

BitVLA: 1-bit Vision-Language-Action Models for Robotics Manipulation

Hongyu Wang Chuyan Xiong Ruiping Wang Xilin Chen

Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences

Abstract

Vision-Language-Action (VLA) models have shown impressive capabilities across a wide range of robotics manipulation tasks. However, their growing model size poses significant challenges for deployment on resource-constrained robotic systems. While 1-bit pretraining has proven effective for enhancing the inference efficiency of large language models with minimal performance loss, its application to VLA models remains underexplored. In this work, we present **BitVLA**, the first 1-bit VLA model for robotics manipulation, in which every parameter is ternary, i.e., $\{-1, 0, 1\}$. To further reduce the memory footprint of the vision encoder, we propose the distillation-aware training strategy that compresses the full-precision encoder to 1.58-bit weights. During this process, a full-precision encoder serves as a teacher model to better align latent representations. Despite the lack of large-scale robotics pretraining, BitVLA achieves performance comparable to the state-of-the-art model OpenVLA-OFT with 4-bit post-training quantization on the LIBERO benchmark, while consuming only 29.8% of the memory. These results highlight BitVLA's promise for deployment on memory-constrained edge devices. We release the code and model weights in <https://github.com/ustcwhy/BitVLA>.

1 Introduction

Recent years have witnessed remarkable progress in vision-language models (VLMs) [HLG⁺24, ZWC⁺25, WBT⁺24, BCL⁺25]. These models have achieved impressive results across a wide range of downstream tasks, such as visual question answering [LLWL23, LLLL24], mathematical reasoning [ZHY⁺25, WQH⁺25], and human-agent interaction [HWL⁺24, QYF⁺25]. Building upon this progress, the field is increasingly moving toward vision-language-action (VLA) models, which extend the modalities of VLMs to incorporate action generation for robotic control [BBC⁺23, ZYX⁺23, DXS⁺23, MZH⁺23, KPK⁺24, LLZ⁺24]. These models aim to endow robots with the ability to understand visual environments, follow natural language instructions, and perform tasks autonomously. VLA models offer a unified framework to bridge perception, language understanding, and motor control, making them a promising paradigm for embodied AI.

However, deploying such large-scale VLA models in real-world robotic systems remains highly challenging, particularly on resource-constrained edge devices. These systems are often limited in terms of memory, computational throughput, and energy availability. Recent efforts in model quantization have shown that reducing the bit-width of model weights and activations can yield substantial improvements in efficiency. In particular, 1-bit large language models (LLMs) [WMD⁺23, MWM⁺24, MWH⁺25], where every parameter is restricted to ternary values (i.e., $\{-1, 0, 1\}$), have emerged as a compelling solution. These models achieve competitive performance on a variety of NLP benchmarks while dramatically reducing memory footprint, energy consumption, and inference latency. Moreover, the ternary parameter space enables efficient hardware execution and can simplify

deployment on edge accelerators. Despite their promise, existing 1-bit models have been largely confined to the language domain. To the best of our knowledge, their extension to multimodal tasks and robotic control has not yet been thoroughly explored.

In this work, we introduce **BitVLA**, the first 1-bit vision-language-action model for robotics manipulation, where every parameter is ternary, i.e., $\{-1, 0, 1\}$. BitVLA is built upon the publicly available 1-bit LLM BitNet b1.58 2B4T [MWH⁺25]. We begin by training a vision-language model using the 1-bit LLM in conjunction with a full-precision vision encoder, following the training paradigm of LLaVA [LLWL23]. To further reduce memory footprint, we introduce distillation-aware training to quantize the vision encoder to 1.58-bit weights and 8-bit activations. During this stage, we only train the vision encoder of the model, in which the full-precision encoder is used as the teacher model to better align the latent representations. Despite the absence of large-scale robotics pretraining, as shown in Figure 1, BitVLA achieves performance on par with the state-of-the-art model OpenVLA-OFT [KFL25] with 4-bit post-training quantization, while only using 29.8% memory footprint. These results demonstrate that BitVLA offers a cost-effective and high-performance solution for robotics manipulation, making it feasible for memory-constrained robotic systems.

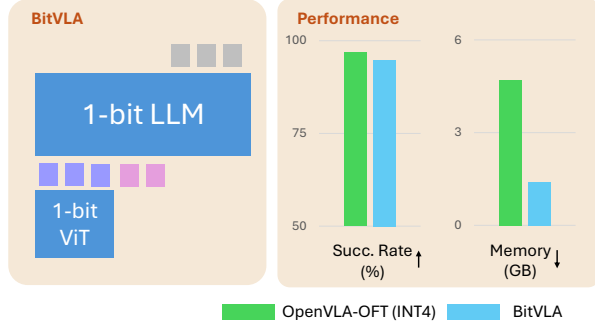


Figure 1: Comparison between BitVLA and OpenVLA-OFT with 4-bit post-training quantization in terms of end task performance and memory footprint. We report the average success rate on LIBERO benchmark.

2 Related Works

Vision-Language-Action models. Inspired by the rapid progress of VLMs, researchers in robotics have begun exploring VLA models that directly generate low-level control signals. The RT series [ZYX⁺23, ORM⁺] introduced Open X-Embodiment (OXE), a large-scale standardized robotics dataset, and used it to train RT-X, a generalist model for robotic manipulation tasks. OpenVLA [KPK⁺24] provided a detailed discussion on the design of VLA, covering aspects from the pretraining architecture to parameter-efficient fine-tuning methods and deployment strategies, while fully open-sourcing the training methods across all stages and the pre-trained model. RoboFlamingo [LLZ⁺24] leveraged pre-trained VLMs for single-step vision-language reasoning, introduced a policy head to capture sequential history, and required minimal fine-tuning via imitation learning. OpenVLA-OFT [KFL25] optimized the fine-tuning process by modeling continuous actions, employing parallel decoding, and applying action chunking from imitation learning [ZKLF23, CFD⁺23]. To improve inference efficiency, TinyVLA [WZL⁺24] adopts a compact 1.3B VLM backbone and skips pretraining to enhance data efficiency. Most recently, NORA [HSH⁺25] demonstrated competitive performance by utilizing Qwen2.5-VL-3B [BCL⁺25] as its backbone, enhanced with the FAST+ tokenizer for action generation.

Native 1-bit models. Modern deep learning research is increasingly focused on quantization-aware training and low-precision inference [PWW⁺23, XLCZ23, LLH⁺23]. Recent studies [WMD⁺23, MWM⁺24, KVM⁺24, ZZS⁺24, WMW24, WMW25] have demonstrated the potential of 1-bit and 1.58-bit pre-training for LLMs. [WMD⁺23] empirically showed that the performance gap between 1-bit and full-precision models narrows as the parameter count increases. Further, BitNet b1.58 [MWM⁺24] showed that 1.58-bit LLMs can match the performance of full-precision models starting from the 3B scale, while significantly reducing inference costs in terms of memory footprint, decoding latency, and energy consumption. OneBit [XHY⁺24] further explored the use of knowledge distillation for training binary LLMs. bitnet.cpp [WZS⁺25] developed an inference system optimized for 1-bit LLMs, substantially lowering energy consumption and improving decoding latency on CPU devices. More recently, [MWH⁺25] trained a 2B-parameter ternary LLM, achieving competitive performance relative to leading open-weight LLMs. The low memory and energy require-

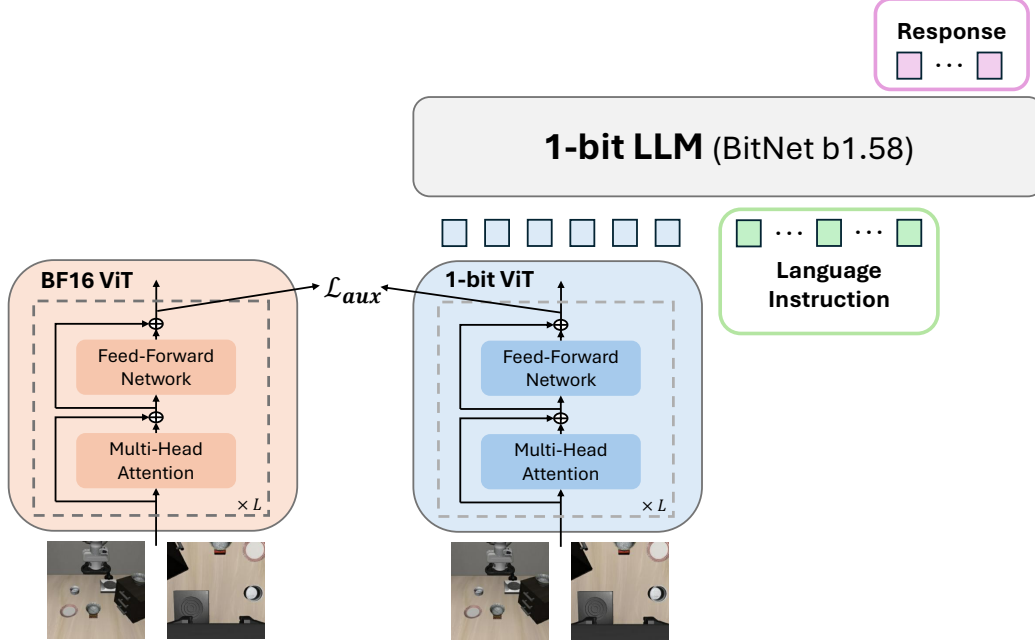


Figure 3: The overview of the distillation-aware training. The original full-precision encoder serves as the teacher model to ensure better alignment of the latent representations.

where $W \in \mathcal{R}^{m \times n}$ denotes the learnable weight of linear layer and $x \in \mathcal{R}^{n \times 1}$ denotes the inputs. The output of a ternary linear layer is computed as $Y = Q_w(W)Q_a(x)$, where Q_w and Q_a denote the quantization functions for weights and activations, respectively.

We apply quantization to all linear layers in the vision encoder, excluding the input and output embedding layers. BitVLA is trained with quantization-aware training, where quantization is performed on-the-fly during the forward pass. Due to the non-differentiable nature of quantization operations, we adopt the straight-through estimator (STE) [BLC13] to approximate gradients during backpropagation. Specifically, the gradients are passed directly through the quantization functions, following the approximation:

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial Q_w(W)}, \quad \frac{\partial \mathcal{L}}{\partial X} = \frac{\partial \mathcal{L}}{\partial Q_a(X)} \quad (4)$$

Both the gradients and optimizer states are maintained in full precision to preserve training stability.

3.2 Distillation-aware Training

In this subsection, we introduce the distillation-aware training to effectively quantize the vision encoder of VLM to 1.58 bit-widths. We illustrate the overview in Figure 3. We first initialize the latent weights of 1.58-bit encoder from its full-precision counterpart. Then we adopt the full-precision encoder as the teacher model. The training objective \mathcal{L}_{total} requires the minimization of both task-specific loss \mathcal{L}_{LM} and an auxiliary alignment loss \mathcal{L}_{aux} of latent representations between full-precision and 1.58-bit encoder.

Language modeling loss. The auto-regressive language modeling loss, \mathcal{L}_{LM} , is widely used in training VLMs. Let \mathcal{T} denote the input text sequence, which is divided into an instruction part \mathcal{T}_{ins} and a response part \mathcal{T}_{ans} . The visual tokens extracted by the 1.58-bit vision encoder are denoted as $\mathcal{V}_{1.58\text{-bit}}$. The language modeling loss can be formulated as:

$$\mathcal{L}_{LM} = - \sum_{\text{token}_i \in \mathcal{T}_{ans}} \log \Pr(\mathcal{Y}^i | \mathcal{V}_{1.58\text{-bit}}, \mathcal{T}^{[:i-1]})$$

where \mathcal{Y}^i represents the model’s predicted token at position i . The loss is computed only over the response tokens \mathcal{T}_{ans} , while the instruction and visual tokens are provided as context.

Representations alignment loss. To enhance the alignment between the latent representations of the 1.58-bit and full-precision vision encoders, we learn the 1.58-bit encoder through knowledge distillation, in which the full-precision encoder is used as the teacher model. Let h_{bf16}^l and $h_{1.58\text{-bit}}^l$ denote the outputs of the l -th layer from the full-precision and 1.58-bit vision encoders, respectively. The alignment loss is defined as:

$$\mathcal{L}_{\text{aux}} = \frac{1}{n} \sum_{l=1}^L \|h_{\text{bf16}}^l - h_{1.58\text{-bit}}^l\|^2$$

where n is the hidden dimension and L is the total number of layers in the vision encoder. This auxiliary loss encourages the 1.58-bit vision encoder to mimic the representational behavior of its full-precision counterpart.

Above all, the training objective $\mathcal{L}_{\text{total}}$ is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{LM}} + \gamma \cdot \mathcal{L}_{\text{aux}} \quad (5)$$

where γ is a coefficient for representations alignment. During the distillation-aware training, only the vision encoder is trainable while the other components (i.e., LLM and connector) are frozen. In our experiments, we observe that, unlike the 1.58-bit pre-training of LLMs, the quantization-aware training of 1.58-bit encoder is highly data-efficient with distillation from a full-precision teacher model. It preserves most of the performance of its full-precision counterpart using only billions of training tokens.

3.3 Robotics Fine-tuning

In this subsection, we describe the fine-tuning procedure of BitVLA for specific robotics tasks. Following OpenVLA-OFT [KFL25], we utilize parallel decoding and action chunking techniques to enhance the throughput of VLA models. Specifically, we replace the conventional causal mask used in LLMs with a bidirectional attention mask, enabling each forward pass to generate a coherent action trajectory over multiple time steps. This approach significantly boosts real-time control efficiency compared to autoregressive, token-by-token predictions. Additionally, we integrate an MLP-based action head to project the latent representations of query tokens into continuous robotic action space. The model is trained to minimize the L_1 loss between predicted actions and ground-truth trajectories.

4 Experiments

4.1 Model Training

BitVLA is trained with a three-stage procedure. Following LLaVA [LLWL23], we first train the connector to align the vision encoder with the LLM using the LLaVA 1.5-558k dataset [LLLL24]. In the second stage, we freeze the vision encoder and train both the LLM and the connector on a 10-million-sample subset of MammoTH-VL [GZB⁺24], consisting of single-image samples. In the final stage, we train the vision encoder from full-precision (W16A16) to 1.58-bit weights and 8-bit activations (W1.58A8) on a 5-million-sample subset from the data in the second stage. The training data in Stage III contains up to 10B tokens. The distillation loss on latent representations is weighted by a coefficient $\gamma = 0.1$. As recommended by [MWM⁺24], we use a large learning rate for instruction tuning. The training requires 14 days on 8 NVIDIA A100 cards with 80GB memory. We present the detailed hyperparameter configurations in Appendix A.

4.2 Experiments on Robotics Manipulation

Benchmark. We adopt the LIBERO simulation environment [LZG⁺23] to evaluate the generalization and performance of robotics manipulation models. As shown in Figure 4, this benchmark assesses robotic intelligence across four critical dimensions: spatial generalization (manipulating

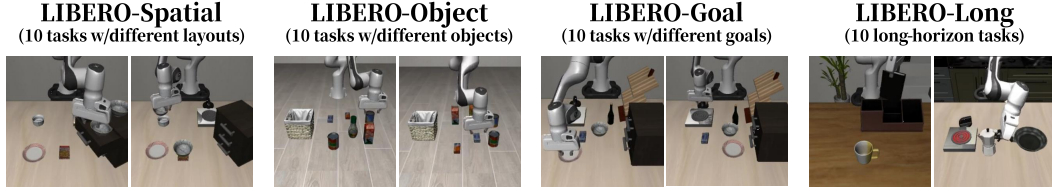


Figure 4: The overview of LIBERO benchmark task suites. It has four different dimensions to evaluate the generalization and performance of robotics manipulation models.

Table 1: The success rate (%) of BitVLA and the baselines on LIBERO simulation environment.

Models	Size	Memory Usage↓	Spatial	Object	Goal	Long	Avg.
<i>w/ Robotics pre-training</i>							
OpenVLA [KPK ⁺ 24]	7.5B	15.1GB (10.79×)	84.7	88.4	79.2	53.7	76.5
SpatialVLA [QSC ⁺ 25]	4.2B	8.5GB (6.07×)	88.2	89.9	78.6	55.5	78.1
CoT-VLA [ZLK ⁺ 25]	8.0B	16.2GB (11.57×)	87.5	91.6	87.6	69.0	81.1
NORA-Long [HSH ⁺ 25]	3.8B	7.5GB (5.36×)	92.2	95.4	89.4	74.6	87.9
π_0 [BBD ⁺ 24]	3.5B	7.0GB (5.00×)	96.8	98.8	95.8	85.2	94.2
OpenVLA-OFT [KFL25]	7.7B	15.4GB (11.00×)	97.6	98.4	97.9	94.5	97.1
<i>w/o Robotics pre-training</i>							
OpenVLA-OFT [KFL25]	7.7B	15.4GB (11.00×)	94.3	95.2	91.7	86.5	91.9
BitVLA (ours)	3.0B	1.4GB (1.00×)	97.4	99.6	94.4	87.6	94.8

objects arranged in novel configurations), object generalization (adapting to previously unseen object categories), goal generalization (interpreting diverse language instructions), and long-horizon reasoning (performing multi-stage tasks involving varied objects, layouts, and objectives). These capabilities are systematically evaluated through four corresponding task suites, namely LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. Each task suite contains 500 expert demonstrations systematically distributed across 10 distinct manipulation tasks. Additional details are included in Appendix B.

Implementation details. We use the same training dataset¹ as OpenVLA-OFT [KFL25] during fine-tuning. We process synchronized multi-view visual inputs from both wrist-mounted and external cameras, while encoding proprioceptive signals such as end-effector positions. The physical state measurements are projected into a single token using an MLP-based projector, which is then appended to the image tokens. For action chunking, we set chunk size to $K = 8$ following OpenVLA-OFT, and execute full chunks before re-planning.

We perform full-parameter fine-tuning for faster convergence across all experiments. Specifically, BitVLA is fine-tuned for 10k steps on LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal, and for 100k steps on LIBERO-Long. We adopt a cosine decay learning rate schedule with a batch size of 64. The 10k-step fine-tuning process takes approximately 4 hours on 8 NVIDIA A100 cards with 80GB of memory. More details can be found in Appendix A.

Baselines. We compare BitVLA with the baselines under supervised fine-tuning on the LIBERO dataset, including OpenVLA-OFT [KFL25], OpenVLA [KPK⁺24], SpatialVLA [QSC⁺25], CoT-VLA [ZLK⁺25], NORA-Long [HSH⁺25], and π_0 [BBD⁺24]. Specifically, π_0 employs the flow-matching architecture built upon a pre-trained VLM. NORA is trained from a strong lightweight VLM Qwen2.5-VL-3B [BCL⁺25] to improve the efficiency. We adopt its NORA-Long variant, which generates five-step action sequences at a time. CoT-VLA introduces visual chain-of-thought reasoning by predicting future frames autoregressively before action generation. SpatialVLA incorporates 3D information and learns a generalist policy for manipulation. OpenVLA is a 7B open-source VLA model trained on the OXE dataset, surpasses closed models like RT-2-X [ORM⁺] in numerous tasks. OpenVLA-OFT is fine-tuned from OpenVLA using a series of techniques, e.g., parallel decoding and continuous action modeling, to improve the speed and performance for specific task. Due to

¹https://huggingface.co/datasets/openvla/modified_libero_rlds

Table 2: The success rate (%) of BitVLA and OpenVLA, OpenVLA-OFT with post-training quantization on LIBERO simulation environment.

Models	Memory Usage↓	Spatial	Object	Goal	Long	Average
<i>INT8 post-training quantization</i>						
OpenVLA [KPK ⁺ 24]	7.4GB (5.29×)	86.4	85.2	77.2	58.8	76.9
OpenVLA-OFT [KFL25]	7.7GB (5.50×)	98.8	98.0	96.6	94.0	96.7
<i>INT4 post-training quantization</i>						
OpenVLA [KPK ⁺ 24]	4.4GB (3.14×)	83.0	84.0	72.0	51.6	72.7
OpenVLA-OFT [KFL25]	4.7GB (3.36×)	98.2	98.2	97.2	93.8	96.9
BitVLA (ours)	1.4GB (1.00×)	97.4	99.6	94.4	87.6	94.8

Table 3: The zero-shot accuracy of BitVLA with full-precision and 1.58-bit vision encoder (VE) on visual question answering tasks.

Models	MMMU (val)	SeedBench (image)	SeedBench ²⁺ (test)	MMStar (test)	AI2D (test)	Avg.
BitVLA w/ 16-bit VE	37.4	70.6	45.0	43.6	68.6	53.0
BitVLA w/ 1.58-bit VE	35.4	69.3	43.7	41.5	67.6	51.5

resource constraints, BitVLA is not pre-trained on large-scale robotics datasets. Therefore, we also report OpenVLA-OFT results fine-tuned directly from its base VLM [KNB⁺24] for reference.

Main results. Table 1 summarizes the success rates of BitVLA and various baselines on the LIBERO benchmark suites. As shown in the table, although BitVLA is not pre-trained on a large-scale robotics dataset (e.g., Open X-Embodiment [ORM⁺]), it still surpasses strong baselines with 3 billion parameters, including π_0 and NORA-Long. In particular, BitVLA outperforms π_0 by 2.4% on LIBERO-Long, highlighting its effectiveness in long-horizon reasoning tasks for robotic manipulation. Moreover, BitVLA has a lightweight memory footprint of only 1.4GB, making it feasible to deploy on a single consumer-grade GPU such as the NVIDIA GeForce RTX 3050 Ti Laptop (4GB).

Compared to the larger OpenVLA-OFT model, BitVLA achieves comparable performance on the Spatial, Object, and Goal subsets of the LIBERO benchmark, but still falls short on LIBERO-Long. We attribute this gap to OpenVLA-OFT’s fine-tuning from OpenVLA, which benefits from large-scale robotics pre-training and thus excels at complex manipulation tasks. As shown in the table, pre-training on a large-scale robotics dataset boosts OpenVLA-OFT’s success rate on LIBERO-Long from 86.5% to 94.5%. Notably, when compared to the OpenVLA-OFT variant without robotics pre-training, BitVLA achieves comparable performance on LIBERO-Long.

Comparison with post-training quantization. We compare BitVLA with OpenVLA and OpenVLA-OFT models subjected to 8-bit and 4-bit post-training quantization. We use their publicly available fine-tuned checkpoints released on Hugging Face. For quantization, we employ the bit-sandbytes toolkit [DLBZ22] to convert the model backbones to INT8 and INT4 precision. We report both the memory footprint and performance of the quantized models on the LIBERO benchmark. As shown in Table 2, OpenVLA exhibits a larger performance degradation under 4-bit quantization compared to OpenVLA-OFT. Notably, BitVLA achieves performance comparable to 4-bit quantized OpenVLA-OFT while using less than one-third of the memory.

4.3 Experiments on Visual Question Answering

We evaluate the zero-shot performance of BitVLA with both full-precision and 1.58-bit vision encoders on visual question answering (VQA) tasks. The evaluation suite includes MMMU [YNZ⁺24], SeedBench [LWW⁺23], SeedBench-2-Plus [LGC⁺24], MMStar [CLD⁺24], and AI2D [KSK⁺16]. We adopt the publicly available LMM-Eval toolkit [ZLZ⁺24] to ensure fair and consistent comparisons. As shown in Table 3, BitVLA equipped with the 1.58-bit encoder achieves performance comparable to its full-precision counterpart. Specifically, the 1.58-bit encoder results in only a 1.5%

Table 4: The ablations on data size and representation alignment loss of distillation-aware training for visual question answering tasks.

Training Tokens (Stage III)	\mathcal{L}_{aux}	MMMU (val)	SeedBench (image)	SeedBench ²⁺ (test)	MMStar (test)	AI2D (test)	Avg.
10B	✓	35.4	69.3	43.7	41.5	67.6	51.5
5B	✓	33.3	69.1	43.3	41.4	66.4	50.8
5B	✗	32.4	52.9	38.8	30.7	57.5	42.4

Table 5: The ablations on data size and representation alignment loss of distillation-aware training on LIBERO benchmark suites.

Training Tokens (Stage III)	\mathcal{L}_{aux}	Spatial	Object	Goal	Long	Average
10B	✓	97.4	99.6	94.4	87.6	94.8
5B	✓	96.8	98.6	93.8	85.2	93.6
5B	✗	96.2	98.6	91.4	85.2	92.9

average accuracy drop across the five benchmarks, while reducing its memory footprint from 0.8GB to 0.1GB. These results demonstrate that distillation-aware training effectively preserves performance on general VQA tasks while significantly lowering memory consumption during inference.

4.4 Ablation Studies

Representations alignment loss. We perform ablation studies to assess the impact of the proposed representation alignment loss during distillation-aware training. As shown in Table 4, incorporating the alignment loss significantly boosts the zero-shot performance of BitVLA with the 1.58-bit vision encoder, raising the average accuracy from 42.4% to 50.8% across five VQA benchmarks. On the LIBERO benchmark suites, where models are fine-tuned for specific tasks, the performance gain is smaller but still meaningful. As reported in Table 5, the alignment loss leads to a 2.4% increase on LIBERO-Goal set.

Data size of distillation-aware training. We compare the performance of BitVLA trained with 5B and 10B tokens during the distillation-aware training (Stage III). As shown in Table 4, increasing the training data during the quantization-aware training of the vision encoder improves overall performance on general VQA tasks. Specifically, BitVLA trained with 10B tokens in Stage III surpasses the 5B-token counterpart by 0.7% in average accuracy. Additionally, on the LIBERO benchmark, the 10B-token model achieves a 1.2% gain in average accuracy after fine-tuning.

5 Qualitative Analysis

In this section, we carefully analyze the failure cases of BitVLA on LIBERO benchmark suites, categorizing them into three types: spatial localization discrepancy, goal misunderstanding, and trajectory planning failure.

- **Spatial localization discrepancy** refers to the failure caused by inaccuracies in pose prediction during manipulation. Typically, such errors occur in four main scenarios: (1) manipulating objects with unstable centers of gravity, such as wine bottles, where small miscalculations lead to instability (2) selecting imprecise grasping poses, causing objects to topple during approach or drop during transportation (3) phantom manipulation attempts in the absence of physical objects (4) positioning errors when placing objects at target locations, resulting in task failures. These issues may arise from either coarse spatial understanding derived from visual encoder or over-reliance on proprioceptive signals relative to visual information under certain operational conditions in BitVLA.
- **Goal misunderstanding** refers to failure arising from BitVLA’s incorrect interpretation or following of the language instructions. A typical scenario of this error occurs when the robot

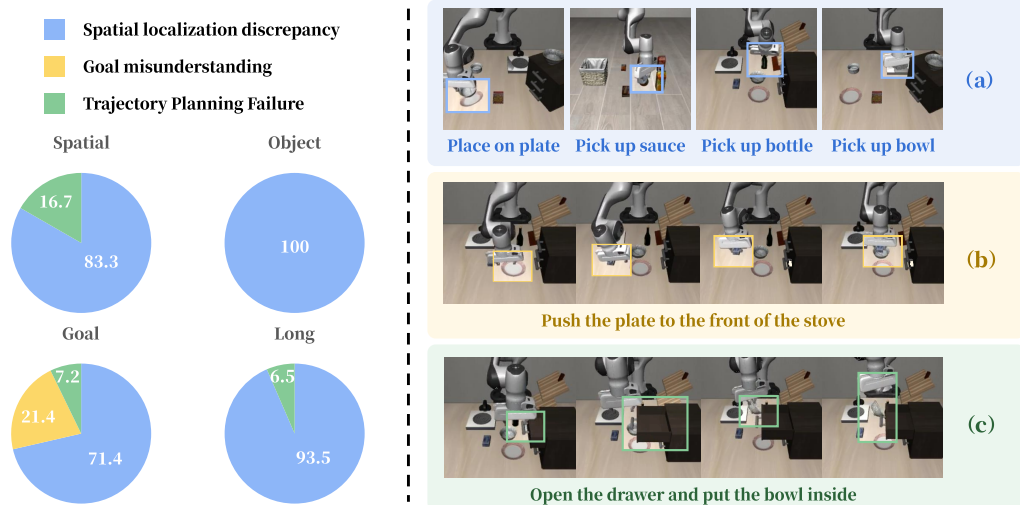


Figure 5: **Left:** distribution of failure types across each task suite on the LIBERO simulation environment. **Right:** typical examples for (a) spatial localization discrepancy, (b) goal misunderstanding and (c) trajectory planning failure.

erroneously interacts with non-target objects during task execution, subsequently initiating another task rollout associated with the contacted objects. We hypothesize that this primarily arises from the dominance of visual and proprioceptive information during the model’s reasoning process at the moment of goal switching. This misalignment is also mentioned in OpenVLA-OFT [KFL25], which proposes OpenVLA-OFT+ augmented variant employing a FiLM strategy to alleviate this issue.

- **Trajectory Planning Failure** refers to execution errors caused by collisions during motion planning. A typical case involves the robotic arm colliding with the lower panel of an open drawer during bowl placement, leading to bowl drops or the arm becoming jammed. These failures highlight the need for BitVLA to better leverage prior knowledge for generating more rational and collision-free trajectories. Moreover, the system should anticipate the feasibility of subsequent sub-goals (e.g., placing a bowl) when executing earlier ones (e.g., opening a drawer) to avoid operational conflicts. For instance, partially opening the drawer may reduce trajectory complexity and lower the risk of collision.

We illustrate the distribution and specific examples for each failure type in Figure 5. The most frequent failure type across all task suites is spatial localization discrepancy. We found that the success criteria of LIBERO are very strict, especially in placement tasks, where success is only determined if the object is placed precisely in the center of the plate. However, in many failure cases within the Long suite, objects were successfully placed on the plate but still failed because they were not centered. Nevertheless, the large number of errors in this category indicates that dexterous manipulation tasks remain the biggest bottleneck for BitVLA.

6 Conclusion

We present BitVLA, the first 1-bit vision-language-action model for robotics manipulation, where every parameter is constrained to ternary values. BitVLA is initially trained using a 1-bit LLM and a full-precision vision encoder. To further compress the vision encoder, we introduce a distillation-aware training strategy that converts its weights to ternary values. In this stage, the full-precision encoder serves as the teacher model to guide the alignment of latent representations. Experimental results on the LIBERO benchmark show that BitVLA achieves performance on par with the state-of-the-art OpenVLA-OFT model with 4-bit post-training quantization, while reducing memory usage by up to $3.36\times$. These results highlight BitVLA as a cost-effective and efficient solution for robotics applications on memory-constrained hardware.

References

- [BBC⁺23] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.
- [BBD⁺24] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024.
- [BCL⁺25] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025.
- [BLC13] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.
- [CFD⁺23] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.
- [CLD⁺24] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.
- [DLBZ22] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.
- [DXS⁺23] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 8469–8488. PMLR, 2023.
- [GZB⁺24] Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhui Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. 2024.
- [HLG⁺24] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone,

- Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codisoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024.
- [HSH⁺25] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.
- [HWL⁺24] Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for GUI agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 14281–14290. IEEE, 2024.
- [KFL25] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *CoRR*, abs/2502.19645, 2025.
- [KNB⁺24] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.
- [KPK⁺24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2024.
- [KSK⁺16] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.
- [KVM⁺24] Ayush Kaushal, Tejas Vaidhya, Arnab Kumar Mondal, Tejas Pandey, Aaryan Bhagat, and Irina Rish. Spectra: Surprising effectiveness of pretraining ternary language models at scale. *arXiv preprint arXiv:2407.12327*, 2024.
- [LGC⁺24] Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024.
- [LLH⁺23] Shih-Yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. LLM-FP4: 4-bit floating-point quantized transformers. In *EMNLP 2023*, pages 592–605. Association for Computational Linguistics, 2023.
- [LLLL24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.
- [LLWL23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36*, 2023.

- [LLZ⁺24] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [LWW⁺23] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *CoRR*, abs/2307.16125, 2023.
- [LZG⁺23] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [MWH⁺25] Shuming Ma, Hongyu Wang, Shaohan Huang, Xingxing Zhang, Ying Hu, Ting Song, Yan Xia, and Furu Wei. Bitnet b1. 58 2b4t technical report. *arXiv preprint arXiv:2504.12285*, 2025.
- [MWM⁺24] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *CoRR*, abs/2402.17764, 2024.
- [MZH⁺23] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhui Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [ORM⁺] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alexander Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew E. Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Paul Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Haoshu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I. Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booyer, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minh Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J. Joshi, Niko Sünderhauf, Ning

- Liu, Norman Di Palo, Nur Muhammad (Mahi) Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R. Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaesan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham D. Sonawani, Shuran Song, Sichun Xu, Siddhant Halder, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Liangwei Xu, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and RT-X models : Open x-embodiment collaboration. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 6892–6903. IEEE.
- [PWW⁺23] Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, Ruihang Li, Miaosen Zhang, Chen Li, Jia Ning, Ruizhe Wang, Zheng Zhang, Shuguang Liu, Joe Chau, Han Hu, and Peng Cheng. FP8-LM: training FP8 large language models. *CoRR*, abs/2310.18313, 2023.
- [QSC⁺25] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *CoRR*, abs/2501.15830, 2025.
- [QYF⁺25] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjuan Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. UI-TARS: pioneering automated GUI interaction with native agents. *CoRR*, abs/2501.12326, 2025.
- [WBT⁺24] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *CoRR*, abs/2409.12191, 2024.
- [WMD⁺23] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. *CoRR*, abs/2310.11453, 2023.
- [WMW24] Hongyu Wang, Shuming Ma, and Furu Wei. Bitnet a4.8: 4-bit activations for 1-bit llms. *CoRR*, abs/2411.04965, 2024.
- [WMW25] Hongyu Wang, Shuming Ma, and Furu Wei. Bitnet v2: Native 4-bit activations with hadamard transformation for 1-bit llms. *arXiv preprint arXiv:2504.18415*, 2025.
- [WQH⁺25] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhua Chen. VI-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025.

- [WZL⁺24] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *CoRR*, abs/2409.12514, 2024.
- [WZS⁺25] Jinheng Wang, Hansong Zhou, Ting Song, Shijie Cao, Yan Xia, Ting Cao, Jianyu Wei, Shuming Ma, Hongyu Wang, and Furu Wei. Bitnet. cpp: Efficient edge inference for ternary llms. *arXiv preprint arXiv:2502.11880*, 2025.
- [XHY⁺24] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models. In *Advances in Neural Information Processing Systems* 38, 2024.
- [XLCZ23] Haocheng Xi, Changhao Li, Jianfei Chen, and Jun Zhu. Training transformers with 4-bit integers. In *Advances in Neural Information Processing Systems* 36, 2023.
- [YNZ⁺24] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.
- [ZHY⁺25] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-VL: learning to reason with multimodal large language models via step-wise group relative policy optimization. *CoRR*, abs/2503.12937, 2025.
- [ZKLF23] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.
- [ZLK⁺25] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *CoRR*, abs/2503.22020, 2025.
- [ZLZ⁺24] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024.
- [ZMKB23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023.
- [ZWC⁺25] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.
- [ZYX⁺23] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023.

[ZZS⁺24] Rui-Jie Zhu, Yu Zhang, Ethan Sifferman, Tyler Sheaves, Yiqiao Wang, Dustin Richmond, Peng Zhou, and Jason K. Eshraghian. Scalable matmul-free language modeling. *CoRR*, abs/2406.02528, 2024.

A Hyper-parameters

We present the hyperparameter configurations used for training BitVLA in Table 6. Following the recommendations of [MWM⁺24], we employ a two-stage weight decay schedule during visual instruction tuning. For fine-tuning on the LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal suites, we report the best results selected from learning rates in the set 5e-5, 1e-4, 3e-4. For LIBERO-Long, all models are trained with a peak learning rate of 8e-5 for the vision encoder and 4e-4 for the LLM.

Table 6: Hyper-parameters for the training of BitVLA.

Hyper-parameter	Stage I	Stage II	Stage III
Peak Learning rate	1e-3	3e-4	1e-4
Batch Size	256	256	256
Weight decay	\times	0.1 \rightarrow 0	0.01
Trainable modules	Connector	LLM, Connector	ViT
Training steps	25k	40k	20k
Training sequence	1024	2048	2048
Vision sequence		256	
Learning rate scheduling		polynomial decay	
AdamW β		(0.9, 0.999)	
AdamW ϵ		1e-8	
Gradient Clipping		1.0	
Dropout		\times	
Attention Dropout		\times	

Table 7: Hyper-parameters for the fine-tuning of BitVLA on LIBERO dataset.

Hyper-parameter	Spatial	Object	Goal	Long
Peak Learning rate	{5e-5, 1e-4, 3e-4}			4e-4, 8e-5
Training steps	10k	10k	10k	100k
Learning rate scheduling		cosine decay		
Warmup steps		375		
Batch Size		64		
Weight decay		0.01		
Trainable modules		LLM, Connector, ViT		
AdamW β		(0.9, 0.999)		
AdamW ϵ		1e-8		
Gradient Clipping		\times		

B Tasks in LIBERO

In this section, we present the detailed task compositions of each task suite in LIBERO. As shown in Table 8, it demonstrates the distinct task configurations across the four task suites within the LIBERO framework. Figure 6 illustrates the scene visualizations for a subset of tasks.

Table 8: Task description in LIBERO benchmark task suites.

Task suite	Task description
Spatial	<p>pick up the black bowl between the plate and the ramekin and place it on the plate</p> <p>pick up the black bowl next to the ramekin and place it on the plate</p> <p>pick up the black bowl from table center and place it on the plate</p> <p>pick up the black bowl on the cookie box and place it on the plate</p> <p>pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate</p> <p>pick up the black bowl on the ramekin and place it on the plate</p> <p>pick up the black bowl next to the cookie box and place it on the plate</p> <p>pick up the black bowl on the stove and place it on the plate</p> <p>pick up the black bowl next to the plate and place it on the plate</p> <p>pick up the black bowl on the wooden cabinet and place it on the plate</p>
Object	<p>pick up the alphabet soup and place it in the basket</p> <p>pick up the cream cheese and place it in the basket</p> <p>pick up the salad dressing and place it in the basket</p> <p>pick up the bbq sauce and place it in the basket</p> <p>pick up the ketchup and place it in the basket</p> <p>pick up the tomato sauce and place it in the basket</p> <p>pick up the butter and place it in the basket</p> <p>pick up the milk and place it in the basket</p> <p>pick up the chocolate pudding and place it in the basket</p> <p>pick up the orange juice and place it in the basket</p>
Goal	<p>open the middle drawer of the cabinet</p> <p>put the bowl on the stove</p> <p>put the wine bottle on top of the cabinet</p> <p>open the top drawer and put the bowl inside</p> <p>put the bowl on top of the cabinet</p> <p>push the plate to the front of the stove</p> <p>put the cream cheese in the bowl</p> <p>turn on the stove</p> <p>put the bowl on the plate</p> <p>put the wine bottle on the rack</p>
Long	<p>put both the alphabet soup and the tomato sauce in the basket</p> <p>put both the cream cheese box and the butter in the basket</p> <p>turn on the stove and put the moka pot on it</p> <p>put the black bowl in the bottom drawer of the cabinet and close it</p> <p>put the white mug on the left plate and put the yellow and white mug on the right plate</p> <p>pick up the book and place it in the back compartment of the caddy</p> <p>put the white mug on the plate and put the chocolate pudding to the right of the plate</p> <p>put both the alphabet soup and the cream cheese box in the basket</p> <p>put both moka pots on the stove</p> <p>put the yellow and white mug in the microwave and close it</p>

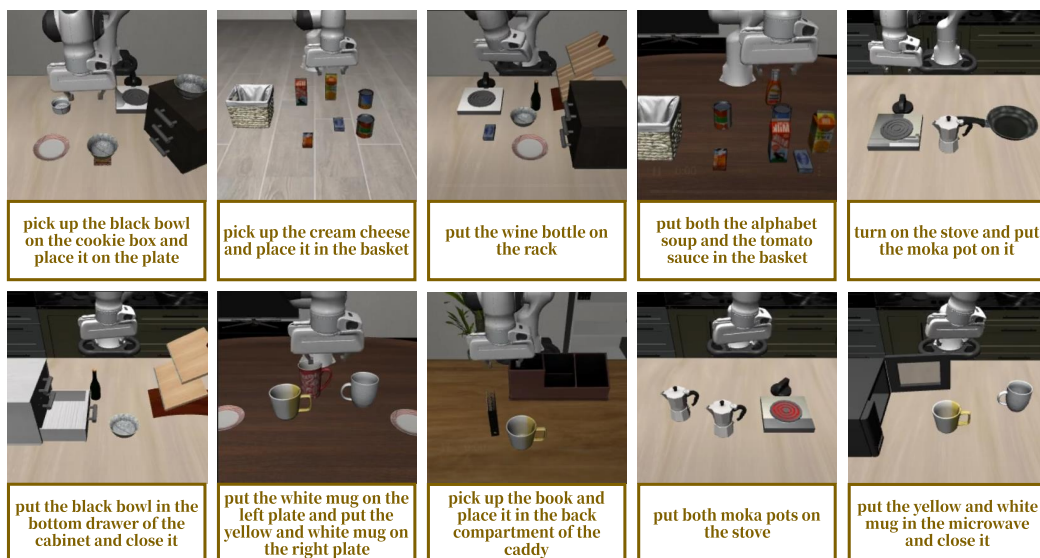


Figure 6: Examples in LIBERO benchmark tasks suites.