{B_1}^{t_1}, \ldots, Y_{B_N}^{t_N}\}$. Under this design, earlier blocks are exposed to lower corruption levels and thus retain more complete information, whereas later blocks remain more heavily masked and uncertain. For the reverse process, we learn a θ-parameterized model that factorizes the conditional distribution in a block-wise autoregressive manner:

$$p_\theta(Y^0 \mid Y^{t_{1:N}}) = \prod_{i=1}^{N} p_\theta\big(Y_{B_i}^0 \mid Y_{B_1}^{t_1}, \ldots, Y_{B_i}^{t_i}\big)\,. \tag{2}$$

This formulation allows the model to progressively refine earlier blocks while concurrently denoising later ones, naturally enabling parallel decoding across blocks without sacrificing temporal consistency.

4.3 Training: Asymmetric Distillation for Efficient Post-Training

To train our Fast-dVLA, a straightforward approach is to train it from scratch while maintaining block-wise attention and diffusion-forcing objective. The loss function is defined as:

$$\mathcal{L}_{\text{BD}} = \mathbb{E} \sum_{i=1}^{N} \Big[-\log p_\theta(Y_{B_i}^0 | Y_{B_{<i}}^{t_{<i}}, c) \Big]\,. \tag{3}$$

However, motivated by Section 4.1, directly inheriting the decoding nature from open-source bidirectional dVLAs [6, 15, 50] (serving as teacher models) can be a more efficient and lower-cost way. Specifically, inspired by [33], we design an asymmetric distillation in which the Fast-dVLA (serving as student models) with block-wise attention is forced to align with the output of the teacher model with bidirectional attention, while they share the same architecture and both condition on the blocks with a monotonic noise schedule. Thus, the distillation loss is formulated as:

$$\mathcal{L}_{\text{AD}} = \mathbb{E}\left[\sum_{i=1}^{N} D_{\text{KL}}\left(p_\theta(Y_{B_i}^0 | Y_{B_{<=i}}^{t_{<=i}}, c) \| p_{\phi^-}(Y_{B_i}^0 | Y_{B_{<=N}}^{t_{<=N}}, c) \right) \right], \tag{4}$$

where D_{KL} represents the KL divergence aggregated over the mask tokens. The distillation is asymmetric in that the teacher p_{ϕ^-} predicts for each block $Y_{B_i}^0$ with a global view of all blocks, while the student p_θ learns to approximate using only a causally restricted view.

Taking the training budget required for training a dVLA from scratch as the reference, Figure 8 shows that asymmetric distillation from finetuned weight ($\mathcal{L}_{\text{AD}}$ in Equation (4)) achieves convergence with only 1/10 steps, which is much more efficient than training with $\mathcal{L}_{\text{BD}}$ on the base of finetuned weight or from scratch. Thus, we adopt asymmetric distillation as the default training objective.

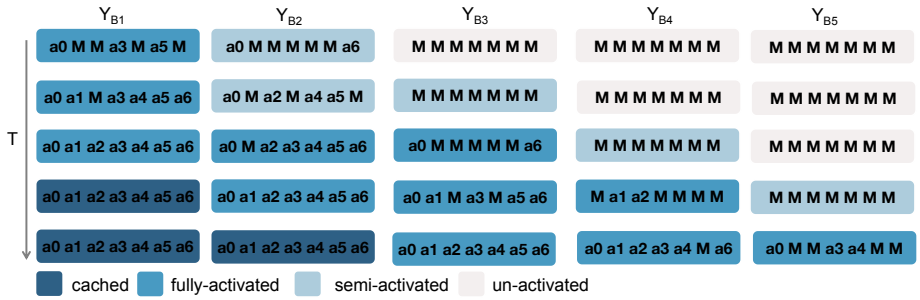


Fig. 6: Overview of the pipelined parallel decoding in Fast-dVLA. Blocks are processed concurrently in a dynamically growing pipeline. A new block is introduced when the tail block exceeds the addition threshold $\tau_{\text{add}} = \frac{2}{7}$, and becomes fully activated after its predecessor surpasses the activation threshold $\tau_{\text{act}} = \frac{4}{7}$.

4.4 Inference: Pipelined Parallel Decoding

As shown in Figure 6, in contrast to traditional block diffusion [1], which performs parallel decoding only within each block while strictly decoding different blocks in sequence, our method enables parallel prediction across multiple blocks.

Specifically, we distinguish activated blocks (i.e., the block currently being decoded) into two states: *semi-activated* and *fully-activated*. The transition between these states is governed by the completion ratio of the preceding block with respect to the thresholds τ_{add} and τ_{act}. When the completion ratio of the previous block exceeds τ_{add}, the subsequent block is introduced as a semi-activated block. We adopt the confidence-aware decoding strategy [38] to selectively decode tokens whose prediction confidence exceeds the threshold τ_{conf}. Once the completion ratio surpasses τ_{act}, the block transitions to a fully-activated state, in which at least $1/n$ of the remaining tokens are guaranteed to be decoded at each step according to confidence ranking.

This multi-state block-parallel decoding mechanism achieves a favorable trade-off between efficiency and performance. At the same time, it ensures that earlier action tokens are decoded in early iterations, thereby preserving the temporal causality inherent in action execution. A pseudocode summary of our inference is available in the supplementary material.

5 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of Fast-dVLA in both simulated and real-world robot manipulation tasks. The experiments are designed to answer five core questions:

(RQ1) Does our Fast-dVLA achieve a favorable performance/speed trade-off among all dVLA acceleration paradigms? Furthermore, is Fast-dVLA consistently effective across diverse dVLA architectures (see Section 5.2)?

(RQ2) How does the performance of existing dVLA methods accelerated by Fast-dVLA compare to SOTA methods (*i.e.*, flow-matching VLAs) on diverse benchmarks and tasks (see Section 5.3)?

Table 1: Comparison between various acceleration strategies on two base models in terms of task-wise success rates (SR) and inference speed on LIBERO.

Decoding Method	Success Rate ↑					Speed ↑
	Spatial	Goal	Object	Long	Avg.	(Tokens/s)
Dream-VLA [43]	0.902	0.920	0.880	0.720	0.856	98.8 (×1.0)
+ Fast-dLLM	0.884	0.894	0.834	0.702	0.828	183.2 (×1.9)
+ Block Diffusion	0.918	0.904	0.886	0.722	0.858	181.7 (×1.8)
+ Fast-dVLA (ours)	0.912	0.920	0.902	0.746	0.870	**313.1** (×3.2)
Discrete Diffusion VLA [15]	0.972	0.986	0.974	0.920	0.963	152.1 (×1.5)
+ Fast-dLLM	0.940	0.952	0.948	0.898	0.935	312.5 (×3.2)
+ Block Diffusion	0.976	0.986	0.972	0.932	0.967	322.1 (×3.3)
+ Fast-dVLA (ours)	0.970	0.988	0.976	0.928	0.966	**402.7** (×4.1)

(RQ3) Could Fast-dVLA facilitate the real-world tasks (see Section 5.4)?
(RQ4) Is the training of Fast-dVLA efficient (see Section 5.5)?
(RQ5) What empirical insights can guide the selection of hyperparameters for Fast-dVLA to ensure optimal performance (see Section 5.6) ?

5.1 Setup

Models. We select Dream-VLA and DD-VLA as representatives of dVLA models, and UD-VLA as the representative of unified dVLA models. For Dream-VLA, we perform distillation for 4k steps, corresponding to approximately 1/5 of the original fine-tuning budget. For DD-VLA, we set the distillation steps to 4k, which is approximately 1/8 of the original fine-tuning steps. We set the block size to be 7, as analyzed in Section 5.6. For UD-VLA, we perform distillation for 3k steps, corresponding to approximately 1/8 of the original UD-VLA fine-tuning steps, with a batch size of 12. Due to the relatively long output sequences (625 tokens) in UD-VLA, we set the block size to be a multiple of 32. All remaining training hyperparameters follow the original model configurations.
Benchmarks. We conduct extensive simulated experiments on three popular benchmarks (CALVIN [20], LIBERO [16], and SimplerEnv [14]) to provide comprehensive results. Detailed introduction of these benchmarks is available in the supplementary material.

5.2 Paradigm Comparison (RQ1)

Fast-dVLA achieves obvious acceleration with competitive performance. Table 1 shows that Fast-dLLM [38] realizes 2× speedup with an obvious performance drop. This proves that directly reusing the KV cache under fully bidirectional attention introduces biased keys and values in the attention states that lead to performance degradation (see Figure 4a). Block Diffusion [1] decodes blocks strictly in sequence, thus also obtaining a limited acceleration. In contrast, our Fast-dVLA built on Dream-VLA and DD-VLA both achieve speedups up to 4.1×, while slightly improving performance compared to their base models.

Table 2: Comparison between various acceleration strategies on UD-VLA in terms of tasks completed in a row, average sentence length, and inference speed on CALVIN.

Decoding Method	Task	Tasks Completed in a Row ↑					Avg.	Speed ↑
		1/5	2/5	3/5	4/5	5/5	Len. ↑	(Tokens/s)
UD-VLA	ABCD→D	0.992	0.968	0.936	0.904	0.840	**4.64**	67.3 (×1.0)
+ Fast-dLLM	ABCD→D	0.972	0.920	0.858	0.808	0.762	4.32	132.5 (×2.0)
+ Block Diffusion	ABCD→D	0.988	0.944	0.894	0.862	0.804	4.50	129.5 (×1.9)
+ Fast-dVLA (ours)	ABCD→D	0.984	0.952	0.922	0.870	0.812	4.54	**186.7** (×2.8)

We attribute the efficiency gain to the discrete diffusion forcing denoising, which effectively reuses the KV cache and offers inter-block parallelism, increasing throughput per iteration step. Meanwhile, our block-wise attention preserves temporal causality in action prediction, leading to more stable optimization during training. In addition, the results further demonstrate the effectiveness of our acceleration strategy across different methods on the same dataset.

Fast-dVLA naturally generalizes to unified dVLA architectures. As shown in Table 2, Fast-dVLA achieves a 2.8× inference speedup over UD-VLA on the long-horizon CALVIN ABCD-D benchmark, while maintaining superior performance. This result demonstrates that our method can be seamlessly extended to unified dVLA frameworks that generate visual foresights together with actions to serve as a process of chain of thought, highlighting its adaptability in accelerating multimodal generation and action prediction.

5.3 Comparison with SOTA (RQ2)

Accelerating UD-VLA on CALVIN. Table 3 shows that compared with world-modeling VLA [48] that jointly generate future images and actions, our Fast-dVLA applied on UD-VLA inherits the strong representational capacity and the advantages of a unified multimodal latent space, while mitigating the long decoding latency induced by future image tokens through our acceleration technique. These results demonstrate that the unified dVLA paradigm can serve as a practical and viable solution for VLAs with a world-modeling process.

Accelerating Dream-VLA on SimplerEnv. On the SimplerEnv, which more closely reflects real-world robotic evaluation settings with high visual fidelity, our results are consistent with those observed on CALVIN. Table 4 shows that our Fast-dVLA achieves the highest decoding speeds among all VLAs with discrete outputs, including AR paradigms (OpenVLA and π_0-FAST), vanilla dVLA paradigms (DD-VLA and LLaDA-VLA), and block diffusion paradigms (Dream-VLA with Fast-dLLM or Block Diffusion). This improvement stems from the effective combination of KV caching and inner/inter-block parallelism.

In terms of task success rates, benefiting from the superior cross-modal alignment of dVLAs, our Fast-dVLA outperforms continuous flow-matching approaches such as GR00T-N1 and π_0. Furthermore, leveraging the sequential action representation induced by our method and the large-scale robot pretraining of Dream-VLA, our Fast-dVLA also surpasses existing dVLA methods.

Table 3: Comprehensive evaluation of long-horizon manipulation on the CALVIN benchmark. UniVLA* denotes the variant without historical frames for fair comparison.

Method	Task	Tasks Completed in a Row					Avg. Len. ↑
		1/5	2/5	3/5	4/5	5/5	
RT-1 [4]	ABCD→D	0.844	0.617	0.438	0.323	0.227	2.45
LLaDA-VLA [37]	ABCD→D	0.956	0.878	0.795	0.739	0.645	4.01
Deer [47]	ABCD→D	0.982	0.902	0.821	0.759	0.670	4.13
GR-1 [39]	ABCD→D	0.949	0.896	0.844	0.789	0.731	4.21
ReconVLA [31]	ABCD→D	0.980	0.900	0.845	0.785	0.705	4.23
UniVLA* [34]	ABCD→D	0.948	0.906	0.862	0.834	0.690	4.26
MODE [25]	ABCD→D	0.971	0.925	0.879	0.835	0.779	4.39
UP-VLA [48]	ABCD→D	0.962	0.921	0.879	0.842	0.812	4.42
MDT [26]	ABCD→D	0.986	0.958	0.916	0.862	0.801	4.52
UD-VLA + Fast-dVLA (ours)	ABCD→D	0.984	0.952	0.922	0.870	0.812	**4.54**

Table 4: Evaluation on WidowX Robot tasks in SimplerEnv. We report the Grasping Success Rate (Grasp) and Task Success Rate (Success) in percentages (%). Besides, we further report the decode speed (Speed) of all VLAs with discrete output, which is calculated in tokens per second.

Method	Spoon on Towel		Carrot on Plate		Stack Green Block		Eggplant in Basket		Average	
	Grasp	Success	Grasp	Success	Grasp	Success	Grasp	Success	Success	Speed
RoboVLM [17]	37.5	20.8	33.3	25.0	8.3	8.3	0.0	0.0	13.5	-
SpatialVLA [24]	20.8	16.7	29.2	25.0	62.5	29.2	**100.0**	**100.0**	42.7	-
OpenVLA-OFT [10]	50.0	12.5	41.7	4.2	70.8	20.8	91.7	37.5	18.8	-
π_0 [3]	45.8	29.1	25.0	0.0	50.0	16.6	91.6	62.5	27.1	-
π_0-FAST [23]	62.5	29.1	58.5	21.9	54.0	10.8	83.3	66.6	32.1	107.5
GR00T-N1 [2]	**83.3**	**62.5**	54.2	45.8	70.8	16.7	41.7	20.8	36.5	-
DDVLA [15]	70.8	29.2	58.3	29.2	62.5	20.8	91.7	70.8	37.5	152.8
LLaDA-VLA [37]	-	56.9	-	**76.3**	-	30.6	-	58.3	55.5	160.0
Dream-VLA [43]	79.2	45.8	62.5	45.8	**83.3**	25.0	**100.0**	87.5	51.0	100.1
+ Fast-dLLM [38]	70.8	41.7	54.2	37.5	70.8	20.8	83.3	66.6	41.7	214.2
+ Block Diffusion [1]	**83.3**	54.1	**66.7**	45.8	**83.3**	29.1	95.8	91.6	55.2	226.4
+ Fast-dVLA (ours)	**83.3**	54.1	62.5	54.1	**83.3**	**37.5**	**100.0**	91.6	**59.3**	**366.4**

For the comprehensive comparison on LIBERO, please refer to the supplementary materials.

5.4 Real-world Experiments (RQ3)

Setup. Real-world experiments were conducted on a bimanual AgileX platform, where each 6-DOF arm is equipped with a gripper. The sensory suite includes a high-mounted overhead camera providing a global perspective and two wrist-mounted cameras for localized views.

Task setting. We designed three distinct tasks, as illustrated in Figure 7: (1) *Conveyor Picking*, which involves picking blocks from a moving conveyor belt and placing them into a tray. (2) *Vegetables Stowing*, which requires sorting vegetables based on their text labels in a container. (3) *Vegetables Retrieving*, which involves grasping a target vegetable and placing it into a pot according to specific language instructions. For each task, we collected 100 expert demonstrations for training. For evaluation, we conducted 40 trials per task, recording the success rate and the average completion time. Specifically, for the conveyor belt task, we utilized the number of successful grasps per minute as the primary evaluation metric to

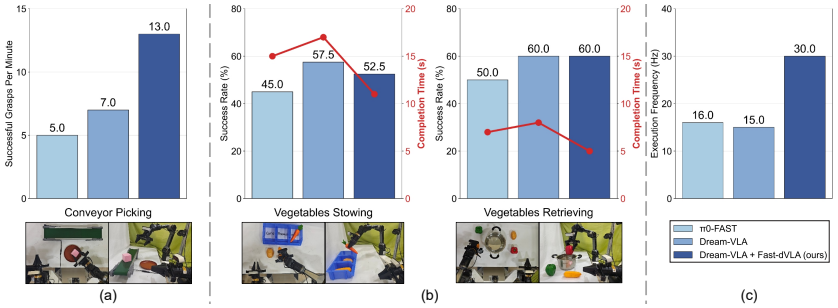


Fig. 7: Real-world experiment results. We report (a) successful grasps per minute (↑); (b) success rates (↑) and completion times (↓); (c) execution frequency (↑).

quantify performance. In addition, we recorded the execution frequency on the real-world robot platform to quantify the real-time performance.

Results. We evaluate our method against two representative models: π_0-FAST [23] (the SOTA AR VLA) and Dream-VLA [43], a representative dVLA that acts as our base model. Figure 7 shows our model demonstrates robust performance across all three tasks. Notably, the conveyor belt picking task requires both precise grasping and real-time responsiveness. Our method achieves nearly double the efficiency of previous approaches, closely aligned with the practical demands of industrial sorting systems. In the remaining two tasks that require semantic understanding, our model maintains competitive performance with only a marginal reduction in success rate relative to the baseline, while further shortening the required completion time. Crucially, our system maintains a consistent execution frequency of 30 Hz across all tasks, satisfying the practical demand of real-time control that other approaches fail to meet. These results underscore our model's efficient execution and precise instruction-following capabilities.

5.5 Training Efficiency (RQ4)

To evaluate the training efficiency of our Fast-dVLA, we compare four training strategies based on Dream-VLA on LIBERO: (1) *Asymmetric Distillation from Finetuned Weight ($\mathcal{L}_{\text{AD}}$).* This approach distills our Fast-dVLA from the task-specific finetuned weight using $\mathcal{L}_{\text{AD}}$. (2) *Training from Finetuned Weight ($\mathcal{L}_{\text{BD}}$).* This approach trains Fast-dVLA from the finetuned weight with $\mathcal{L}_{\text{BD}}$. (3) *Training From Scratch ($\mathcal{L}_{\text{BD}}$).* This approach trains our Fast-dVLA from the pretrained dVLA. (4) *Training From Scratch ($\mathcal{L}_{\text{act}}$).* This approach finetunes a normal dVLA on the specific tasks to serve as the baseline for comparison.

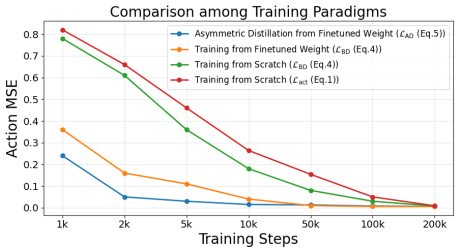


Fig. 8: Action Mean Squared Error (MSE) of the dVLA at varying training steps on LIBERO. The MSE of our asymmetric distillation exhibits the fastest decline, indicating the most rapid convergence speed.

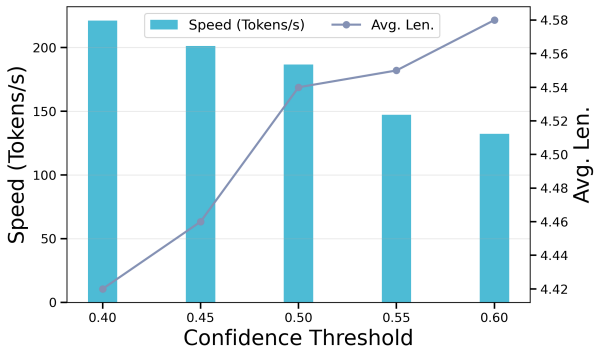


Fig. 9: Ablation study on the confidence threshold τ_{conf} of Fast-dVLA based on UD-VLA.

Figure 8 shows that our asymmetric distillation demonstrates significantly superior training efficiency. Notably, the distillation strategy (blue line) requires only 2,000 training steps to converge, which is 5× faster than continuing to train from the finetuned weight (orange line) and approximately 1/10 of the steps needed for training from scratch (green line). From another perspective, our asymmetric distillation offers a cost-efficient pathway to accelerate existing dVLA models to real-time performance required for practical applications. Besides, all strategies for training our Fast-dVLA converge faster than finetuning a normal dVLA (red line), denoting that our architecture is more efficient.

5.6 Ablation Studies (RQ5)

Block size aligned with action dimensionality brings better performance. We validate the importance of choosing the multiples of the action dimensionality as the block size. To ensure fairness, the results are averaged from several values between the sizes of 7 (action token numbers in one step) and 14. Specifically, we find that this choice better maintains success rates and speedup, demonstrating its coherence with the action architecture that better keeps the intrinsic temporal dependencies of the action tokens.

Table 5: Choice of block size on LIBERO-Long based on DreamVLA. Here, multiples denotes that the value is the multiples of the dimensionality of action, while random denotes random numbers for the block size.

Block	Success Rate	Speedup
Multiples	74.7%	4.01×
Ramdom	73.3%	3.95×

Ablation of τ_{conf}. For the confidence threshold τ_{conf} in the semi-activated block, lowering the threshold yields an approximately linear drop in performance while improving inference speed. As shown in Figure 9, we set the confidence threshold to 0.5 to balance these two factors, achieving a 2.8× acceleration while incurring a marginal performance drop of 2%.

6 Conclusion

In this paper, we tackle key limitations in the inference speed of dVLAs. Specifically, we reveal an implicit block-wise AR decoding tendency in the fully bidirectional dVLA. Thus, we propose Fast-dVLA, which leverages block-wise diffusion with a corresponding attention pattern to allow KV cache reuse, while allowing inter-block parallelism through diffusion forcing. We also curate an efficient training process and a pipelined inference for real-time inference. Extensive experiments on simulated benchmarks and real-world tasks demonstrate up to 4.1× acceleration over existing dVLA models, while maintaining SOTA-level success rates. These findings offer a practical solution for deploying dVLAs as competitive alternatives to continuous flow-matching VLAs in real-world applications.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant 62403401, in part by the National Science and Technology Innovation 2030 - Major Project under Grant 2022ZD0208800, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011992 and Grant 2026A1515012323, and in part by the Guangdong Provincial Project under Grant 2024QN11X127.

References

1. Arriola, M., Gokaslan, A., Chiu, J.T., Yang, Z., Qi, Z., Han, J., Sahoo, S.S., Kuleshov, V.: Block diffusion: Interpolating between autoregressive and diffusion language models. arXiv preprint arXiv:2503.09573 (2025)
2. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
3. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
4. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. Robotics: Science and Systems Foundation (2023)
5. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2024)
6. Chen, J., Song, W., Ding, P., Zhou, Z., Zhao, H., Tang, F., Wang, D., Li, H.: Unified diffusion vla: Vision-language-action model via joint discrete denoising diffusion process. arXiv preprint arXiv:2511.01718 (2025)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)

8. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: pi0.5: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
9. Jiang, T., Jiang, X., Ma, Y., Wen, X., Li, B., Zhan, K., Jia, P., Liu, Y., Sun, S., Lang, X.: The better you learn, the smarter you prune: Towards efficient vision-language-action models via differentiable token pruning. arXiv preprint arXiv:2509.12594 (2025)
10. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
11. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., et al.: Openvla: An open-source vision-language-action model. In: Conference on Robot Learning. pp. 2679–2713. PMLR (2025)
12. Li, H., Yang, S., Chen, Y., Tian, Y., Yang, X., Chen, X., Wang, H., Wang, T., Zhao, F., Lin, D., et al.: Cronusvla: Transferring latent motion across time for multi-frame prediction in manipulation. arXiv preprint arXiv:2506.19816 (2025)
13. Li, L., Zhang, Q., Luo, Y., Yang, S., Wang, R., Han, F., Yu, M., Gao, Z., Xue, N., Zhu, X., et al.: Causal world modeling for robot control. arXiv preprint arXiv:2601.21998 (2026)
14. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., Xiao, T.: Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941 (2024)
15. Liang, Z., Li, Y., Yang, T., Wu, C., Mao, S., Nian, T., Pei, L., Zhou, S., Yang, X., Pang, J., et al.: Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. arXiv preprint arXiv:2508.20072 (2025)
16. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. arXiv preprint arXiv:2306.03310 (2023)
17. Liu, H., Li, X., Li, P., Liu, M., Wang, D., Liu, J., Kang, B., Ma, X., Kong, T., Zhang, H.: Towards generalist robot policies: What matters in building vision-language-action models. arXiv preprint arXiv:2412.14058 (2025)
18. Liu, J., Chen, H., An, P., Liu, Z., Zhang, R., Gu, C., Li, X., Guo, Z., Chen, S., Liu, M., et al.: Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. arXiv preprint arXiv:2503.10631 (2025)
19. Liu, J., Liu, M., Wang, Z., Lee, L., Zhou, K., An, P., Yang, S., Zhang, R., Guo, Y., Zhang, S.: Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. arXiv e-prints pp. arXiv–2406 (2024)
20. Mees, O., Hermann, L., Rosete-Beas, E., Burgard, W.: Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters (RA-L) **7**(3), 7327–7334 (2022)
21. Pai, J., Achenbach, L., Montesinos, V., Forrai, B., Mees, O., Nava, E.: mimic-video: Video-action models for generalizable robot control beyond vlas. arXiv preprint arXiv:2512.15692 (2025)
22. Pei, X., Chen, Y., Xu, S., Wang, Y., Shi, Y., Xu, C.: Action-aware dynamic pruning for efficient vision-language-action manipulation. arXiv preprint arXiv:2509.22093 (2025)
23. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747 (2025)

24. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al.: Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830 (2025)
25. Reuss, M., Pari, J., Agrawal, P., Lioutikov, R.: Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In: The Thirteenth International Conference on Learning Representations (2025)
26. Reuss, M., Yağmurlu, Ö.E., Wenzel, F., Lioutikov, R.: Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. Robotics: Science and Systems XX. Ed.: D. Kulic (2024)
27. Reuss, M., Zhou, H., Rühle, M., Yağmurlu, Ö.E., Otto, F., Lioutikov, R.: Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. arXiv preprint arXiv:2509.04996 (2025)
28. Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., et al.: Smolvla: A vision-language-action model for affordable and efficient robotics. arXiv preprint arXiv:2506.01844 (2025)
29. Song, W., Chen, J., Ding, P., Huang, Y., Zhao, H., Wang, D., Li, H.: Ceed-vla: Consistency vision-language-action model with early-exit decoding. arXiv preprint arXiv:2506.13725 (2025)
30. Song, W., Chen, J., Ding, P., Zhao, H., Zhao, W., Zhong, Z., Ge, Z., Ma, J., Li, H.: Accelerating vision-language-action model integrated with action chunking via parallel decoding. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2025)
31. Song, W., Zhou, Z., Zhao, H., Chen, J., Ding, P., Yan, H., Huang, Y., Tang, F., Wang, D., Li, H.: Reconvla: Reconstructive vision-language-action model as effective robot perceiver. arXiv preprint arXiv:2508.10333 (2025)
32. Wang, H., Xiong, C., Wang, R., Chen, X.: Bitvla: 1-bit vision-language-action models for robotics manipulation. arXiv preprint arXiv:2506.07530 (2025)
33. Wang, X., Xu, C., Jin, Y., Jin, J., Zhang, H., Yu, K., Deng, Z.: Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=t5uLZSRjhF
34. Wang, Y., Li, X., Wang, W., Zhang, J., Li, Y., Chen, Y., Wang, X., Zhang, Z.: Unified vision-language-action model. arXiv preprint arXiv:2506.19850 (2025)
35. Wen, J., Zhu, M., Liu, J., Liu, Z., Yang, Y., Zhang, L., Zhang, S., Zhu, Y., Xu, Y.: dvla: Diffusion vision-language-action model with multimodal chain-of-thought. arXiv preprint arXiv:2509.25681 (2025)
36. Wen, J., Zhu, Y., Li, J., Zhu, M., Tang, Z., Wu, K., Xu, Z., Liu, N., Cheng, R., Shen, C., Peng, Y., Feng, F., Tang, J.: Tinyvla: Toward fast, data-efficient vision-language-action models for robotic manipulation. IEEE Robotics and Automation Letters **10**(4), 3988–3995 (2025). https://doi.org/10.1109/LRA.2025.3544909
37. Wen, Y., Li, H., Gu, K., Zhao, Y., Wang, T., Sun, X.: Llada-vla: Vision language diffusion action models. arXiv preprint arXiv:2509.06932 (2025)
38. Wu, C., Zhang, H., Xue, S., Liu, Z., Diao, S., Zhu, L., Luo, P., Han, S., Xie, E.: Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. arXiv preprint arXiv:2505.22618 (2025)
39. Wu, H., Jing, Y., Cheang, C., Chen, G., Xu, J., Li, X., Liu, M., Li, H., Kong, T.: Unleashing large-scale video generative pre-training for visual robot manipulation. In: International Conference on Learning Representations (2024)

40. Xu, S., Wang, Y., Xia, C., Zhu, D., Huang, T., Xu, C.: Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation (2025)
41. Xu, Y., Yang, Y., Fan, Z., Liu, Y., Li, Y., Li, B., Zhang, Z.: Qvla: Not all channels are equal in vision-language-action model's quantization. arXiv preprint arXiv:2602.03782 (2026)
42. Yang, Y., Wang, Y., Wen, Z., Zhongwei, L., Zou, C., Zhang, Z., Wen, C., Zhang, L.: Efficientvla: Training-free acceleration and compression for vision-language-action models. arXiv preprint arXiv:2506.10100 (2025)
43. Ye, J., Gong, S., Gao, J., Fan, J., Wu, S., Bi, W., Bai, H., Shang, L., Kong, L.: Dream-vl & dream-vla: Open vision-language and vision-language-action models with diffusion language model backbone. arXiv preprint arXiv:2512.22615 (2025)
44. Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., Kong, L.: Dream 7b: Diffusion large language models. arXiv preprint arXiv:2508.15487 (2025)
45. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22963–22974 (2025)
46. Yu, R., Li, Q., Wang, X.: Discrete diffusion in large language and multimodal models: A survey. arXiv preprint arXiv:2506.13759 (2025)
47. Yue, Y., Wang, Y., Kang, B., Han, Y., Wang, S., Song, S., Feng, J., Huang, G.: Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
48. Zhang, J., Guo, Y., Hu, Y., Chen, X., Zhu, X., Chen, J.: Up-vla: A unified understanding and prediction model for embodied agent. arXiv preprint arXiv:2501.18867 (2025)
49. Zhang, R., Dong, M., Zhang, Y., Heng, L., Chi, X., Dai, G., Du, L., Du, Y., Zhang, S.: Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. arXiv preprint arXiv:2503.20384 (2025)
50. Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., Dong, R., He, J., Wang, H., Zhang, Z., et al.: Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. arXiv preprint arXiv:2507.04447 (2025)
51. Zheng, C., Vuong, T.L., Cai, J., Phung, D.: Movq: Modulating quantized vectors for high-fidelity image generation. Advances in Neural Information Processing Systems **35**, 23412–23425 (2022)