Spec-VLA: Speculative Decoding for Vision-Language-Action Models with Relaxed Acceptance

Songsheng Wang[1*†] **and Rucheng Yu**[2*] **and Zhihang Yuan**[2] **and Chao Yu**[34] **and Feng Gao**[3] **and Yu Wang**[3] **and Derek F. Wong**[1§]

[1] NLP[2]CT Lab, Department of Computer and Information Science, University of Macau
[2] Infinigence AI [3] Tsinghua University [4] Zhongguancun Academy
{nlp2ct.songsheng,ruchengyu1130,hahnyuan}@gmail.com
{yuchao,yu-wang}@mail.tsinghua.edu.cn
gaof22@mails.tsinghua.edu.cn
derekfw@um.edu.mo

Abstract

Vision-Language-Action (VLA) models have made substantial progress by leveraging the robust capabilities of Visual Language Models (VLMs). However, VLMs' significant parameter size and autoregressive (AR) decoding nature impose considerable computational demands on VLA models. While Speculative Decoding (SD) has shown efficacy in accelerating Large Language Models (LLMs) by incorporating efficient drafting and parallel verification, allowing multiple tokens to be generated in one forward pass, its application to VLA models remains unexplored. This work introduces Spec-VLA, an SD framework designed to accelerate VLA models. Due to the difficulty of the action prediction task and the greedy decoding mechanism of the VLA models, the direct application of the advanced SD framework to the VLA prediction task yields a minor speed improvement. To boost the generation speed, we propose an effective mechanism to relax acceptance utilizing the relative distances represented by the action tokens of the VLA model. Empirical results across diverse test scenarios affirm the effectiveness of the Spec-VLA framework, and further analysis substantiates the impact of our proposed strategies, which enhance the acceptance length by 44%, achieving $1.42\times$ speedup compared with the OpenVLA baseline, without compromising the success rate. The success of the Spec-VLA framework highlights the potential for broader application of speculative execution in VLA prediction scenarios. We make our code and data publicly available at https://github.com/PineTreeWss/SpecVLA.

1 Introduction

The Vision-Language-Action (VLA) models (Brohan et al., 2022, 2023; Mees et al., 2024; Wu et al.; Cheang et al., 2024; Vuong et al., 2023) have achieved significant progress by leveraging the rich understanding and generation capabilities from pre-trained visual encoders or Visual Language Models (VLMs). These models can generate robot actions following language instructions. With the development of large-scale robot prediction datasets, recently proposed VLA models such as OpenVLA (Kim et al., 2024) demonstrate high generalizability across diverse tasks and environments (Li et al., 2024b).

To achieve the goals above, the parameter size of backbone VLMs is substantial, increasing the computational demand for robot control systems. Meanwhile, the VLMs' Autoregressive (AR) next-token-prediction strategy further increases the decoding latency of VLA models. A series of studies address the efficiency issue through model architecture redesign (Wen et al., 2025; Liu et al., 2024b) or task-specific optimizations (Kim et al., 2025). Other efforts incorporate Large Language Model (LLM) inference acceleration methods such as Early-Exit (Schuster et al., 2022) and Jacobi-Decoding (Kou et al., 2024) into VLA inference (Yue et al., 2024; Song et al., 2025). However, incorporating such methods requires resource-intensive fine-tuning of the backbone VLM for Early-Exit (Yue et al., 2024) or pretraining for Jacobi-Decoding (Song et al., 2025). Moreover, in Jacobi-Decoding, enabling parallel decoding degrades the model performance compared to AR decoding (Song et al., 2025).

Speculative Decoding (SD) provides a lossless solution and also allows for the parallel generation of LLMs. A typical SD architecture, such as Eagle (Li et al., 2024d), employs a draft model to generate draft tokens efficiently, with the LLMs serving as the verification model to ensure the correctness of these tokens. As the parameters of the draft model are decoupled, additional fine-tuning of the verification model is not required.

[*] Equal contribution.
[†] Work was done during the internship at NICS-EFC Lab, Department of Electronic Engineering, Tsinghua University.
[§] Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26928–26940
November 4-9, 2025

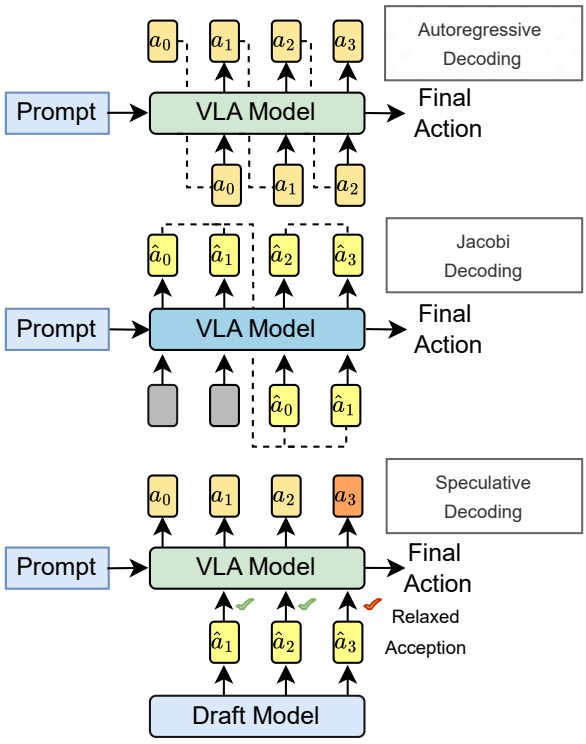

Figure 1: Comparison of Autoregressive Decoding, Jacobi-Decoding, and Spec-VLA Decoding Framework. Spec-VLA framework enables parallel generation without tuning or retraining for target VLA model.

Recent works have applied the SD framework in visual generation (Jang et al., 2024; Park et al., 2025), designing a task-specific methodology to relax the acceptance for the verification model. The application of SD architecture to accelerate the VLA prediction task is intuitive, enabling efficient adaptation to speed up the generation of downstream tasks while retaining the knowledge of the VLA backbone model. However, its application to the VLA model has not yet been explored.

This work introduces the speculative decoding framework to the AR robot action generation. We propose the Spec-VLA, the first SD framework designed for VLA inference acceleration, which applies the advanced features of the speculative decoding to the robot action generation scenarios. Surprisingly, the direct application of the SD framework yields minor speed improvements due to the intricate difficulty of VLA prediction for the draft model and the greedy decoding strategy. To further boost the generation speed, we propose utilizing VLA models' token representation to relax the acceptance based on the action distance between draft tokens and ground-truth tokens. Empirical results across various test scenarios demonstrate the effectiveness of the Spec-VLA framework, enabling an acceptance length from 2.10 to 2.94. Analysis confirms that our proposed relaxation of acceptance strategy significantly enhances the acceptance length by 26% to 44%, enhances the generation speed by $1.22\times$ to $1.42\times$ while maintaining the success rate of the VLA models.

Existing reinforcement learning studies have highlighted the importance of robustness, either in multi-task settings where policy degradation occurs (Bai et al., 2023; Liu et al., 2025), or under adversarial perturbations where agents exhibit vulnerability (Bai et al., 2025). Inspired by these studies, we further analyze the robustness of VLA models under speculative decoding with relaxed acceptance by exploring the relaxation threshold under multiple tasks, demonstrating the potential of SD frameworks in the VLA prediction domain.

2 Related Works

2.1 Acceleration for VLA Models

Recent advances in accelerating VLA models can be broadly categorized into multiple directions. Token-level optimization methods reduce computational redundancy through vision-language token selection. The FastV (Pertsch et al., 2025) distills task-relevant visual features using auxiliary transformers, while SparseVLM (Zhang et al., 2024c) dynamically prunes tokens via spatial attention thresholds. Though efficient without architectural changes, these approaches rely heavily on heuristic token selection, risking generalization failures in novel scenarios.

Conventional LLM acceleration techniques like quantization, pruning, and early-exit strategies have also been adapted for VLA scenarios. QAIL (Park et al., 2024) employs quantization-aware fine-tuning but suffers from precision loss. Mope-CLIP (Lin et al., 2024) explores modality-specific pruning for vision-language models, and DeeR (Yue et al., 2024) implements early-exit mechanisms that compromise action trajectory coherence. While effective in constrained settings, such methods often degrade cross-modal interaction quality and require task-specific tuning.

Structural modifications, such as Robomamba (Liu et al., 2024b) and TinyVLA (Wen et al., 2025), redesign model backbones using lightweight SSM or distilled vision encoders, achieving latency reduction through structural simplification. The Kim et al. (2025) propose

temporal consistency losses to regularize action smoothness, and Song et al. (2025) reformulate decoding via Jacobi iteration for parallel trajectory generation. The aforementioned methodologies not only require domain-specific data fine-tuning or retraining but also introduce augmented system complexity through model architectural redesign. Beyond architectural redesign and decoding strategies, recent RL-based studies also target efficiency in VLA inference. SEER (Bai et al., 2024) improves sample efficiency via aligned experience estimation and policy regularization, while D3P (Yu et al., 2025) accelerates inference by adaptively adjusting diffusion steps. Together, they highlight complementary RL-driven strategies for efficient VLA inference.

2.2 Speculative Decoding for LLMs

The SD has emerged as an effective paradigm for inference acceleration in AR generative models, such as machine translation models (Stern et al., 2018) and decoder-only LLMs (Chen et al., 2023). The evolutionary trajectory of SD frameworks reveals three distinct development phases. Pioneering SD frameworks exemplified by Medusa (Cai et al., 2024) and Medusa-CTC (Wen et al., 2024) introduced parallel generation capabilities through multi-head decoding architectures coupled with tree-attention verification mechanisms. Subsequent developments in the Eagle series, including Eagle (Zhang et al., 2024a) and Eagle-2 (Li et al., 2024c), advanced the paradigm through architectural innovations in draft modeling, achieving superior speedup ratios via high-quality draft token generation. Recently, the Eagle-3 (Li et al., 2025) and HASS (Zhang et al., 2024b) have further improved the generation capabilities by employing a training-time testing strategy. The framework have shown remakable superiority for LLM acclleration ($3.2\times$ - $5.6\times$), compared with Jacobi-Decoding ($2.5\times$ - $3.0\times$) (Kou et al., 2024) and Early-Exit Decoding ($1.9\times$ - $1.8\times$) (Liu et al., 2024a).

Recent works have further extended SD applications to emerging scenarios, including retrieval-argumented generation (Wang et al., 2024) and long-context generation (Yang et al., 2025). However, empirical validation remains insufficient for multimodal generation contexts. Initial investigations by Jang et al. (2024) demonstrated significant performance degradation when applying existing SD frameworks to visual AR generation tasks. Gagrani et al. (2024) conducted systematic analyses of visual feature utilization in multimodal applications such as visual question answering and image captioning. Despite these advances, the application of SD methodologies within the VLA generation scenario remains unexplored.

Relaxed acceptance proves effective in the SD framework, demonstrating particular promise for extending efficiency gains to novel application scenarios. It boosts throughput by loosening the criteria for accepting proposed tokens, striking a balance between efficiency and fidelity. SpecDec (Xia et al., 2022) replaces the strict greedy check by accepting any drafted token appearing in the AR model's top-k candidates, significantly raising token acceptance rates and overall throughput without degrading output quality. Meanwhile, the Lantern framework (Jang et al., 2024) accepts the top-k similar tokens in the dictionary, which significantly boosts the generation speed for visual generation. (Further improvements (Li et al., 2024a; Zhang et al., 2023)). These advancements proves the potential of relaxed acceptance in enhancing the efficiency of multimodal models, such as VLA models.

3 Background

3.1 Decoding of VLA Models

Large VLA models (Ma et al., 2024), such as OpenVLA (Kim et al., 2024) and RT-2 (Brohan et al., 2023) series, predict action sequences to control robots. They employ a sequence of action tokens $A = \{a_0, ..., a_L\}$ which represent the actions at each timestep. Using VLM inference, the model autoregressively predicts seven action tokens to define a control action, including "Δpos_x", "Δpos_y" ,"Δpos_z", "Δrot_x" , "Δrot_y", "Δrot_z" and "gripper_extension". Specifically, they utilize greedy decoding, predicting the most probable action token a_i based on the previously predicted tokens $a_{0:i-1}$, visual observations o, language instruction prompts p, and the learnable model parameters θ.

$$a_i = \underset{a_i}{\text{argmax}} \left[P(a_i \mid a_{0:i-1},\, o,\, p,\, \theta) \right] \quad (1)$$

Due to the substantial parameter size of contemporary VLA models and their AR prediction strategy, the action speed of robot is inherently limited.

3.2 Speculative Decoding Framework

The SD framework utilizes an efficient **draft model** M_D to produce initial draft tokens and

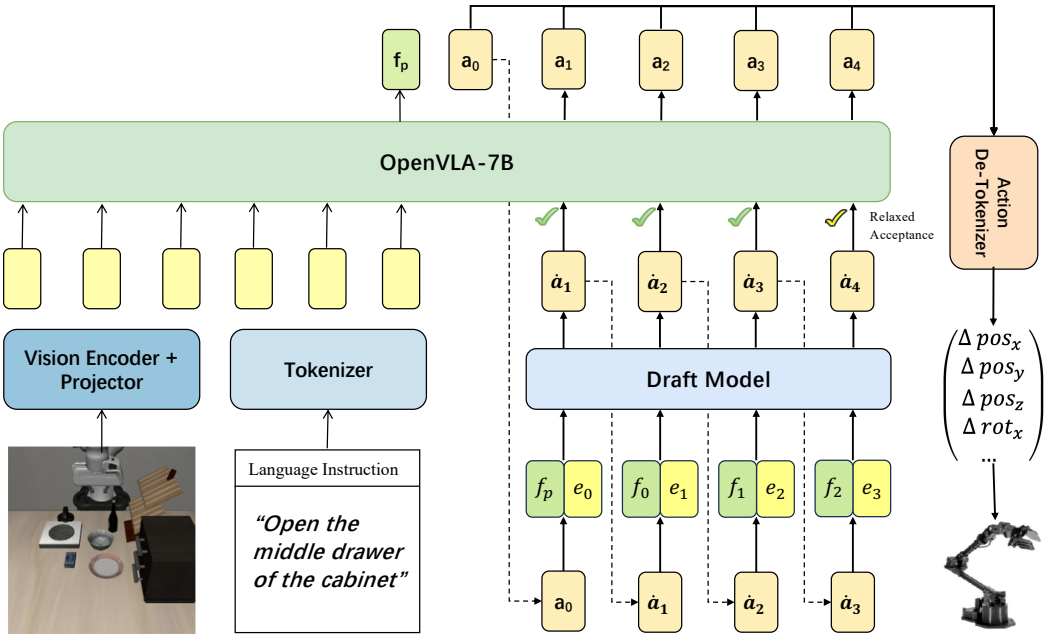

Figure 2: The overall Spec-VLA framework. The draft model predicts action tokens through AR decoding with the fused textual and visual features. During verification, a relaxed acceptance mechanism is adopted to broadly retain high-quality outputs. This mechanism allows synonym to be accepted, while maintaining the success rate of action generation, achieving optimal balance between caption accuracy and efficiency.

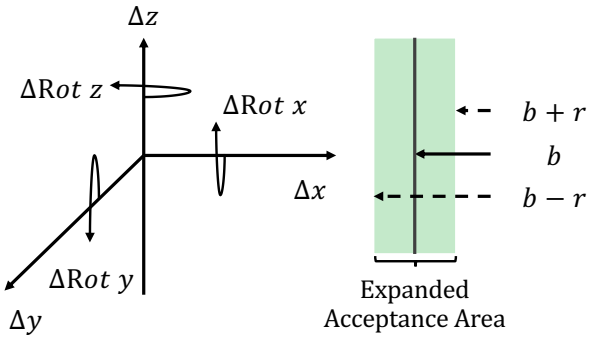

Figure 3: Illustration of relaxation of the acceptance criteria. Instead of strictly accepting the predicted verify token a_i, the verification model M_V accept action tokens within a predefined margin.

concurrently verifies these tokens using a **verification model** M_V. The Eagle framework (Li et al., 2024c) incorporates a Llama layer as the draft model, which predicts multiple draft tokens a_i autoregressively, conditioned on the previous draft token states $\hat{a}_{t+1:i-1}$, hidden states and token embeddings from verification model $f_{1:t}$ and $e_{0:t}$. It is noteworthy that the output states of the draft tokens also assist in calculations, and for the sake of simplicity, we use the notation $\hat{a}_{t+1:i-1}$ to denote both embeddings and hidden features.

$$\hat{a}_i = M_D(f_{1:t}, e_{0:t}, \hat{a}_{t+1:i-1}) \tag{2}$$

During the verification phase, the verification model M_V ensures the generation quality of the draft tokens by correcting the mispredicted tokens from the draft model. When conducting a greedy search, the draft token $\dot{a}_i$ will be accepted only if it strictly matches the token a_i predicted by the verification model.

$$a_i = M_V\big(a_1 \sim \hat{a}_{i-1},\, p,\, \theta\big),$$
$$\begin{cases} \text{Accept}, & a_i == \hat{a}_i, \\ \text{Resample } \hat{a}_i = a_i, & a_i \neq \hat{a}_i. \end{cases} \tag{3}$$

Noticeably, the tokens subsequent to the first rejected token $a_{(i+1:L)}$ will be abandoned. Thus, the acceptance length is critical for the SD system as it determines the number of tokens to be predicted in a single forward pass.

4 Spec-VLA Framework

In this section, we provide a detailed description of the Spec-VLA framework and our exploration of the adaptation of speculative execution for VLA prediction tasks.

4.1 Overall Framework

The Spec-VLA framework incorporates a Llama decoder layer (Touvron et al., 2023) as its draft generator model. It incorporates a linear layer to

Dataset	AR	Spec-VLA			Spec-VLA (relaxed)		
	SR	Length	Speedup	SR	Length	Speedup	SR
LIBERO-Goal	**78.0%**	2.04	1.09×	74.2%	**2.94**	**1.42×**	74.4%
LIBERO-Object	**89.0%**	1.75	1.15×	**89.0%**	**2.38**	**1.38×**	85.0%
LIBERO-Spatial	85.0%	1.59	1.08×	83.8%	**2.14**	**1.28×**	**85.8%**
LIBERO-Long	52.0%	1.67	1.13×	50.8%	**2.10**	**1.22×**	**55.0%**

Table 1: Experimental results of the Spec-VLA framework on the LIBERO-Goal, Object, Spatial, Long dataset. 'SR' denotes the Success Rate of the control policy, 'Length' indicates the number of tokens predicted in each forward pass, and 'Speedup' reflects the generation speed as compared to the AR baseline.

integrate feature-level and token-level loss data effectively. During the prefill stage, the draft generator receives hidden states from the verification model, alongside textual and visual embeddings from the textual tokenizer and visual encoder, respectively. Mirroring the OpenVLA model, the visual embeddings e_v and textual embeddings e_T are concatenated, collectively providing the feature-level information for the draft model.

$$\hat{a}_i = M_D(f_{1:t}, \text{concat}(e_v, e_p), \hat{a}_{t+1:i-1})) \quad (4)$$

In the draft prediction phase, the draft generator model predicts the action token a_i conditioned on previous hidden states, embeddings, and action tokens. We employ the dynamic draft tree strategy of Eagle-2 (Li et al., 2024c), where the Top-K predictions from the draft generator M_D are recorded and subsequently form a tree structure with multiple paths. These paths are then verified in parallel by the verification model.

4.2 Problem by Direct Application

However, directly implementing the SD framework yields only minor speed improvements, from $1.08\times$ to $1.15\times$ (as shown in Table 1). Surprisingly, in the VLA prediction task, the draft generator models fail to predict the initial draft tokens in about half of the samples (refer to Table 2).

In natural language generation tasks, the draft generator of the SD framework typically produces common words and punctuation. Conversely, the VLA draft model must understand multiple modalities and predict robotic motions in VLA prediction tasks. Intuitively, the VLA prediction task poses a greater complexity for the draft generator than language generation.

Moreover, VLA models such as OpenVLA and RT-2 incorporate greedy decoding during the drafting phase. This setup requires an exact match between draft tokens $\hat{a}_i$ and the verification model's predictions a_i. Often, allowing for synonym tokens could improve generation speed without compromising quality. Building upon prior research, we propose relaxing the acceptance criteria within the Spec-VLA framework by allowing the acceptance of top-k similar tokens in the action space.

4.3 Relaxation of Acceptance

We introduce the Relaxation Threshold r to facilitate acceptance relaxation, quantifying the permissible distance between the draft action token $\hat{a}_i$ and the predicted action token a_i. The draft token $\hat{a}_i$ will be accepted if the distance D between $\hat{a}_i$ and a_i is not larger than threshold r.

$$\begin{aligned} a_i &= M_V\big(a_1 \sim \hat{a}_{i-1},\, p,\, \theta\big), \\ &\begin{cases} \text{Accept}, & D(a_i, \hat{a}_i) \le r \\ \text{Resample } \hat{a}_i = a_i, & D(a_i, \hat{a}_i) > r. \end{cases} \end{aligned} \quad (5)$$

VLA models, notably OpenVLA and RT-2, discretize continuous dimensions into 256 bins and map them to 256 action tokens to predict action sequences. The VLA token representation inherently provides information on token similarity, where the distance between tokens can be directly inferred from the absolute difference between bin IDs. For instance, the token a represents bin b and the token $\hat{a}$ represents bin $\hat{b}$, the distance between a and $\hat{a}$ can be directly determined by the absolute difference between bin IDs b and $\hat{b}$. The token acceptance area will be widened from strictly b to $\hat{b} \in (b-r, b+r)$, enabling the acceptance of the top $2 \times r$ similar tokens.

By utilizing this characteristic, our proposed method eliminates the need for additional token similarity calculations from token embeddings, introducing virtually no computational overhead.

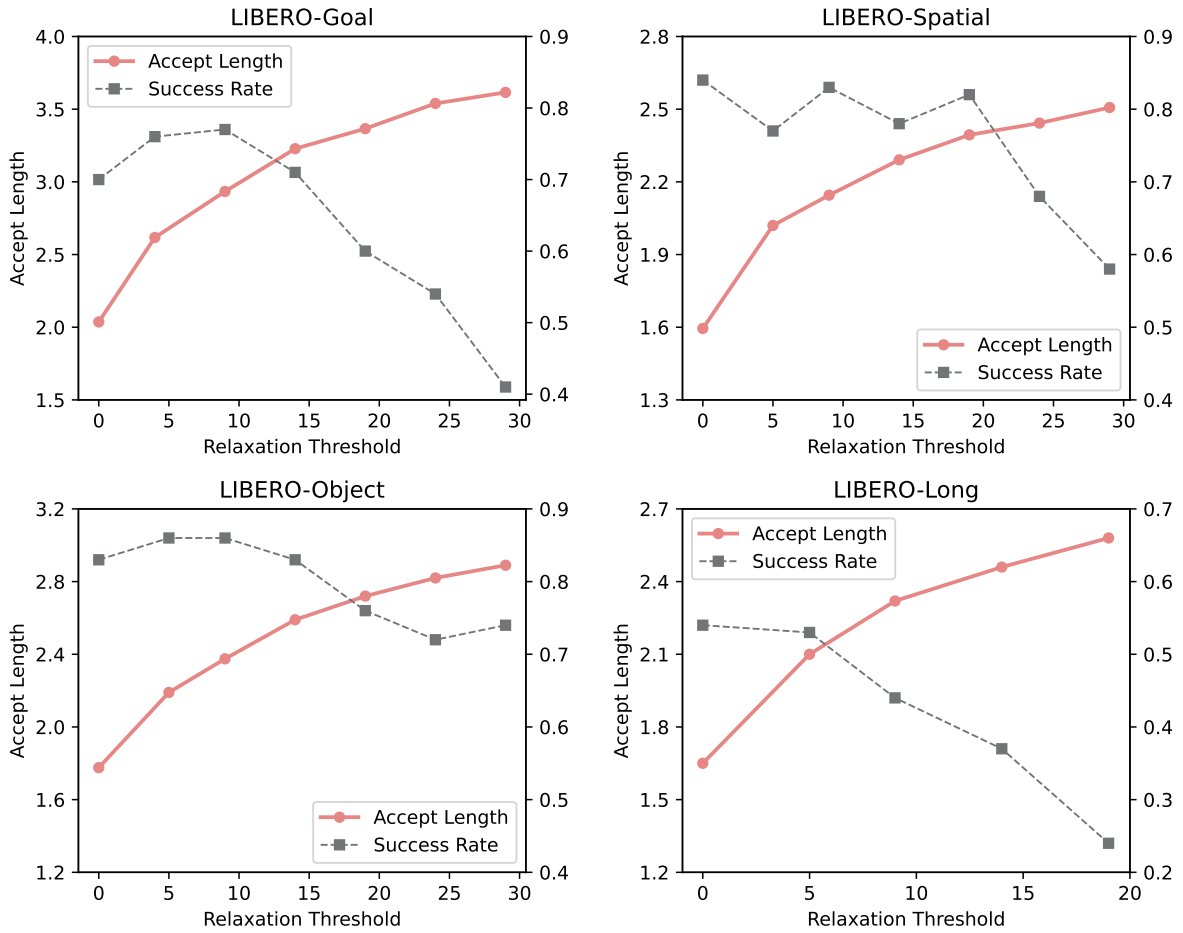

Figure 4: Acceptance Length and Success Rate of the Spec-VLA framework on the LIBERO-Goal, LIBERO-Spatial, LIBERO-Object, and LIBERO-Long datasets. An increase in the Relaxation Threshold shows a minor impact on the Success Rate while significantly boosting the Acceptance Length.

5 Experiment

5.1 Main Result

Following OpenVLA, we evaluated the Spec-VLA framework on the LIBERO simulation benchmark (Liu et al., 2023). We utilized four task suites: LIBERO-Object, LIBERO-Spatial, LIBERO-Goal, and LIBERO-Long, each providing 10 tasks and 500 expert demonstrations. We employed the fine-tuned OpenVLA as the verification model and used this model to regenerate the dataset for training the draft model. We conducted 50 trials on each task with our SD frameworks for testing scenarios. The training was completed in 6 hours using $4\times$ Tesla A100 (80G) GPU, with a batch size of 16. We inherent the implementation of draft model structure and tree decoding mechenism from Eagle-2 (Li et al., 2024c). For tree-decoding, we set the maximum nodes to 50, tree depth to 4, and used the top 8 tokens to construct the draft tree. Drawing on prior works in SD (Jang et al., 2024), we report the number of tokens predicted in each forward pass and speedup compared to AR decoding.

The main results are reported in Table 1. Firstly, the results validate the effectiveness of the SD framework in VLA prediction scenarios. Applying the Eagle framework achieves an acceleration ratio ranging from $1.08\times$ to $1.15\times$ without sacrificing generation quality. Secondly, the relaxed acceptance mechanism further enhances the generation speed of the SD framework, increasing the acceptance length by 25% to 44%, demonstrating the potential for developing specialized SD mechanisms in the VLA scenario.

5.2 Ablations on Relaxation Threshold

This section further analyzes the relationship between the relaxation threshold, success rate, and acceptance length (as shown in Figure 4). We conducted analyses on the LIBERO-Goal, LIBERO-Spatial, LIBERO-Object, and LIBERO-Long benchmarks, each containing 10 tasks, with

Dataset	Relaxed	Acceptance Length					
		0	1	2	3	4	5
Libero-Goal	×	50.24%	33.28%	13.96%	2.23%	0.19%	0.00%
	✓	23.01%	18.98%	39.99%	15.41%	2.53%	0.08%
Libero-Object	×	47.93%	34.72%	12.72%	4.07%	0.56%	0.00%
	✓	28.23%	37.62%	17.29%	10.11%	6.22%	0.53%
Libero-Spatial	×	55.96%	31.52%	9.90%	2.46%	0.16%	0.00%
	✓	37.07%	33.52%	19.15%	8.23%	1.99%	0.03%
Libero-Long	×	55.08%	28.77%	11.30%	4.30%	0.50%	0.05%
	✓	42.39%	28.57%	16.65%	8.84%	3.35%	0.20%

Table 2: Acceptance length distribution on the LIBERO-Goal, LIBERO-Object, LIBERO-Spatial, and LIBERO-Long datasets under non-relaxed and relaxed settings. Each row reports the proportion of trials that succeeded with a specific acceptance length. The threshold for relaxation is 9 for LIBERO-Goal, LIBERO-Object, and LIBERO-Spatial, and 5 for LIBERO-Long.

10 trials performed for each task. We tested starting from a relaxation threshold of 0, which corresponds to strict matching acceptance.

First, Relaxation of acceptance criteria effectively enhances the acceptance length, boosting the generation speed of the VLA models. The increase in relaxation distance can enhance the acceptance length by 50% to 70% across various datasets. Moreover, we surprisingly found that the OpenVLA model displays high robustness on the LIBERO-Goal, LIBERO-Object, and LIBERO-Spatial datasets. The relaxation threshold could be relaxed from 5 to 9 without sacrificing the success rate of the VLA model.

Additionally, the better a model performs in a scenario, the larger the relaxation threshold it can tolerate. In the LIBERO-Long dataset, the success rate drops significantly when the relaxation threshold exceeds 5. However, in LIBERO-Goal, the success rate remains stable even with the relaxation threshold set to 15. This analysis verifies the effectiveness of our proposed relaxed acceptance strategy and also highlights the high potential for speculative execution within the VLA framework.

6 Analysis

This section provides an analysis of the Spec-VLA framework under non-relaxed and relaxed acceptance conditions, focusing on acceptance length distribution patterns and prediction performance across distinct action tokens on four benchmark datasets (Libero-Goal, Libero-Object, Libero-Spatial, and Libero-Long). Consider that the verification model invariably emits an accept length of 1, which carries no discriminative information; our analysis here considers only the accept lengths produced by the draft model. Once verification outputs are excluded, the minimum accept length becomes 0 (indicating no speculative tokens were accepted), so the average accept length on each position can legitimately fall below 1.

6.1 Acceptance Length Proportion

Table 2 quantifies the distribution of acceptance lengths (0–5) under the Spec-VLA framework, comparing non-relaxed and relaxed conditions across four datasets. The data reveals a distinct trend: non-relaxed acceptance disproportionately favors shorter sequences (lengths 0–1), with proportions sharply declining for longer lengths (2–5), whereas relaxed acceptance exhibits a more balanced distribution. The dominance of short sequences under non-relaxed conditions (e.g., 50.24% at length 0) highlights a critical inefficiency: models prioritize 'safe' short predictions to avoid constraint violations. This artificially low conversion rate for longer sequences implies that strict constraints act as a bottleneck preventing the model from predicting longer action sequence. The most pronounced contrast occurs in Libero-Object at length 4: non-relaxed acceptance plummets to 0.56% versus 6.22% under relaxed conditions—an 11-fold relative increase. Similarly, Libero-Long exhibits dramatic divergence at length 4 (0.50% vs. 3.35%, 6.7× improvement) and Libero-Spatial at length 3 (2.46% vs. 8.23%, 3.3× improvement).

Dataset	Relaxed	Position					
		0	1	2	3	4	5
Libero-Goal	×	0.47	0.30	0.73	0.78	1.13	0.98
	✓	1.44	1.36	2.18	2.13	1.76	0.98
Libero-Object	×	0.60	0.64	1.02	0.67	0.88	0.99
	✓	1.39	2.09	1.66	1.46	1.40	0.99
Libero-Spatial	×	0.36	0.50	0.88	0.72	0.70	0.96
	✓	0.89	1.39	1.55	1.56	1.20	0.98
Libero-Long	×	0.79	0.42	1.19	0.87	0.65	0.64
	✓	1.27	0.94	1.70	1.35	1.12	0.72

Table 3: Average acceptance lengths at each position (0–5) on the LIBERO-Goal, LIBERO-Object, LIBERO-Spatial, and LIBERO-Long datasets under non-relaxed and relaxed conditions. Each entry reports the average acceptance length observed at the given token position. The relaxed setting is consistent with Table 2.

Even at maximum length 5, relaxed acceptance achieves non-zero proportions (e.g., 0.53% for relaxed in Libero-Object vs. 0% non-relaxed). These disparities highlight a critical limitation of strict constraints: they disproportionately penalize longer sequences. Relaxation alleviates this by allowing semantically compatible draft tokens to be accepted, thereby increasing sequence diversity without compromising task success rates.

6.2 Acceptance Length on Multiple Positions

We perform further analysis to evaluate the acceptance length in each starting position. As shown in Table 3, relaxed acceptance consistently achieves longer average lengths than non-relaxed acceptance across all positions. For Libero-Object, acceptance length at position 1 surges from 0.64 (non-relaxed) to 2.09 under relaxation (3.3× improvement), reflecting reduced bias toward short-term predictions. Similarly, Libero-Goal shows a 3.1× increase at position 0 (0.47 $\rightarrow$ 1.44), highlighting the model's willingness to explore initial reasoning steps when constraints are loosened. Libero-Spatial also exhibits a 2.2× gain at position 3 (0.72 $\rightarrow$ 1.56), revealing that relaxation mitigates premature truncation of valid action sequences, whereas relaxation balances risk and exploration to unlock the potential for longer action sequence generation. These results align with findings in Table 2.

6.3 Case Study

This section provides a representative case to show the effectiveness of our proposed relaxation of acceptance method. As shown in Figure 5, under the strict verification model (Non-Relaxed), the series appends only those candidate tokens that satisfy a stringent acceptability threshold, resulting in a gradual accretion of the action sequence. For instance, Action 1 extends from a solitary context token [137] to the fully verified sequence [137, 128, 128, 109, 98, 82, 256] over four iterative refinement steps. In contrast, by relaxed acceptance, the relaxed criterion admits a broader spectrum of draft proposals at an earlier stage; Action 1 already incorporates the tokens [119, 121, 109] in its initial iteration and further augments this set with [98, 77, 256] in the second iteration. The same pattern holds for the other cases. Action 3, for example, reaches the whole sequence [191, 121, 123, 109, 79, 69, 256] in only three iterations under the relaxed acceptance, whereas the non-relaxed acceptance requires five iterations. These results show that relaxing the acceptance threshold significantly reduces the number of iterations needed for plan generation while still preserving the quality of the final action sequences. This relaxation also accelerates the action sequence completion process, reducing the number of iterations without compromising functional validity.

7 Conclusion

In this study, we explore the application of the SD framework in VLA prediction tasks. We propose Spec-VLA, which enhances the Eagle framework for VLA predictions. To further boost the generation speed of the framework, we introduce the distance-sensitive relaxation of the acceptance strategy, which utilizes the token representation of VLA

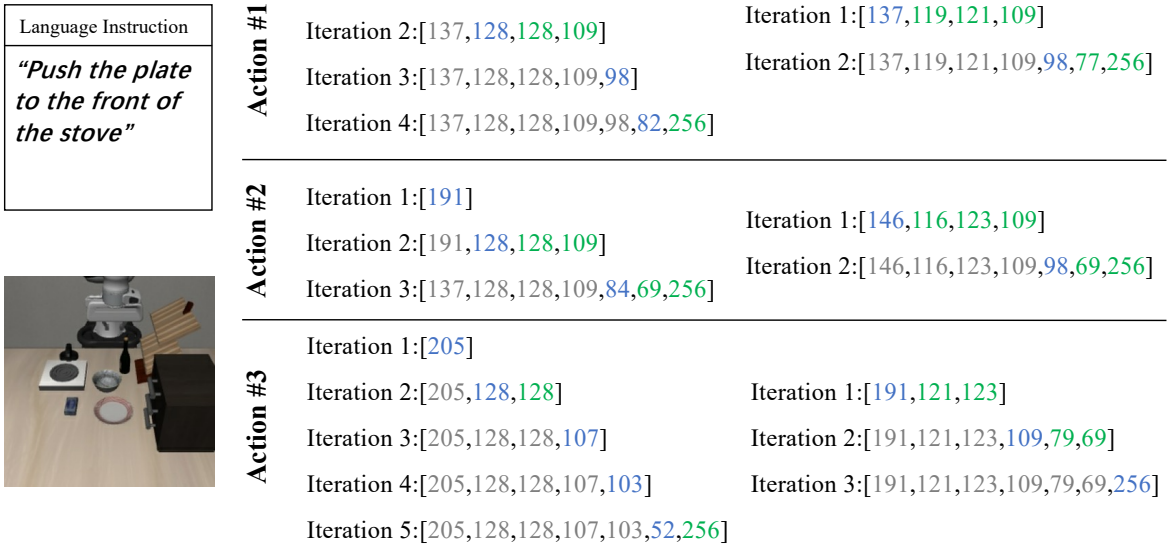

Figure 5: Illustration of action sequence generation cases under non-relaxed and relaxed acceptance conditions in the Spec-VLA framework. Three representative action trajectories are juxtaposed for systematic comparison across both conditions. Gray denotes context tokens. Blue represents verification model outputs. Green indicates draft model outputs.

models to effectively identify the distance between action tokens and relax the acceptance threshold within the SD framework. Experimental results verify the effectiveness of the Spec-VLA framework, where the relaxation of acceptance criteria further boosts the acceptance length by 25% to 44% without compromising the success rate. Our findings on the relaxation of acceptance show high robustness of the VLA models, demonstrating the potential of speculative systems in the VLA prediction domain.

Limitations

This work explores speculative decoding in VLA prediction tasks. Due to time and resource constraints, experiments were not conducted in real-world robotic settings. Additionally, due to limitations of the verification model, Action Chunking was not explored. Future work could incorporate additional methodologies into the SD framework for VLA models.

Acknowledgements

This work was supported in part by the Science and Technology Development Fund of Macau SAR (Grant No. FDCT/0007/2024/AKP), the Science and Technology Development Fund of Macau SAR (Grant No. FDCT/0070/2022/AMJ, China Strategic Scientific and Technological Innovation Cooperation Project Grant No. 2022YFE0204900), the Science and Technology Development Fund of Macau SAR (Grant No. FDCT/060/2022/AFJ, National Natural Science Foundation of China Grant No. 62261160648), the UM and UMDF (Grant Nos. MYRG-GRG2023-00006-FST-UMDF, MYRG-GRG2024-00165-FST-UMDF, EF2024-00185-FST), and the National Natural Science Foundation of China (Grant No. 62266013).

References

Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. 2025. Rat: Adversarial attacks on deep reinforcement agents for targeted behaviors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(15):15453–15461.

Fengshuo Bai, Hongming Zhang, Tianyang Tao, Zhiheng Wu, Yanna Wang, and Bo Xu. 2023. Picor: Multi-task deep reinforcement learning with policy correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6728–6736.

Fengshuo Bai, Rui Zhao, Hongming Zhang, Sijia Cui, Ying Wen, Yaodong Yang, Bo Xu, and Lei Han. 2024. Efficient preference-based reinforcement learning via aligned experience estimation. *arXiv preprint arXiv:2405.18688*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, and 1 others. 2023. Rt-2: Vision-language-action

models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, and 1 others. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, and 1 others. 2024. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Mukul Gagrani, Raghavv Goel, Wonseok Jeon, Junyoung Park, Mingu Lee, and Christopher Lott. 2024. On speculative decoding for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8289.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. 2024. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*.

Moo Jin Kim, Chelsea Finn, and Percy Liang. 2025. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. 2024. Cllms: Consistency large language models. In *Forty-first International Conference on Machine Learning*.

M. Li, X. Chen, A. Holtzman, and et al. 2024a. Nearest neighbor speculative decoding for llm generation and attribution. 37:80987–81015. NeurIPS 2024.

Shunlei Li, Jin Wang, Rui Dai, Wanyu Ma, Wing Yin Ng, Yingbai Hu, and Zheng Li. 2024b. Robonurse-vla: Robotic scrub nurse system based on vision-language-action model. *arXiv preprint arXiv:2409.19590*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024c. Eagle-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024d. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*.

Haokun Lin, Haoli Bai, Zhili Liu, Lu Hou, Muyi Sun, Linqi Song, Ying Wei, and Zhenan Sun. 2024. Mopeclip: Structured pruning for efficient vision-language models with module-wise pruning error metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27370–27380.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791.

Fangcheng Liu, Yehui Tang, Zhenhua Liu, Yunsheng Ni, Kai Han, and Yunhe Wang. 2024a. Kangaroo: Lossless self-speculative decoding via double early exiting. *arXiv preprint arXiv:2404.18911*.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. 2024b. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *Advances in Neural Information Processing Systems*, 37:40085–40110.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. 2025. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. 2024. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*.

Oier Mees, Dibya Ghosh, Karl Pertsch, Kevin Black, Homer Rich Walke, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, and 1 others. 2024. Octo: An open-source generalist robot policy. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

Seongmin Park, Hyungmin Kim, Wonseok Jeon, Juyoung Yang, Byeongwook Jeon, Yoonseon Oh, and Jungwook Choi. 2024. Quantization-aware imitation-learning for resource-efficient robotic control. *arXiv preprint arXiv:2412.01034*.

Sihwan Park, Doohyuk Jang, Sungyub Kim, Souvik Kundu, and Eunho Yang. 2025. Lantern++: Enhanced relaxed speculative decoding with static tree drafting for visual auto-regressive models. *arXiv preprint arXiv:2502.06352*.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. 2025. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472.

Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Jun Ma, and Haoang Li. 2025. Accelerating vision-language-action model integrated with action chunking via parallel decoding. *arXiv preprint arXiv:2503.02310*.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, and 1 others. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*.

Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, and 1 others. 2024. Speculative rag: Enhancing retrieval augmented generation through drafting. *arXiv preprint arXiv:2407.08223*.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, and 1 others. 2025. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*.

Zhuofan Wen, Shangtong Gui, and Yang Feng. 2024. Speculative decoding with ctc-based draft model for llm inference acceleration. *Advances in Neural Information Processing Systems*, 37:92082–92100.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *The Twelfth International Conference on Learning Representations*.

H. Xia, T. Ge, P. Wang, and et al. 2022. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. *Computing Research Repository*, arXiv:2203.16487.

Penghui Yang, Cunxiao Du, Fengzhuo Zhang, Haonan Wang, Tianyu Pang, Chao Du, and Bo An. 2025. Longspec: Long-context speculative decoding with efficient drafting and verification. *arXiv preprint arXiv:2502.17421*.

Shu-Ang Yu, Feng Gao, Yi Wu, Chao Yu, and Yu Wang. 2025. D3p: Dynamic denoising diffusion policy via reinforcement learning. *arXiv preprint arXiv:2508.06804*.

Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. 2024. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. *Advances in Neural Information Processing Systems*, 37:56619–56643.

Jiacheng Zhang, Yang Jiao, Shaoxiang Chen, Jingjing Chen, and Yu-Gang Jiang. 2024a. Eagle: Towards efficient arbitrary referring visual prompts comprehension for multimodal large language models. *arXiv preprint arXiv:2409.16723*.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2023. Draft & verify: Lossless large language model acceleration via self-speculative decoding. *arXiv preprint arXiv:2309.08168*.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. 2024b. Learning harmonized representations for speculative sampling. *arXiv preprint arXiv:2408.15766*.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024c. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Parameter	Value
Learning Rate	5e-5
Batch Size	16
Warmup Steps	2000
p_w	0.1
v_w	1.0
Gradiant Clipping	0.5
Top-k	8
Tree Depth	5
Max Nodes	50

Table 4: Parameter Settings of Spec-VLA Framework.

A Parameter Settings

This section details the parameter settings of the Spec-VLA model. Table 4 presents the training and inference parameters of the Spec-VLA framework. For LIBERO-Goal, LIBERO-Spatial, and LIBERO-Object, the relaxation threshold is set at 9, while for LIBERO-Long, it is set at 5. The parameters p_w and v_w represent the weights of the Cross-Entropy loss and Regression loss, respectively, as implemented in the Eagle configuration (Li et al., 2024d).

B Spec-VLA Decoding Algorithm

To enhance the understanding of the decoding process within the Spec-VLA framework, we provide pseudocode that illustrates Spec-VLA decoding with relaxed acceptance, as outlined in Algorithm 1.

C Accelleration for Quantilized Models

To illustrate SD's complementary potential, we explored further by combining Speculative Decoding with quantization.

We compared inference speeds for the OpenVLA model using int8, int4 quantization, and BF16 representations on the Tesla A100 (80G) GPU. We observed that the int8 and int4 quantization lead to decreased inference speed (Table 5). This result may be attributed to the additional overhead incurred by the quantization operation, consistent with the analysis of OpenVLA (Kim et al., 2024). Additionally, we accelerated the quantized model using the Spec-VLA framework, discovering that SD could speed up the quantized verification model. It achieves a significant speedup compared with AR decoding (Table 6).

Dataset	OpenVLA	
	Precision	Speedup
LIBERO-Goal	bf16	1.00×
	int8	0.24×
	int4	0.61×
LIBERO-Object	bf16	1×
	int8	0.21×
	int4	0.59×
LIBERO-Spatial	bf16	1×
	int8	0.23×
	int4	0.55×
LIBERO-Long	bf16	1×
	int8	0.23×
	int4	0.57×

Table 5: Speedup of the quantilized OpenVLA model on the LIBERO-Goal, Object, Spatial, Long dataset. The 'Precision' shows the quantization precision, and 'Speedup' reflects the generation speed compared to the bf16 baseline.

Dataset	SpecVLA	
	Precision	Speedup
LIBERO-Goal	bf16	1.42×
	int8	1.61×
	int4	1.34×
LIBERO-Object	bf16	1.38×
	int8	1.41×
	int4	1.33×
LIBERO-Spatial	bf16	1.28×
	int8	1.31×
	int4	1.29×
LIBERO-Long	bf16	1.22×
	int8	1.32×
	int4	1.15×

Table 6: Speedup of the quantilized SpecVLA framework on the LIBERO-Goal, Object, Spatial, Long dataset. The 'Precision' shows the quantilization precision, 'Speedup' reflects the generation speed as compared to the OpenVLA AR baseline.

Algorithm 1 Spec-VLA Decoding

Input: Prompt p, Observation o, Verification Model M_V, Draft Model M_D, Verification model hidden states $f_{1:t}$, Visual and textual embeddings $e_{0:t}$, Search Depth d, Target Length L, Relaxation Threshold r
init $n \leftarrow t$
while $n < L$ **do**
    for i in {1,...,d} **do**
        Sample draft in AR manner $\hat{a}_i = M_D(f_{1:t}, e_{0:t}, \hat{a}_{t+1:i-1})$
    end for
    Compute the reference token set $a_{t+1:t+1+d}$ in parallel: $a_i = M_V(\hat{a}_{t+1:i-1}, o, p)$
    for i in {t+1,...,t+1+d} **do**
        if D(a_i,$\hat{a}_i$)<=r **then**
            Set $a_i \leftarrow \hat{a}_i$
        else
            $a_i \leftarrow a_i$
            break
        end if
    end for
    if all drafts accepted, sample an extra token $a_{t+d+2} = M_V(a_{t+1:t+d+1}, o, p)$
end while
return $a_{t+1:t+d+1}$ or $a_{t+1:t+d+2}$