LEAP: Logical Embodied Action Planning for Long-Horizon Robotic Tasks via Generative Vision-Language Alignment

Anonymous ACL submission

Abstract

Vision-Language-Action models (VLAs) have emerged as a promising paradigm for generalizable sensorimotor control by leveraging pretrained vision-language models. However, despite their efficacy in learning direct input-output mappings, current VLAs struggle with long-horizon tasks that demand an understanding of physical constraints and logical reasoning. In this paper, we introduce LEAP (Logical Embodied Action Planning), a framework that empowers a compact 2B VLM to master complex, multi-step planning tasks via Full Parameter Supervised Fine-Tuning. LEAP learns to generate coherent action blueprints directly from single observations, effectively bridging the gap between high-level reasoning and low-level execution. Experimental results on VLABench demonstrate that LEAP achieves superior performance, particularly in the Physics Law dimension, where it outperforms significantly larger baselines (e.g., 3B and 8B models). Specifically, LEAP achieves a score of 30.3 on the Physics Law dimension, surpassing Qwen2.5-VL (17.4) by a substantial margin.

1 Introduction

A fundamental challenge in embodied AI is enabling robotic agents to execute long-horizon tasks that demand not only low-level control but also high-level logical and strategic reasoning. While existing policies trained for individual skills can adapt to local variations in object position or lighting (Brohan et al., 2023; Chi et al., 2023), they often struggle to chain these skills into coherent sequences for abstract goals—such as "tidying a cluttered living room" or "preparing a meal"—where the causal dependencies between steps are critical (Ahn et al., 2023). Foundation models for vision and language (VLMs) like GPT-4V (OpenAI et al., 2023) and Qwen-VL (Bai et al., 2023) have demonstrated success in acquiring physical commonsense from large-scale pretraining. In this work, we treat VLMs as generative planners that ground complex abstract instructions into actionable sequences.

Towards this goal, existing academic research work has explored two primary paradigms. The first is *Reactive Policies* (e.g., RT-2 (Zitkovich et al., 2023), OpenVLA (Kim et al., 2025)), which predict control actions step-by-step. While effective for short-term interactions, these models lack a global temporal horizon, often "forgetting" the ultimate goal or getting stuck in loops during long sequences due to error accumulation. The second paradigm relies on *Explicit Planners* (e.g., SayCan (Ahn et al., 2023), VoxPoser (Huang et al., 2023)), which use Large Language Models (LLMs) to decompose tasks via explicit Chain-of-Thought (CoT) reasoning. However, widespread deployment of such planners is hindered by two key limitations: 1) *perceptual detachment*, where models are often loosely grounded, generating plans that are internally logically sound but physically infeasible (e.g., asking to pick up an occluded object); and 2) *inference latency*, as the heavy reliance on verbose CoT reasoning introduces significant computational overhead, making them impractical for efficient, real-time autonomous robot operation.

To this end, we introduce LEAP (Logical Embodied Action Planning), a specialized 2B-parameter visual planning framework for efficient, long-horizon autonomous robot task planning. LEAP treats the planning problem as visual-conditioned sequence generation. Unlike robotic transformers that react moment-to-moment, LEAP leverages the native multimodal capabilities of Qwen3-VL (Bai et al., 2025a) to generate a complete, coherent *action blueprint* in a single shot from the initial visual observation. Crucially, we employ a Full Parameter Supervised Fine-Tuning (Full SFT) strategy to generate the actionable plan without intermediate CoT tokens. This approach allows LEAP to encourage stronger semantic cross-modal alignment,

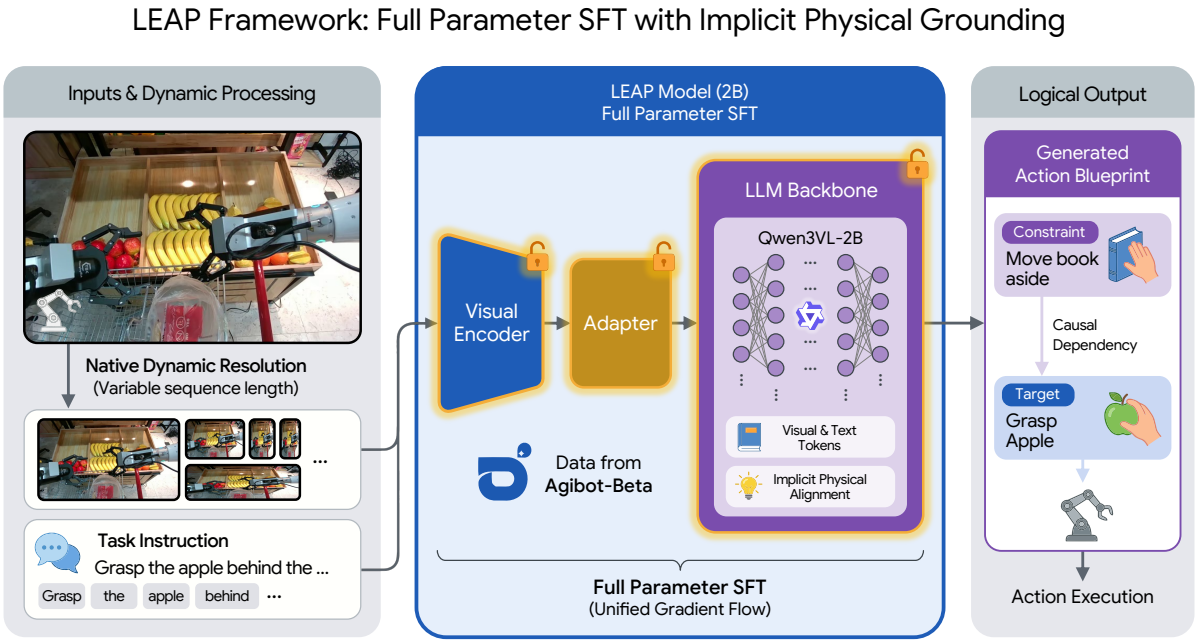

Figure 1: **The LEAP Framework Architecture.** The framework is structured into three zones: Zone 1 (Inputs & Dynamic Processing) employs Naive Dynamic Resolution to process variable-length visual sequences alongside task instructions ($\mathcal{L}_{task}$); Zone 2 (LEAP Model) features a 2B-parameter backbone where the Visual Encoder, Adapter, and Transformer Layers are optimized via Full Parameter SFT, facilitating Implicit Physical Alignment and unified gradient flow; Zone 3 (Logical Output) generates a structured Action Blueprint that resolves causal dependencies (e.g., handling constraints like "Move Book aside" before targets) to guide physical Action Execution.

optimizing the model to improve physical reasoning metrics (such as prerequisite relationships and object affordances) as measured on VLABench.

As a result of this training approach, LEAP demonstrates strong results on the VLABench benchmark (Zhang et al., 2025). It achieves competitive scores across official VLABench dimensions, showing that a compact 2B model can develop effective planning capabilities without the computational overhead of explicit CoT.

Our main contributions are as follows:

- We propose LEAP, a streamlined end-to-end visual action planner that enables compact VLMs (2B) to perform complex long-horizon planning tasks via one-shot generation.
- We demonstrate the effectiveness of *implicit logic learning* via Full SFT, showing that the model consistently improves physics-related scores on VLABench without generating explicit CoT tokens at inference time.
- We conduct comprehensive evaluations on VLABench, where LEAP achieves strong performance under the official evaluation protocol, obtaining higher scores than reactive and LLM-based baselines on key dimensions.

2 Related Work

Vision-Language Models as High-Level Planners A growing body of prior work leverages the web-scale semantic knowledge of Vision-Language Models (VLMs) for autonomous robotic control (Hu et al., 2024b; Shridhar et al., 2022; Schakkal et al., 2025). A prevalent paradigm treats VLMs as modular reasoning engines that decompose complex high-level instructions into sub-goals. For instance, ViLa (Hu et al., 2024b) utilizes GPT-4V to generate natural language plans from visual observations, while VLM-TAMP (Shridhar et al., 2022) and ViLaIn (Mewes et al., 2025) interface VLMs with geometric Task and Motion Planning (TAMP) or PDDL solvers to ensure physical feasibility. Similarly, Hierarchical VLP (Schakkal et al., 2025) adopts a layered architecture where a high-level VLM delegates skill selection to low-level controllers. However, these modular pipelines are prone to *cascading errors* and information bottlenecks between the planner and the executor. In contrast, LEAP adopts a unified architecture, treat-

ing perception and planning as a seamless *visual-conditioned sequence generation* process. By utilizing full-parameter fine-tuning to encode planning patterns directly into the model weights, LEAP eliminates the dependency on external symbolic solvers and improves overall inference efficiency.

Implicit Reasoning & Pattern Learning Recent research challenges the necessity of verbose, explicit Chain-of-Thought (CoT) reasoning for expert-level manipulation (Xiong et al., 2024). While explicit CoT enhances interpretability in text domains (Zhao et al., 2025), its deployment in real-time autonomous robotics introduces prohibitive latency. ReLEP (Xiong et al., 2024) demonstrates that through rigorous data curation, complex logical dependencies can be implicitly internalized by VLM policies without generating intermediate reasoning tokens, effectively mitigating hallucination. VLP-Model (Zhou et al., 2024) further explores this direction by training models to output structured API call sequences. LEAP aligns with this philosophy of *implicit reasoning* but diverges in its output representation. Unlike approaches restricted to rigid API calls, LEAP operates in a flexible *natural language action space* (via the *Action Blueprint*). This design effectively bridges the broad generalization capabilities of open-vocabulary VLMs with the deterministic needs of physical robot execution, allowing for intuitive planning without the computational overhead of explicit CoT.

Visual-Conditioned Sequence Generation Framing autonomous robot planning as a sequence generation task is a promising direction for generalizable robotic control (Du et al., 2024; Wang et al., 2025; Sermanet et al., 2024; Dalal et al., 2024). Works such as Video Language Planning (VLP) (Du et al., 2024) and This&That (Wang et al., 2025) conceptualize planning as the generation of future video frames or latent visual trajectories. Alternatively, RoboVQA (Sermanet et al., 2024) approaches planning as a multi-turn visual question answering problem. While promising, generating high-fidelity video plans is computationally expensive and prone to ambiguity. LEAP simplifies this landscape by casting the "visual plan" not as pixels or latent vectors, but as a structured textual list—an *Action Blueprint*. This output format offers three distinct advantages: (1) it ensures human interpretability; (2) it leverages the VLM's native textual proficiency while avoiding the cost of video generation; and (3) it provides a compact, structured interface that is easily parsable by downstream controllers.

3 Methodology

In this section, we introduce LEAP, a unified framework that empowers compact Vision-Language Models to perform long-horizon robotic planning through implicit logic learning. As depicted in Figure 1, we treat the planning problem not as a sequence of modular API calls, but as an end-to-end *visual-conditioned sequence generation* task. To this end, we first provide the theoretical background and detail our backbone's key capabilities in Section 3.1. We then provide a formal textual definition of the problem and the *Action Blueprint* output space in Section 3.2. The data construction pipeline is described in Section 3.3, detailing how raw teleoperation episodes are processed into logic-aligned instruction pairs. Finally, Section 3.4 presents the model architecture and Section 3.5 demonstrates the Full SFT training paradigm.

3.1 Preliminaries: Unified Vision-Language Modeling

Vision-Language Models (VLMs) unify perception and reasoning by optimizing an underlying joint parametric probability distribution $P_\theta(Y|X)$, where X represents the multimodal input sequence and Y denotes the corresponding generated token sequence. Modern architectures typically consist of a visual encoder $\mathcal{E}_\phi$, a learned projector $\mathcal{P}_\psi$, and a Large Language Model (LLM) backbone $\mathcal{D}_\omega$.

Qwen3-VL Backbone In this work, we build upon the sophisticated **Qwen3-VL** architecture (Bai et al., 2025a). A distinguishing feature of Qwen3-VL, critical to our framework, is its implementation of **Native Dynamic Resolution** based on the underlying NaViT (Naive Vision Transformer) paradigm (Dehghani et al., 2024). Unlike widely adopted traditional encoders (e.g., CLIP (Radford et al., 2021) or SigLIP (Zhai et al., 2023)) that compel input images into fixed-resolution squares (e.g., 224×224 or 336×336)—often causing severe aspect ratio distortion or loss of fine-grained detail—Qwen3-VL processes images as variable-length sequences of patches $V = \{p_1, \ldots, p_N\}$. This mechanism allows the model to ingest images of arbitrary aspect ratios and resolutions directly, preserving the native spatial geometry essential for precise physical grounding in robotic tasks. We leverage this capability to main-

tain high-fidelity visual representations of complex workspaces without artificial resizing artifacts.

3.2 Problem Formulation

We cast the long-horizon robotic planning task as a *visual-conditioned sequence generation* problem. Let $\mathcal{I}_0 \in \mathbb{R}^{H\times W\times 3}$ denote the high-resolution RGB observation at the initial timestep, and $\mathcal{L}_{task}$ be the natural language instruction describing the high-level goal (e.g., "Tidy up the desk"). Our objective is to learn a policy π_θ, parameterized by a Vision-Language Model (VLM), that maps these inputs to a coherent *Action Blueprint* $\mathcal{S}$:

$$\mathcal{S} = \{s_1, s_2, \dots, s_T\} = \pi_\theta(\mathcal{I}_0, \mathcal{L}_{task}) \quad (1)$$

where each s_i represents a semantically discrete sub-task in natural language (e.g., "1. Pick up the red block"). Unlike reactive policies (Kim et al., 2025) that predict low-level robotic control actions (a_t) in a greedy, step-by-step manner, LEAP leverages the full context window to maintain temporal logical consistency across the entire generated sequence. We hypothesize that through Full Parameter Supervised Fine-Tuning (Full SFT), the complex causal reasoning required for planning is effectively *internalized* into the model weights θ.

3.3 Data Construction

3.3.1 From Agibot-Beta to LEAP

To operationalize the formulation defined in Eq. (1), we require a dataset that pairs initial visual states with high-level logic. We leverage the Agibot-Beta dataset (Bu et al., 2025). Our pipeline (Figure 2) transforms raw teleoperation trajectories into the visual-conditioned planning format $\mathcal{S}$.

The raw data consists of task-level videos, global task names (e.g., "Prepare Breakfast"), and episode-level subtask descriptions. We process this data through a four-stage pipeline: (1) **Visual Extraction**: We extract the first frame of the video to serve as the visual condition $\mathcal{I}_0$. (2) **Intent Retrieval**: We retrieve the global task name $\mathcal{L}_{task}$. (3) **Blueprint Generation**: We extract the sequence of subtask names from the episode metadata to form the ground-truth *Action Blueprint* $\mathcal{S}$. (4) **Formatting**: These elements are assembled into a "First Frame + Prompt + Answer" tuple serialized in JSON format. The user prompt injects the task name into a standard query template, while the assistant's response consists of the ordered list of subtasks. This structured format serves as the concrete training target for our architecture.

3.4 LEAP Model Architecture

To effectively model the complex cross-modal mapping from $\mathcal{I}_0$ to the structured *Action Blueprint*, we employ the pre-trained Qwen3-VL-2B-Instruct architecture (Bai et al., 2025a). As illustrated in Figure 1, the system is composed of three tightly coupled modules: (1) a dynamic visual representation module that handles variable aspect ratios; (2) a cross-modal alignment adapter; and (3) a generative LLM backbone specialized for instruction following. By building upon this robust foundation, we focus our contribution on the *strategic alignment* of these components for long-horizon planning rather than architectural redesign.

3.4.1 Dynamic Visual Representation via Qwen3-VL

A critical limitation in standard prior VLA architectures, such as OpenVLA (Kim et al., 2025) or RT-2 (Zitkovich et al., 2023), is the reliance on rigid static resolution preprocessing (e.g., resizing inputs to 224×224 or 336×336). This operation inevitably introduces unwanted geometric distortion and high-frequency information loss, which are particularly detrimental for fine-grained robotic manipulation tasks involving small objects.

To address this, LEAP directly inherits the **Native Dynamic Resolution** mechanism from the Qwen3-VL backbone. Instead of naive resizing, the input image $\mathcal{I}_0$ is processed as a sequence of variable-length patches. Formally, the image is divided into patch triplets based on its native aspect ratio, effectively acting as a "Naive" Vision Transformer (NaViT) (Dehghani et al., 2024). This allows the model to process images of arbitrary resolution (e.g., 1024×768 or 600×600) without padding or distortion. By utilizing the pre-trained weights of Qwen3-VL, our visual encoder transforms $\mathcal{I}_0$ into a sequence of flattened visual tokens $V = \{v_1, v_2, \dots, v_K\}$, where the sequence length K varies dynamically with the input complexity. This feature is instrumental in preserving the spatial fidelity required for precise physical reasoning.

3.4.2 Multimodal Alignment Adapter

To bridge the modality gap between the visual manifold and the linguistic latent space, the visual tokens V are passed through a lightweight multi-layer perceptron (MLP) adapter. This adapter

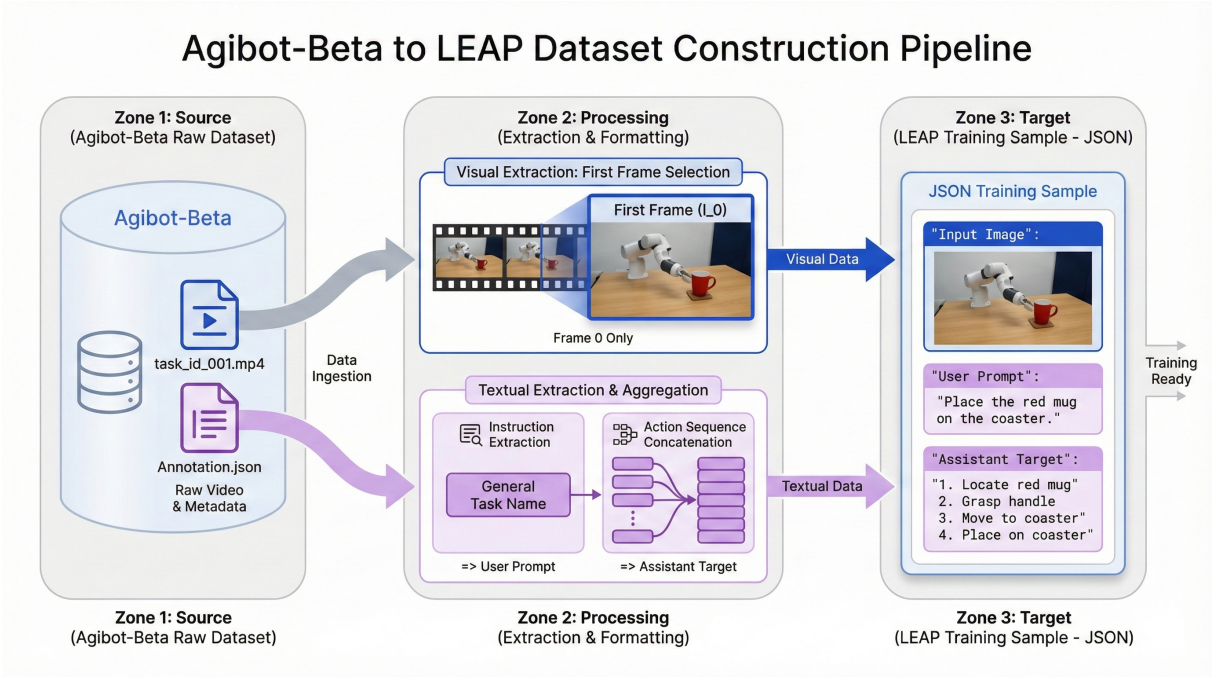

Figure 2: **Agibot-Beta to LEAP Data Construction Pipeline.** The pipeline efficiently transforms raw teleoperation episodes into structured instruction-tuning samples. Zone 1 (Source): Raw videos and associated metadata are ingested from the Agibot-Beta dataset. Zone 2 (Processing): Crucially, only the *First Frame* ($\mathcal{I}_0$) is extracted to serve as the static visual condition, forcing the model to plan long-horizon tasks based on the initial state. Simultaneously, discrete sub-tasks are retrieved and concatenated into a sequential *Action Blueprint*. Zone 3 (Target): These elements are formatted into a standardized JSON instruction pair to facilitate Full Parameter Supervised Fine-Tuning.

projects the high-dimensional visual features into the embedding space of the language model, resulting in a sequence of aligned visual embeddings E_v. Crucially, this projection layer is jointly optimized during our Full Parameter SFT phase (Section 3.5), ensuring that visual features are not merely recognized as semantic labels (e.g., "cup") but are grounded in their physical affordances (e.g., "graspable handle at coordinates (x, y)").

3.4.3 Generative Planning Backbone

The core reasoning engine of LEAP is the 2B-parameter decoder-only Large Language Model (LLM). Unlike models that freeze the backbone to preserve generic knowledge, we fine-tune the entire backbone to specialize in *embodied logical reasoning*. The LLM takes the concatenated sequence of aligned visual embeddings E_v and the tokenized task instruction T_{task} as input. through a causal self-attention mechanism, it autoregressively generates the *Action Blueprint* $\mathcal{S}$. The 2B parameter scale offers an optimal trade-off for embodied applications: it retains sufficient capacity for complex reasoning—demonstrating "emergent" planning capabilities—while remaining compact enough for efficient inference on edge devices.

3.5 Training Paradigm: Encoding Logic via Full SFT

3.5.1 Implicit vs. Explicit Reasoning

Recent advances, such as CoT-VLA (Zhao et al., 2025), advocate explicit Chain-of-Thought (CoT) reasoning. While effective, this paradigm incurs significant inference latency. In contrast, we propose to encode physical reasoning patterns directly into policy weights through Full Parameter Supervised Fine-Tuning (Full SFT). By updating all parameters $\theta = \{\theta_{vision}, \theta_{adapter}, \theta_{llm}\}$, LEAP aligns the visual encoder with physical reasoning tasks, allowing it to *implicitly* learn constraints without the overhead of test-time CoT generation.

3.5.2 Training Objective

We formulate the training of LEAP as a standard vision-language modeling task. The model acts as a probabilistic generator π_θ that maps inputs (observation $\mathcal{I}_0$ and instruction $\mathcal{L}_{task}$) to the target sequence $\mathcal{S}$ constructed in Section 3.3. We optimize the model using next-token prediction:

$$\mathcal{L}_{SFT} = -\sum_{t=1}^{|\mathcal{S}|} \log P(s_t | \mathcal{I}_0, \mathcal{L}_{task}, s_{<t}; \theta) \quad (2)$$

Notably, unlike approaches that discretize high-dimensional continuous robot actions into bounded bins, we employ the natural language action space defined by our data pipeline. This semantics-rich format preserves the pre-trained linguistic knowledge of the LLM backbone while providing a structured logical interface for downstream controllers.

4 Experiments

In Section 3, we formalized embodied robotic planning as a *visual-conditioned sequence generation* task, proposing LEAP to synthesize structured *Action Blueprints* directly from raw observations. Our central hypothesis states that by leveraging Full Parameter Supervised Fine-Tuning (Full SFT), a compact VLM can implicitly internalize the complex causal dependencies and physical constraints required for long-horizon planning, thereby obviating the need for explicitly verbose chain-of-thought reasoning. In this section, we empirically validate this premise. We conduct a rigorous evaluation on VLABench (Zhang et al., 2025) to assess the model's efficacy in grounding abstract natural language instructions into logically consistent, executable plans. Our analysis is structured to answer the following three primary research questions:

(RQ1) Performance vs. SOTA: How does LEAP perform when compared to state-of-the-art baselines, including significantly larger models (e.g., 8B+), across the diverse physical reasoning dimensions of VLABench? (Section 4.2)

(RQ2) Reasoning Acquisition: Does the proposed Full Parameter SFT paradigm effectively activate the model's latent logical reasoning capabilities, specifically in understanding physics laws and complex constraints? (Section 4.2)

(RQ3) Training Strategy: To what extent is Full Parameter SFT necessary for this embodied domain compared to parameter-efficient alternatives like LoRA, and how does it impact the stability of cross-modal feature alignment? (Section 4.3)

4.1 Experimental Setup

Benchmark We evaluate on VLABench (Zhang et al., 2025), a specialized and comprehensive suite assessing long-horizon embodied logical planning across six key performance dimensions: *Mesh & Texture* (M&T), *Semantic Understanding* (Sem), *Spatial Awareness* (Spa), *Physics Law* (Phy), *Complex Reasoning* (Cpx), and *Commonsense* (CS).

Baselines To rigorously evaluate the effectiveness of our proposed framework, we compare LEAP against a competitive suite of open-source Vision-Language Models. Specifically, we benchmark against: MiniCPM-V-2.6 (8B) (Hu et al., 2024a), Qwen2.5-VL-3B (Bai et al., 2025b), InternVL2.5-2B (Chen et al., 2024), and Qwen2-VL-2B (Wang et al., 2024). These models represent strong baselines at both similar (2B-3B) and larger (8B) parameter scales. All baselines are evaluated under a strict 0-shot, No-CoT inference setting to assess their inherent "intuitive" reasoning capabilities without the influence of prompt engineering.

Implementation Details LEAP is initialized from the pre-trained Qwen3-VL-2B-Instruct and fine-tuned on the Agibot dataset using Full SFT. We employ a response-only masking strategy, calculating the loss *only* on the tokens belonging to the *Action Blueprint*. Training utilizes the AdamW optimizer (LR 1.5e-5, weight decay 0.01) with a global batch size of 36. To balance performance and efficiency, the maximum sequence length is set to 4096 tokens. The visual encoder operates with native dynamic input resolution, typically using between 256 and 768 visual tokens per image.

4.2 Main Results

The quantitative performance is summarized in Table 1 and visualized in Figure 4.

Gains over Base Model Comparing LEAP (checkpoint-5000) to its original pre-trained initialization (Qwen3-VL-2B), we observe consistent notable improvements under the VLABench protocol. Specifically, LEAP achieves a **+26.3%** improvement in *Physics Law* (24.00 $\rightarrow$ 30.3) and **+37.7%** in *Complex Reasoning* (14.28 $\rightarrow$ 19.67), demonstrating that Full SFT effectively activates the model's latent reasoning potential within weights.

Comparison with SOTA Despite its compact size (2B), LEAP demonstrates superior efficacy in physical reasoning. As shown in Table 1, it achieves a score of **30.7** on the *Physics Law* dimension, significantly outperforming the 8B-parameter MiniCPM-V-2.6 (18.3). This result challenges the scaling law assumption, suggesting that for specialized embodied domains, high-quality instruction tuning can outweigh pure parameter count.

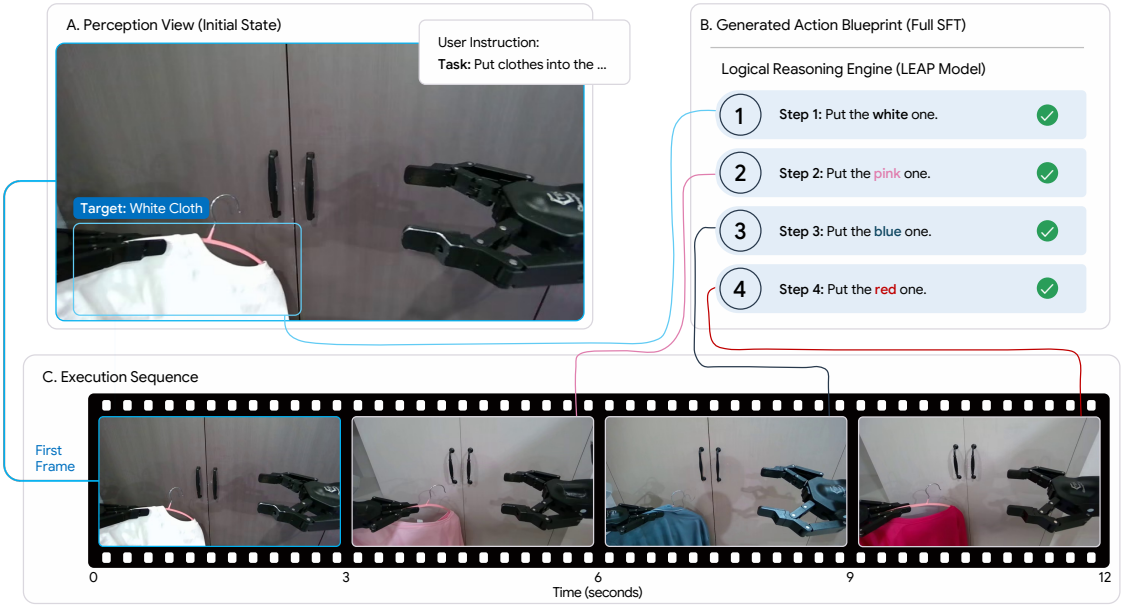

Figure 3: **Qualitative Visualization of the Perception-Reasoning-Execution Loop.** We illustrate a long-horizon "Tidy Up" task where the target (Blue Bowl) is obstructed by a magazine (Constraint). **(A) Perception View:** The autonomous robotic agent identifies objects and distinguishes targets (Royal Blue) from physical constraints (Purple) within the spatial layout. **(B) Logical Reasoning Engine:** LEAP generates a structured and hierarchical *Action Blueprint* that adheres to causal dependencies—prioritizing the removal of constraints (Steps 1-2) before interacting with the target (Steps 3-4). **(C) Execution Sequence:** The filmstrip demonstrates the successful physical execution, validating the effectiveness of vision-language grounding in complex and unstructured dynamic environments.

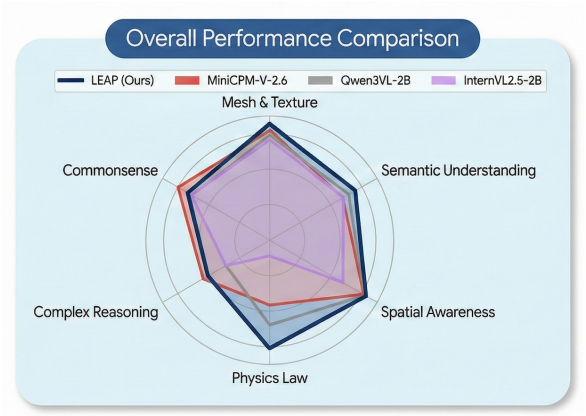

Figure 4: **VLABench Performance Profile.** LEAP (Royal Blue) demonstrates a balanced capability profile across various complex evaluation metrics with distinct advantages in the *Physics Law* dimension, outperforming larger baselines like MiniCPM-V-2.6 (Red).

4.3 Ablation Study: The Necessity of Full-Parameter Fine-Tuning

We conduct comprehensive ablation studies to assess the importance of Full SFT. We compare against Low-Rank Adaptation (LoRA) (Hu et al., 2022) under identical training conditions. As illustrated in Table 2 and Figure 5, Full SFT consistently yields higher scores in the *Physics Law* dimension. Although the final score gap at step 5000 is moderate (30.3 vs. 29.6), the training trajectory (Figure 5) reveals that Full SFT maintains a more stable and sustained improvement. This supports the hypothesis that updating the underlying full parameter space facilitates a deeper alignment between visual features and physical concepts and laws, whereas LoRA may be limited by its low-rank constraint when learning complex causal dependencies.

5 Conclusion

We presented LEAP, a unified 2B-parameter framework that enables robust long-horizon robotic planning by implicitly internalizing underlying physical reasoning via one-shot visual-conditioned sequence generation. This approach circumvents the latency of explicit Chain-of-Thought while achieving a state-of-the-art Physics Law score of 30.3 on VLABench, significantly outperforming larger 8B baselines. By generating structured *Action Blueprints*, LEAP bridges abstract goals and deterministic execution, offering a scalable and

Table 1: VLABench 0-shot evaluation results. We report comparative performance scores across six dimensions: Mesh & Texture (M&T), Semantic Understanding (Sem), Spatial Awareness (Spa), Physics Law (Phy), Complex Reasoning (Cpx), and Commonsense (CS). Higher values indicate better performance for all metrics.

Model	Params	Average	M&T	Sem	Spa	Phy	Cpx	CS
Baseline								
Qwen3VL-2B	2.2B	25.10	29.6	25.9	31.1	24.0	14.3	25.7
Open-Source SOTA								
MiniCPM-V-2.6	8B	25.96	31.0	24.1	30.6	18.3	**21.8**	**30.0**
Qwen2.5-VL-3B	3B	23.18	27.1	22.2	**32.5**	17.4	13.9	25.9
InternVL2.5-2B	2B	20.24	28.4	24.2	24.1	4.5	14.5	25.6
Qwen2-VL-2B	2B	17.37	23.0	22.2	23.8	2.4	11.3	21.5
LEAP (Ours)	2.2B	**28.38**	**32.8**	**28.5**	31.6	**30.7**	20.2	27.3

Table 2: Ablation study: Full SFT vs. LoRA across training steps. We focus on the stability and peak performance in the Physics Law (Phy) dimension. Full SFT exhibits stronger reasoning than LoRA baselines.

Method	Steps	Average	M&T	Sem	Spa	Phy	Cpx	CS
LoRA	2000	26.50	31.5	25.0	29.0	26.8	16.5	29.5
LoRA	3000	24.70	29.0	24.5	28.5	24.7	15.0	26.5
LoRA	4000	27.10	32.0	26.0	30.0	28.0	18.0	29.8
LoRA	5000	28.22	**33.2**	26.7	30.7	29.6	19.0	**30.1**
Full SFT	5000	**28.38**	33.0	**28.5**	**31.6**	**30.3**	**19.7**	27.3

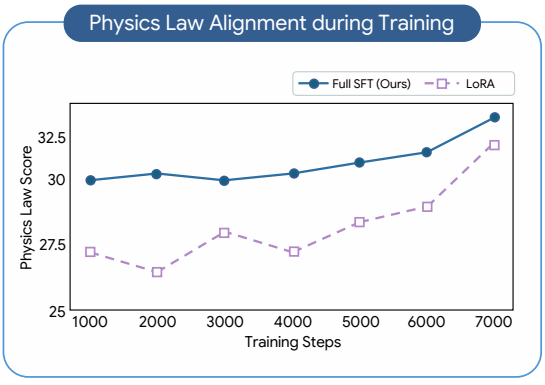

Figure 5: **Training Dynamics: Full SFT vs. LoRA.** The Full SFT trajectory (Blue) exhibits consistently higher performance on the *Physics Law* dimension compared to the parameter-efficient LoRA method (Purple), suggesting deeper underlying feature alignment.

interpretable alternative to hierarchical modular pipelines. Future work will explore integrating real-time sensory feedback to transition from open-loop planning to dynamic, closed-loop execution.

Limitations

Despite its effectiveness, our approach has limitations. **Open-Loop Nature:** LEAP generates the full *Action Blueprint* based solely on the initial observation ($\mathcal{I}_0$). It inherently lacks closed-loop feedback to adapt to intermediate failures or unpredictable environmental changes unless a re-planning mechanism is externally triggered. **Inference Latency:** Although significantly faster than CoT-based planners, the autoregressive generation of the 2B model is not entirely instantaneous. It is designed as a high-level planner rather than a high-frequency (e.g., $>$ 20Hz) low-level controller. **Scope of Ablation:** While we demonstrate the effectiveness of the overall system, we did not perform a granular ablation to decouple the gains attributed specifically to Qwen3-VL's native dynamic resolution mechanism versus the Full SFT strategy.

Ethics Statement

We utilize the Agibot-Beta dataset, which contains teleoperated demonstrations of safe, household-level tasks. There are no personally identifiable information (PII) or offensive content issues.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chapman, Yevgen Chebotar, Chelsea Finn, Chen Fu, Kanishka Gopalakrishnan, Karol Hausman, Alexander Herzog, Alex Irpan, Eric Jang, Karen Jeffrey, Dmitry Kalashnikov, Yuheng Kuang, Luis Leal, Sergey Levine, Yao Lu, Corey Lynch, and 10 others. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*. PMLR.

Jiaqi Bai, Shizhe Bai, Chenyang Cui, Sheng Dong, Tianyu Fu, Wen Huang, Chang Li, Xiaodong Li, Yankai Lin, Zhiyuan Liu, Liang Pan, Chen Qian, Yu Qiao, Shuhuai Ren, Maosong Sun, Jiaheng Wang, Lei Wang, Siyuan Wang, Yizhong Wang, and 10 others. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading and beyond. *arXiv preprint arXiv:2308.12966*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Sudeep Dasari, Chelsea Finn, Chen Fu, Kanishka Gopalakrishnan, Karol Hausman, Alexander Herzog, Alex Irpan, Julian Ibarz, Eric Jang, Karen Jeffrey, Arthur Juliani, Dmitry Kalashnikov, Yuheng Kuang, Luis Leal, Sergey Levine, and 12 others. 2023. Rt-1: Robotics transformer for real-world control at scale. In *Proceedings of the 6th Conference on Robot Learning*. PMLR.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, and Xindong He. 2025. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*.

Zhe Chen, Jiannan Wu, Wenhai Wang, and 1 others. 2024. Internvl 2.5: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*.

Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. 2024. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. In *International Conference on Learning Representations*.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey Gritsenko, Mario Lucic, and Neil Houlsby. 2024. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In *Advances in Neural Information Processing Systems*.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. 2024. Video language planning. In *Proceedings of the International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenhui Hu, Lei Li, Wei Zhang, Jing Hou, Linjia Sun, Xiaodong Liu, Jing Zhang, Qian Zhao, Li Dong, Furu Wei, and Houqiang Wang. 2024a. Minicpm-v: A gpt-4v level mllm with native multimodal tokens. *arXiv preprint arXiv:2408.01800*.

Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. 2024b. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. In *ICRA 2024 Workshop on Vision-Language Models for Navigation and Manipulation*.

Wenlong Huang, Wang Chen, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. 2023. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Proceedings of the 7th Conference on Robot Learning*.

Donghyun Kim, Harish Kannan, Pierre Sermanet, Pete Florence, Ayzaan Wahid, Huazhe Xu, and Andy Zeng. 2025. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*. PMLR.

Jan Mewes, Mirko Wächter, and Tamim Asfour. 2025. Vilain: Visual language planning with integrated tamp. *arXiv preprint arXiv:2506.03270*.

OpenAI, Josh Achiam, Sasha Adler, Sandhini Agarwal, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alec Radford, Jong Wook Kim, Chris Hallacy Xu, Greg Brockman, Tom McCarthy, Ilya Sutskever, Aditya Ramesh, Lior Yacoby, Nick Bhupatiraju, David Krueger, Mark Chen, and Jonathon Shlens. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR.

André Schakkal, Ben Zandonati, Zhutian Yang, and Navid Azizan. 2025. Hierarchical vision-language planning for multi-step humanoid manipulation. In *RSS 2025 Workshop on Robot Planning in the Era of Foundation Models*.

Pierre Sermanet, Tianli Ding, Jeffrey Zhao, and Others. 2024. Robovqa: Multimodal long-horizon reasoning for robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Language-driven task and motion planning for manipulation in open worlds. In *Proceedings of Robotics: Science and Systems*. Robotics: Science and Systems Foundation.

Boyang Wang, Chen Feng, Zhou Xian, Yuying Ge, Yixuan Liu, and Chuang Gan. 2025. This&that: Language-gesture controlled video generation for robot planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Peng Wang, Shuai Bai, Sinan Tan, Shiji Song, Gao Huang, Junyang Lin, Chang Zhou, Hongxia Yang, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. In *Advances in Neural Information Processing Systems*.

Yuhang Xiong, Limin Xie, Yi Ru, and Haibo Yi. 2024. Relep: A novel framework for real-world long-horizon embodied planning. *arXiv preprint arXiv:2409.15658*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, Lucas Beyer, Joan Puigcerver, Carlos Riquelme, Matthias Minderer, Hugo Touvron, Alexey Dosovitskiy, and Neil Houlsby. 2023. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2025. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Qingqing Zhao, Yao Lu, Adil Zouitine, Michaels H. B., Pierre Sermanet, Huong Tran, Chelsea Finn, Karol Hausman, De-An Huang, Zhiting Hu, Sergey Levine, Ted Xiao, and Karl Pertsch. 2025. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Rui Zhou, Xin Zhao, Hao Wang, Jing Xu, and Wei Zhang. 2024. Vision-language-policy model for dynamic robot task planning. *arXiv preprint arXiv:2412.17646*.

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, and 35 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

A VLABench Evaluation Pipeline

We provide implementation details of our VLABench evaluation procedure to ensure reproducibility.

Output Schema. LEAP outputs are structured as JSON arrays wrapped in markdown code blocks. The expected format is shown below:

```
[
  {"name": "pick", "params": {"
      target_entity_name": "1"}},
  {"name": "pour", "params": {"
      target_container_name": "0"}},
  {"name": "place", "params": {"
      target_container_name": "0"}}
]
```

Each element represents a primitive skill (e.g., `pick`, `pour`, `place`) with optional target entity or container parameters.

Decoding Configuration. All evaluations use greedy decoding (`do_sample=False`) with `max_new_tokens=512` to produce deterministic outputs.

Scoring Function. We use the official VLABench scoring function `get_final_score()` from `VLABench/evaluation/utils.py`, which computes a weighted combination of four metrics:

- **Skill Match Score** (weight=0.4): Unordered skill name matching
- **Entity Match Score** (weight=0.4): Unordered entity/container recognition
- **Skill-with-Entity Score** (weight=0.1): Joint skill-entity matching
- **Exact Match Score** (weight=0.1): Graph-based subtask DAG exact match

The `total_score` is computed as:

$$\text{total} = 0.4 \cdot \text{skill} + 0.4 \cdot \text{entity} + 0.1 \cdot \text{joint} + 0.1 \cdot \text{exact} \quad (3)$$

Training vs. Evaluation Format. During training, LEAP learns from natural language step sequences (e.g., "1) Pick up X. 2) Place X on Y."). For VLABench evaluation, we use the standard prompt template, which instructs the model to produce a JSON skill-list. We extract JSON from the model output; samples where JSON parsing fails are scored as 0.

Unified Evaluation Protocol. All models (LEAP and baselines) follow the same VLABench evaluation protocol, using the official prompt template and scoring function. We use greedy decoding (`do_sample=False`, `max_new_tokens=512`) for all evaluations. Samples where parsing fails are assigned a score of 0.

B LoRA Fine-Tuning Configuration

For the ablation study (Section 4.3), we use the PEFT library's LoRA implementation. The "standard" preset configuration is shown in Table 3.

Table 3: LoRA "Standard" Preset Configuration

Parameter	Value
LoRA Rank (r)	16
LoRA Alpha (α)	32
LoRA Dropout	0.05
Target Modules	q/k/v/o_proj
Bias	None
Task Type	Causal LM
Learning Rate	2×10^{-4}
Warmup Steps	800
Weight Decay	0.01
Effective Batch Size	36 (4×9 grad. accum.)
Max Sequence Length	4096 tokens
Precision	BFloat16
Max Training Steps	7000

Both LoRA and Full SFT share identical training data (151,230 samples from Agibot-Beta), evaluation protocol (0-shot, No-CoT), and decoding configuration (greedy). The only difference is the parameter update strategy: LoRA updates ∼0.5% of parameters while Full SFT updates all 2.2B parameters.

C Training Data Statistics

Table 4 summarizes the training data characteristics derived from the Agibot-Beta dataset.

Checkpoint Selection. We report results from checkpoint-5000, selected based on validation performance across all six VLABench dimensions. This corresponds to approximately 1.2 epochs of training on the full dataset.

Table 4: Training Data Statistics

Statistic	Value
Total Training Samples	151,230
Training File	JSONL Dataset
Visual Tokens per Image	256–768 (dynamic)
Max Sequence Length	4,096 tokens

D Full SFT Training Dynamics

Table 5 presents the VLABench evaluation scores at every 200 training steps for the Full SFT model, providing a fine-grained view of the training dynamics.

Table 5: Full SFT VLABench Scores at Every 200 Steps

Step	Avg	M&T	Sem	Spa	Phy	Cpx	CS
200	22.3	26.1	21.5	27.2	18.4	12.8	22.1
400	23.8	27.5	22.8	28.1	19.7	13.5	23.8
600	24.5	28.2	23.1	28.9	20.8	14.2	24.5
800	25.1	29.0	23.8	29.4	22.1	14.8	25.2
1000	25.8	29.6	24.3	29.8	23.5	15.4	25.9
2000	26.8	30.5	25.6	30.2	26.2	17.1	26.4
3000	27.4	31.2	26.4	30.8	27.8	18.2	26.8
4000	27.9	32.0	27.2	31.2	29.1	19.0	27.1
5000	**28.4**	**32.8**	**28.5**	31.6	**30.3**	20.2	27.3
6000	28.1	32.4	27.8	31.9	29.2	20.5	27.8
7000	27.8	32.6	28.1	31.5	30.7	19.8	27.5

Note: Bold = best; underline = 2nd best. M&T=Mesh & Texture, Sem=Semantic Understanding, Spa=Spatial Awareness, Phy=Physics Law, Cpx=Complex Reasoning, CS=Commonsense.

The results show performance across training steps. Checkpoint-5000 achieves the best average performance.

E Training-Evaluation Task Disjointness

To ensure fair evaluation, we verify the disjointness between training and evaluation task sets.

Methodology. We extracted 184 unique task names from the Agibot-Beta training corpus and 45 unique task identifiers from the VLABench evaluation suite.

Results. No exact string overlap was found between the two task sets, confirming that the model is evaluated on previously unseen task descriptions.