Plug-and-Think: Structured Reasoning for Vision–Language–Action Models

Kaikai Wei[1,*] Di Wen[2,*] Xinhai Li[1] Senwei Xiang[1,†]

[1]Hangzhou International Innovation Institute of Beihang University, China
[2]Beijing Jiaotong University, China

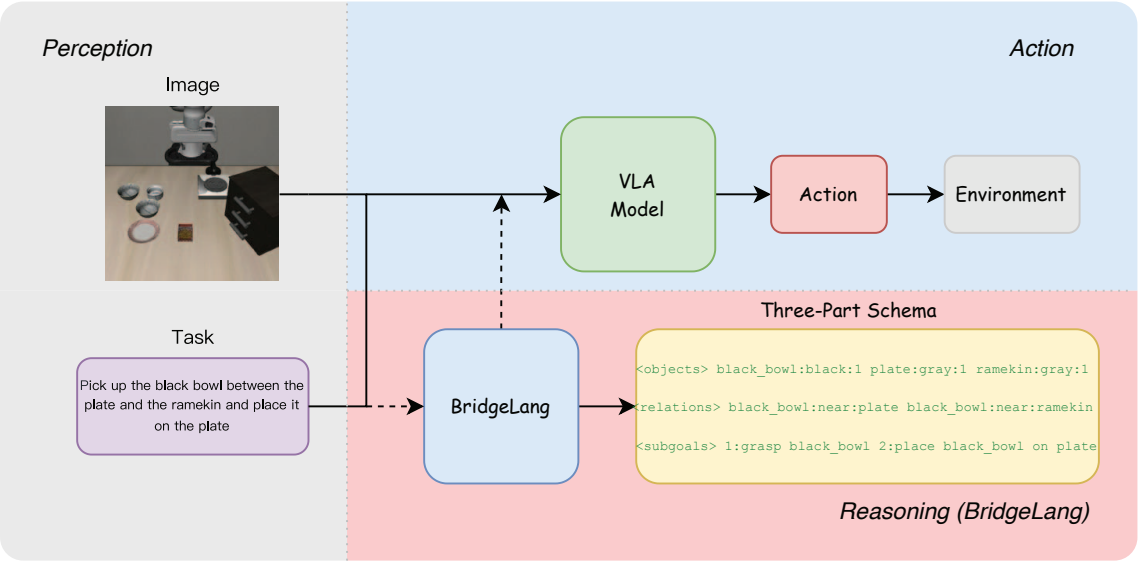


Figure 1. Conceptual overview of **BridgeLang**, a lightweight and pluggable visual-language supplementor that acts as an external "plug-and-think" layer bridging perception and action. Given an image and a task instruction, BridgeLang generates a structured three-part schema (`<objects>`, `<relations>`, and `<subgoals>`), from which subgoals are extracted and appended to the original instruction. This produces an augmented prompt that provides downstream VLA models with explicit reasoning, leading to more coherent and successful action execution.

Abstract

Vision–Language–Action (VLA) systems often fail on out-of-distribution tasks and lack interpretable reasoning, requiring costly retraining to adapt. We address these limitations by proposing ***BridgeLang****, a lightweight external reasoning supplementor that introduces a "Plug-and-Think" paradigm to enhance unmodified, pre-trained VLA models. BridgeLang is an efficient visual language model trained on our new* ***Bridge-CoT dataset*** *using a prompt-based instruction-finetuning strategy. This preserves its general abilities while teaching it to act as a scene-aware planner. Given an initial observation and task, BridgeLang performs hierarchical reasoning—internally identifying* `<objects>` *and* `<relations>` *as a scaffold to generate a high-quality, executable* `<subgoals>` *plan. This "think-before-act" process occurs once, after which only the semantically cleaned subgoals string is concatenated with the original instruction. When integrated with OpenVLA, BridgeLang improves average success rates on the LIBERO benchmark by* ***+5.45%*** *(up to +8.2%) without any VLA retraining and at the cost of only a small, one-time pre-execution latency. Our work demonstrates the efficacy of decoupled, scaffolded reasoning and introduces the Bridge-CoT dataset to facilitate structured multimodal planning. The dataset and code are available at* `https://github.com/CliffKai/BridgeLang`.

1. Introduction

Recent advances in vision-language-action (VLA) systems have made them a central framework for embodied intelligence [1, 6]. By jointly modeling visual perception, natural language instructions, and motor control, these models enable robots to perform complex tasks in realistic en-

*Equal contribution.
†Corresponding author: `xiangsw@buaa.edu.cn`

vironments. Models such as RT series [5, 45] and OpenVLA [11, 12] have achieved remarkable progress through large-scale training and multimodal representation learning [6, 10, 27]. However, despite this progress, existing VLA systems still face key limitations in generalization and reasoning.

In control-oriented scenarios, large-scale robot data training often weakens their original vision-language understanding ability. Moreover, their performance heavily depends on the training distribution, leading to dramatic degradation when deployed in out-of-distribution environments [3, 4, 21, 23, 26]. Re-training or fine-tuning VLA models for each new task is prohibitively expensive and contradicts the goal of general-purpose embodied agents. More critically, current VLA models lack explicit reasoning: they operate as black boxes without interpretable thought processes. Although several works attempt to incorporate Chain-of-Thought (CoT) reasoning into VLA systems, such as CoT-VLA [41, 43], these methods require generating full reasoning chains step by step, resulting in up to 7× higher inference latency—making them impractical for real-world deployment.

This poses a critical challenge [2, 39]: How can we endow existing VLA models with interpretable CoT-style reasoning and stronger generalization without costly VLA retraining or per-step inference latency? The key lies in designing a lightweight, external reasoning mechanism that preserves end-to-end efficiency while enhancing vision-language-action understanding.

To address this challenge, we propose **BridgeLang**—a lightweight, plug-and-play Visual-Language Supplementor (Fig. 1). BridgeLang introduces the principle of "Plug-and-Think" [7], enabling explicit Chain-of-Thought reasoning before task execution. Crucially, our analysis reveals a fundamental fragility in existing VLA systems: while they are adept at following natural language, they are catastrophically sensitive to out-of-distribution (OOD) structured inputs. As demonstrated in our experiments (Tab. 3), directly injecting the full, tagged reasoning chain (e.g. with `<objects>` tags) causes a total performance collapse to 0.0% success. Therefore, BridgeLang's core mechanism is designed to resolve this tension: it performs structured, hierarchical reasoning internally, but then translates this reasoning into a clean, natural-language `<subgoals>` plan. Only this semantically-cleaned subgoals string is then concatenated with the original task instruction, allowing the VLA model to *think before acting* [30, 41] without modifying its architecture or encountering OOD symbolic noise.

Thanks to its fully modular design that interfaces via language enhancement, BridgeLang is in principle compatible with other VLA systems (*e.g.* OpenVLA, RT series) without retraining or fine-tuning the VLA model itself. This "external reasoning injection" alleviates the degradation of visual-language ability in control tasks and significantly improves generalization across environments. Importantly, our study demonstrates the crucial role of CoT in VLA reasoning—particularly when the task language differs from natural language—showing that joint vision–language reasoning greatly enhances decision consistency and interpretability.

In addition, we build **Bridge-CoT**, a new multimodal reasoning dataset derived from BridgeData V2 [35], formatted as *Image + Task + Structured CoT text (objects, relations, subgoals)*. This dataset provides the structured CoT supervision used to train BridgeLang, which is obtained through an efficient prompt-based instruction-finetuning strategy on the Bridge-CoT dataset. This approach trains the 3B model to associate our specific reasoning prompt with the structured output format, thereby preserving its general-purpose VLM capabilities while learning from the teacher's (Qwen2.5-VL-72B) structured CoT outputs [14, 15, 37]. This process can be viewed as a form of offline data-level distillation, transferring the reasoning capability of large models into a lightweight supplementor.

It is worth noting that while our work emphasizes methodological consistency and practical feasibility, its central objective is not merely to introduce a new paradigm for VLA enhancement, but to systematically verify whether structured Chain-of-Thought (CoT) reasoning can substantially improve task execution and generalization [29, 43]. Furthermore, the proposed Bridge-CoT dataset provides a standardized foundation for structured reasoning supervision, fostering reproducibility and methodological rigor in multimodal reasoning research [1, 35].

BridgeLang offers four major advantages: (1) *Zero-cost transferability* – no VLA retraining required; (2) *Plug-and-play non-intrusiveness* – no architectural changes; (3) *High efficiency* – minimal amortized latency (one-time pre-execution cost); (4) *Reasoning enhancement* – explicit CoT-style thinking before execution. Comprehensive experiments on LIBERO benchmarks [19] validate BridgeLang's universality and performance gains, providing a new modular paradigm for vision–language–action reasoning.

Despite having no exposure to LIBERO data, BridgeLang improves the success rates of released OpenVLA models by **2.6–8.2%** across different tasks, without any VLA retraining or architectural modification, and with only a small, one-time additional inference latency.

2. Related Work

2.1. End-to-End Vision-Language-Action Systems

End-to-end Vision-Language-Action (VLA) models have become a dominant paradigm in embodied intelligence, aiming to learn a single policy that maps visual observations and language instructions directly to low-level actions. Seminal works like RT-1 [5] and VIMA [10] demonstrated the scalability of Transformer-based architectures for visuomotor

control. This line of research has rapidly evolved towards large-scale, generalist agents trained on massive, diverse datasets [23, 24]. Models like RT-2 [45] and Octo [34] leverage the pre-trained representations of large vision-language models, treating action generation as a sequence modeling problem.However, a key challenge in these end-to-end systems is the "black box" nature of their decision-making. While powerful on in-distribution tasks, their generalization can be limited [17]. More critically, as these models are heavily fine-tuned for control, their latent language understanding capabilities can become brittle. As our work demonstrates (Section Sec. 3.3), this brittleness makes them catastrophically sensitive to out-of-distribution (OOD) structured inputs, creating a critical barrier to integration with explicit reasoning mechanisms. Our work, BridgeLang, is designed to augment these unmodified, pre-trained VLA models (*e.g*. OpenVLA [11]) by providing a compatible, natural-language reasoning supplement.

2.2. LLM and VLM as Planners in Embodied AI

A parallel paradigm decouples high-level reasoning from low-level control, using Large Language Models (LLMs) or Vision-Language Models (VLMs) as task planners. SayCan [1] was a pioneering work in this area, using a non-visual LLM to generate high-level plans, which were then grounded in the physical world via a separate affordance function. This inspired a range of works where LLMs generate programmatic plans, such as code-as-policies [18] or Python-like scripts [31].More recent works, like PALM-E [6], integrate visual information directly into the planner, creating true VLM-based planners. These systems often generate textual plans or internal monologues to guide behavior [8]. However, these approaches typically assume they are the primary planning system. They do not address how to enhance an existing, pre-trained VLA model that cannot be easily modified. BridgeLang fills this gap, acting as a "translation layer" that allows a structured VLM planner (BridgeLang) to communicate its reasoning to a brittle end-to-end VLA policy (OpenVLA) without causing the 0.0% failure mode we identified.

2.3. Chain-of-Thought in Embodied Reasoning

The concept of Chain-of-Thought (CoT) reasoning, first introduced in NLP [36], has shown significant promise for improving the interpretability and robustness of complex reasoning tasks. In robotics, CoT-VLA [43] incorporates this idea by prompting the VLA model to generate step-by-step reasoning during task execution. While this enhances transparency, it incurs a substantial, per-step (run-time) inference latency, making it impractical for real-world deployment.Our approach diverges from this "online" reasoning paradigm. BridgeLang performs CoT reasoning once as a pre-execution step, generating a complete, structured plan (Objects $\rightarrow$ Relations $\rightarrow$ Subgoals) before the first action is taken. This "think-before-act" strategy provides the benefits of explicit reasoning with near-zero amortized latency. Furthermore, while early works used linguistic inputs to modulate visual features, such as with FiLM [25], our method provides explicit, high-level symbolic planning rather than feature-level modulation.

2.4. Modular Reasoning and Structured Datasets

The "Plug-and-Think" paradigm of BridgeLang draws inspiration from a long line of work on modular reasoning. In VQA, early works like Neural Module Networks (NMN) [2] and NS-VQA [40] proposed composing specialized modules to answer complex questions. This modular spirit is also seen in NLP with tools and adapters [28, 32], and in the design of modern VLMs that build upon separate visual and language backbones [9, 13, 16, 33]. BridgeLang extends this by proposing a lightweight, external reasoning module for VLA.A critical component of our work is the Bridge-CoT dataset, which provides the necessary supervision. Existing multimodal reasoning datasets [20, 29, 38] typically offer free-form textual annotations. While useful for generating human-readable explanations, this "unstructured" CoT is difficult to parse and poorly aligned with the action space required for robotic planning. Our Bridge-CoT dataset fills this data-level gap by providing machine-parsable, hierarchical, and action-oriented structured supervision, enabling the training of robust, specialized planners like BridgeLang.

3. Method

3.1. Overview

BridgeLang is designed to provide existing Vision–Language–Action (VLA) systems with an externalized, plug-and-play structured reasoning layer, enhancing task understanding and generalization without retraining or modifying the target VLA model. As shown in Fig. 1, BridgeLang serves as a *Structured Reasoning Supplementor* that, prior to task execution, receives visual observations and language instructions and produces a structured supplementary text composed of Objects / Relations / Subgoals.

This module performs inference independently outside the VLA system and supplies its planning output as supplementary input to the original model, thereby introducing explicit task decomposition and environment understanding into the action generation stage. This mechanism requires no architectural modification or retraining of the target VLA, achieving a lightweight "Plug-and-Think" enhancement solely through pre-execution reasoning. In practical deployments, BridgeLang can be directly integrated into various open-source VLA models (*e.g*. OpenVLA), enabling zero-cost transfer and efficient reasoning enhancement.

3.2. Structured Reasoning Supplementor

The core functionality of BridgeLang is to generate, prior to task execution and based on visual observations and lan-

guage instructions, a parsable, structured reasoning supplement $S = \{O, R, G\}$. As established in our integration analysis (Sec. 3.3), the <objects> (O) and <relations> (R) components serve as an internal reasoning scaffold for BridgeLang itself, ensuring the semantic grounding and quality of the final <subgoals> (G) plan. Only the extracted and cleaned Subgoals (G') are then appended as an external input to the original task instruction, thereby guiding the model toward more logical and interpretable decisions without modifying or retraining the VLA model.

BridgeLang is implemented on a multimodal language model (Qwen2.5-VL-3B). The inputs are the visual observation I and the language task instruction T, and the output is a structured reasoning supplement $S = \{O, R, G\}$, where O, R, and G denote the three components Objects, Relations, and Subgoals, respectively. The entire generation process is single-pass inference, requiring neither recursive generation nor multi-step decoding.

Inspired by research on structured Chain-of-Thought in multimodal reasoning [22, 42, 44], we adopt a three-stage structure—Objects → Relations → Subgoals—to emulate the hierarchical human process of "recognition–understanding–planning":

(1) **Objects:** Identify key entities and their attributes involved in manipulation, providing the basis for vision–language grounding;
(2) **Relations:** Model spatial or functional relationships among objects, establishing the semantic context of the scene;
(3) **Subgoals:** Produce executable stepwise action plans grounded in the first two components, forming high-level task planning.

This hierarchical "recognition – understanding – planning" paradigm explicitly separates perception from planning, thereby maintaining semantic consistency and logical constraints during generation and significantly improving success rates and interpretability at execution time.

BridgeLang employs a strictly constrained, templated prompt to ensure output structural stability. The system instruction explicitly restricts the output to three fields (<objects>, <relations>, <subgoals>) and prohibits free-form natural language descriptions. For example:

```
<objects> black_bowl:black:1 plate:gray:1 ramekin
    :gray:1 </objects>
<relations> black_bowl:near:plate black_bowl:near
    :ramekin </relations>
<subgoals> 1:grasp black_bowl 2:place black_bowl
    on plate </subgoals>
```

To ensure the semantic quality of the supplement, we introduce semantic post-processing, including rule-based normalization and semantic deduplication, which are detailed in the supplementary material.

The cleaned Subgoals are converted into a natural-language sequence and can be directly concatenated to the original task instruction to form an enhanced input. The entire reasoning process requires only a single forward pass, maintaining very low computational overhead.

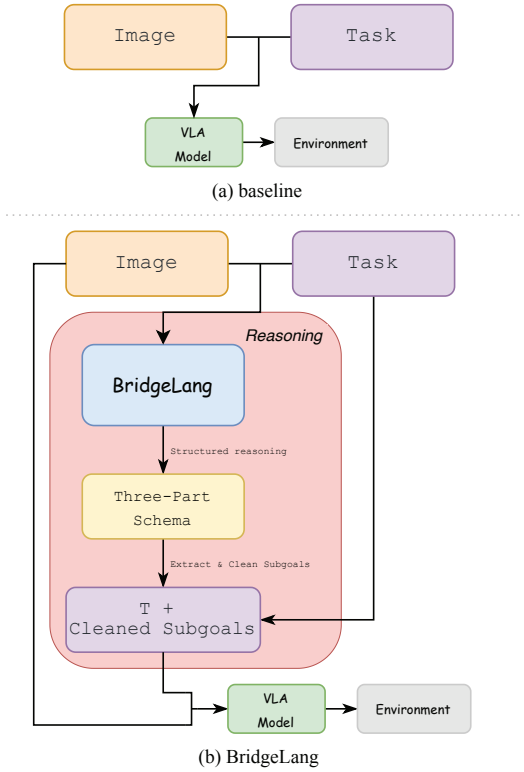


Figure 2. Comparison between standard VLA and BridgeLang-enhanced pipelines. (a) Baseline: direct mapping from vision and instruction to actions. (b) BridgeLang: adds a structured reasoning layer producing explicit subgoals for interpretable "think-before-act" control.

Through the structured template, BridgeLang explicitly distinguishes the three stages of "recognition–understanding–planning" at the generation level, providing VLA models with a logically structured action plan and achieving a lightweight reasoning enhancement of "think-before-act."

3.3. Integration with VLA Models

In a standard VLA framework, the model takes visual observations I and task instructions T as input and outputs an action sequence. Serving as a pre-reasoning layer, BridgeLang generates structured Subgoals prior to VLA execution and fuses them with the original task instruction to form an enhanced input, thereby explicitly introducing reasoning logic into the action-generation stage and realizing a "think-before-act" control paradigm (see Fig. 2).

To systematically analyze the impact of different integration strategies, we compare three fusion modes: Raw Structured Mode: directly input the complete three-part structured text; Uncleaned Subgoal Mode: input the original numbered Subgoals; Cleaned Subgoal Mode (default): input semantically cleaned natural-language Subgoals. Experiments show that the Cleaned Subgoal mode achieves the best performance in reasoning consistency and task success rate, primarily because its natural-language input better aligns with the VLA's language embedding space and avoids interference from symbolic noise. Therefore, we adopt this mode in all experiments.

The concatenation template for cleaned Subgoals is as follows:

```
<task>? <subgoals>.
```

Example:

```
Pick up the black bowl between the plate and the
    ramekin and place it on the plate? grasp
    black_bowl; place black_bowl on plate.
```

This language-level fusion fully leverages the VLA's language understanding capability, achieving interpretability enhancement under a unified interface.

At the system level, BridgeLang runs as a lightweight inference service, independently deployed on a single GPU, and can provide supplementary reasoning in parallel for multiple VLA models. Inference is executed only once before the task starts and thus does not add any per-step (run-time) control latency. This "once-think, act-throughout" design ensures system independence and cross-model reusability, providing an efficient and scalable augmentation paradigm for multimodal embodied intelligence.

3.4. Bridge-CoT Dataset and Training

Dataset Construction. Building upon BridgeData V2 [35], we construct the **Bridge-CoT** dataset. BridgeData V2 contains 53,192 episodes, each consisting of 4 images and a natural-language task instruction (totaling 212,768 images). For each episode, we randomly sample 1 image and pair it with its instruction (I, T), then use Qwen2.5-VL-72B (teacher) to generate a matching three-part structured reasoning text $S =$ <objects>, <relations>, <subgoals>. During generation, each sample is subjected to strict structural consistency checks (field closure and signature format). After filtering empty instructions and anomalous samples, we retain 38,660 (I, T, S) samples for training.

We perform two types of quality control: (1) *regular-expression checks* for field-format and action-vocabulary consistency; (2) *random manual inspection* of 5% of the samples (over 1,900 pairs). This rigorous check was performed to validate higher-level properties beyond regex, such as semantic correctness (*e.g.* ensuring objects in the plan exist in the image) and plan validity. Within this robust subset, we found no observable errors, indicating an exceptionally high quality of distillation from the teacher model.

Training. To preserve the model's general VLM capabilities while teaching it the specialist task of structured planning, we employ a prompt-based instruction-finetuning strategy. Instead of standard finetuning, each (I, T, S) sample from Bridge-CoT is formatted using the strict instructional prompt (denoted as P_{tpl}), which is detailed in our supplementary material (and code).The model is then trained only to generate the structured supplement S when conditioned on this specific prompt, in addition to the image I and task T. The objective is therefore modified to condition on this prompt:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, I, T, P_{tpl}). \tag{1}$$

This approach isolates the finetuning task to the specific prompt, preventing catastrophic forgetting of the model's general-purpose skills. Detailed hyperparameters and training specifics are provided in the supplementary material.

Discussion. Bridge-CoT provides standardized, machine-parsable structured CoT supervision, which offers better hierarchical consistency and reproducibility than free-form CoT. It also verifies the effectiveness of an offline structured distillation paradigm from large models to small models for multimodal reasoning.

3.5. Design Rationale and Ablation Guidance

Structured and Hierarchical CoT Design. While free-form CoT exhibits certain reasoning capabilities, it often suffers in multimodal settings from semantic redundancy, structural instability, and inconsistent entity alignment. To address these issues, BridgeLang adopts a hierarchical structured reasoning mechanism that explicitly partitions the process into Objects, Relations, and Subgoals, corresponding to the human cognitive stages of "recognition–understanding–planning." This structured design not only improves syntactic stability and semantic controllability of the outputs but also provides a formalized foundation for data annotation and model optimization.

Externalization of the Reasoning Layer. Unlike joint-trained internal reasoning, BridgeLang explicitly externalizes reasoning as an independent module, thereby decoupling reasoning from control. This externalization offers two advantages: (i) enabling zero-cost transfer across different VLA architectures; (ii) preserving the separation between reasoning and execution, which enhances generalization across tasks and environments.

Key Hypotheses. We summarize three core hypotheses to be validated in Sec. 4: **H1.**Performance gains primarily stem from semantically consistent *Subgoals*, rather than mere string concatenation; **H2.**Within the three-part schema, *Subgoals* contribute most critically to performance;

H3. Semantic cleaning significantly improves linguistic consistency and action alignment.

4. Experiments

4.1. Experimental Setup

We evaluate BridgeLang on the LIBERO benchmark [19]. LIBERO contains four suites—`libero_spatial`, `libero_object`, `libero_goal`, and `libero_10`. Each suite consists of 10 tasks, and each task is evaluated with 50 independent episodes to compute the Success Rate.

Baseline Model. We adopt the open-source OpenVLA [11] as our baseline. To ensure fairness and reproducibility, we do not introduce additional models or retraining; only the released OpenVLA checkpoints are used for evaluation.

Supplementor Model. BridgeLang is implemented with Qwen2.5-VL-3B and trained via structured distillation from Bridge-CoT annotations generated by Qwen2.5-VL-72B, resulting in a lightweight external structured reasoning supplementor.

Metrics. We report task Success Rate (%) as the primary metric, computed as the average success over 50 episodes per task.

Implementation Details. Our evaluation protocol, hardware configuration (*e.g*. NVIDIA RTX 3090s), random seeds, and full implementation details are deferred to the supplementary material to conserve space. BridgeLang runs once *before* VLA execution to generate the structured supplement; thus, it does not affect real-time control latency and adds only a negligible pre-execution overhead relative to standard VLA inference.

Evaluation Protocol. In all experiments, BridgeLang outputs are pre-generated once before task execution, *i.e*. the structured text is produced once and concatenated with the task instruction before being fed to OpenVLA. The entire evaluation is fully automated to ensure fairness and reproducibility.

4.2. Overall Performance

Table 1 summarizes the overall results across LIBERO suites. Under the same environment (NVIDIA RTX 3090, 24 GB), OpenVLA achieves an average Success Rate of 72.1%, while OpenVLA + BridgeLang reaches 77.55%. Without any retraining on LIBERO and without accessing LIBERO data, BridgeLang yields absolute gains of 2.6%–8.2%, with an average improvement of 5.45%.

Table 1. Performance comparison between baseline OpenVLA and BridgeLang-enhanced OpenVLA on LIBERO (Success Rate, %).

Model	spatial	object	goal	10	Avg
OpenVLA	84.6	71.2	76.4	56.2	72.1
OpenVLA + BridgeLang	**90.4**	**79.4**	**81.6**	**58.8**	**77.55**

The gains are consistent across all suites and are most pronounced on `libero_object` and `libero_goal` (improvements of 8.2% and 5.2%, respectively). This indicates that BridgeLang's structured reasoning (Objects / Relations / Subgoals) strengthens semantic grounding and action consistency, especially for tasks involving object-centric relations and goal reasoning.

Overall, BridgeLang injects explicit reasoning into existing VLA systems with *no re-training, no architectural changes, and near-zero latency*, providing an efficient and interpretable performance boost. This validates the effectiveness and generality of the proposed External Structured Reasoning Layer.

4.3. Ablation and Analysis

We conduct ablations to validate key design choices and identify the source of gains: (1) whether improvements arise from structured semantic reasoning rather than additional text; (2) contributions of different structured components; and (3) effects of component weighting during distillation.

(a) Semantic Effectiveness of Structured Subgoals

Question. Do the improvements come from genuine semantic reasoning instead of text length or language activation?

Experiment. We compare the true Cleaned Subgoals against two length-matched but non-semantic controls:

1. *Cleaned Subgoals* (the actual structured reasoning output);
2. *Random Strings* (length-matched random character sequences);
3. *Common-action Strings* (length-matched strings containing common action tokens such as *take*, *place*, but unrelated to the task).

Table 2. Semantic effectiveness of structured subgoals.

Input Type	spatial	object	goal	10	Avg
Cleaned Subgoals	**90.4**	**79.4**	**81.6**	**58.8**	**77.55**
Random Strings	54.0	22.8	25.8	24.0	31.65
Common-action Strings	40.6	20.4	12.4	18.6	23.00

Analysis. Only *Cleaned Subgoals* yield significant gains, whereas both non-semantic controls perform poorly (average 23–31%). Notably, *Common-action Strings* (23.00%) underperform *Random Strings* (31.65%), indicating that *misleading semantics can be more disruptive than pure noise*. When OpenVLA receives concrete but irrelevant action tokens (*e.g*. "take, place"), it is more likely to be confused than with random noise, leading to lower success rates. This further suggests that the VLA model indeed *interprets* the injected text, and semantic accuracy is critical.

(b) Effect of Structured Reasoning Composition

Question. Are the three components (Objects / Relations / Subgoals) and the logical cleaning necessary?

Experiment. We compare:

1. *Full three-part structure* (Objects + Relations + Subgoals);
2. *Raw Subgoals* (un-cleaned original subgoal text);

3. *Cleaned Subgoals* (logically cleaned, natural-language subgoals).

Table 3. Effect of structured reasoning composition.

Input Type	spatial	object	goal	10	Avg
Cleaned Subgoals	**90.4**	**79.4**	**81.6**	**58.8**	**77.55**
Full three-part structure	0.0	0.0	0.0	0.0	0.0
Raw Subgoals	64.8	48.4	51.5	31.8	49.13

Analysis. Directly injecting the full three-part structured text collapses performance to 0.0% across suites, indicating that OpenVLA's language embedding is highly sensitive to unseen structural tags (*e.g.* `<objects>`) and non-natural formats. Such *structural noise*, being out-of-distribution, disrupts task understanding entirely. Using *Raw Subgoals* avoids complete failure but remains much worse than *Cleaned Subgoals*: the average gap is 28.4 percentage points (49.13% vs. 77.55%). This shows that numbering (*e.g.* "1:"), redundancy, and irregular punctuation in raw subgoals substantially degrade decision consistency.

(c) Impact of Weighting among Reasoning Components

Question. How do loss weightings for Objects / Relations / Subgoals during distillation affect performance?

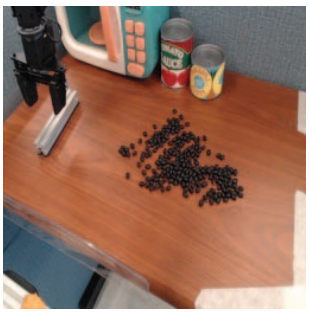
Experiment. We test multiple weight configurations: (1:1:1), (1:1:2), (2:1:2), and (1:2:2).

Table 4. Impact of component weighting during distillation.

Weight (O:R:S)	spatial	object	goal	10	Avg
Q-3B*	88.8	78.4	79.6	56.6	75.8
1:1:1	88.2	78.2	79.6	57.0	75.75
1:1:2	**90.4**	**79.4**	**81.6**	**58.8**	**77.55**
2:1:2	88.6	78.4	79.4	57.2	75.9
1:2:2	88.2	78.4	79.1	57.4	75.73

* **Q-3B** denotes the non-distilled **Qwen2.5-VL-3B** zero-shot baseline.

Analysis. We first assess the zero-shot (ZS) ability of the non-distilled Qwen2.5-VL-3B. It already achieves an average of 75.80%, outperforming baseline OpenVLA (72.1%) by 3.7 percentage points. This result is significant, as it validates our core hypothesis that modern lightweight VLMs possess considerable *latent structured reasoning* abilities useful for VLA. However, zero-shot performance is notoriously unreliable and unconstrained. A ZS VLM may generate non-template prose, hallucinate irrelevant details, or fail to adhere to the strict format required for stable downstream execution. The primary contribution of our Bridge-CoT distillation is not merely the +1.75% performance gain, but the transformation of an unreliable ZS reasoner into a robust, format-consistent planning module. This "harnessing" and "alignment" of the model's latent ability via distillation is a prerequisite for stable, large-scale evaluation. Furthermore, the (1:1:2) weighting schema highlights the role of the Objects (O) and Relations (R) components. Although O and R are discarded at inference time (per Tab. 3), they serve as a critical inductive bias during training. By forcing the model to first explicitly ground its understanding of the scene's entities (O) and their spatial context (R), we ensure the final Subgoals (G) plan is visually grounded rather than a non-contextual linguistic hallucination. This 'perceive-understand-plan' chain within the training objective is essential for generating semantically accurate and executable plans, even if only the final plan is used.



(a) Input Scene with many beans.

Listing (1) Generated Plans

```
ZS Q-3B
<objects> bean : black :
     1 bean : black : 1
     ...
(repeats until max
    token length)
BridgeLang
<objects> beans : black
     : many
<relations> <subgoals>
```

(b) Generated Plans.

Figure 3. Case 1: The "Beans" Case. Given the same input scene (a), the ZS model (b, top) fails by repeating symbols. Our BridgeLang (b, bottom) successfully abstracts and generates a valid plan.

Summary. The primary sources of improvement are:

1. the semantic consistency and structured expression of *Subgoals*;
2. logical cleaning aligning text with the VLA language space; and
3. emphasizing *Subgoals* in the distillation loss.

Lightweight VLMs act as strong *zero-shot structured reasoners*; Bridge-CoT–based distillation that emphasizes subgoal supervision further *refines* this ability to better match downstream action execution.

4.4. Qualitative Analysis

To complement our quantitative results, we provide two key qualitative case studies that illustrate the mechanisms behind BridgeLang's performance gains.

Case 1: Distillation for Robust Reasoning (The "Beans" Case). First, we demonstrate why our Bridge-CoT distillation (supporting Tab. 4) is critical for robust reasoning. As shown in Fig. 3, when the zero-shot (ZS) Q-3B model encounters a scene with many identical items, it enters a repetitive loop, failing to abstract. This "symbolic collapse" prevents it from generating a usable plan. In contrast, our finetuned BridgeLang learns to abstract the concept (*e.g.* beans: many), successfully completing the O-R-S chain. This provides concrete evidence that our distillation is essential for transforming an unreliable ZS model into a robust planner.

Case 2: Overcoming Policy Brittleness (The "Frozen" & "Jitter" Cases). Second, we show how BridgeLang's plans enhance the downstream VLA policy (supporting Tab. 1). We identify two common baseline failure modes, shown in Fig. 4. The baseline OpenVLA can suffer from

(a) Baseline "Frozen" state.

(b) Baseline "Jitter" state.

(c) Ours: Success (same seed).

Figure 4. Case 2: Policy Brittleness. The baseline model (under the same seed) gets stuck in "Frozen" (a) or "Jitter" (b) states. Our BridgeLang-enhanced model (c) successfully overcomes this brittleness to execute the task.

policy inertia ("Frozen" state, Fig. 4a), failing to initiate any action. In other cases, after a failed grasp, it enters a "jitter" loop (Fig. 4b), unable to recover. Under the identical seed, BridgeLang's explicit subgoals eliminate both failure modes, "kick-starting" the policy and enabling robust error correction (*e.g.* re-attempting the grasp), ultimately succeeding in the task (Fig. 4c). This shows our method transforms a brittle policy into an actionable one.

Furthermore, we observe this improvement is highly asymmetric: while BridgeLang frequently converts baseline failures into successes (as shown in Fig. 4), we rarely observe the opposite. That is, BridgeLang does not 'break' or introduce new failures to scenarios where the baseline already succeeds. This strongly suggests that our structured reasoning supplement acts as a purely positive robustness enhancement, guiding the policy out of brittle states without disrupting its existing competencies.

4.5. Efficiency, Interpretability, and Generalization

(a) Efficiency and Overhead BridgeLang adds minimal computational cost. On an NVIDIA RTX 3090, it performs a single pre-task inference with an average latency of **1.5 s**. This one-time cost contrasts sharply with the **50–200 s** VLA task execution time and, critically, *does not* affect real-time control latency, embodying the lightweight and deployable *Plug-and-Think* paradigm.

(b) Interpretability of Structured Reasoning The three-part output (Objects–Relations–Subgoals) forms a *readable* reasoning chain, improving interpretability and stability. During action-focused training, VLA models may see degraded vision–language understanding; BridgeLang's cleaned subgoals explicitly specify actionable targets (*e.g.* *"grasp black_bowl; place black_bowl on plate"*), which sharpens semantic grounding for action localization. Unlike CoT-VLA [43], which introduces substantial latency via full visual CoT, BridgeLang injects task decomposition and execution planning via *structured subgoals* without added runtime overhead. Conceptually, it enables *language as thought*, delivering explicit CoT without architectural changes.

(c) Cross-Task Generalization Although BridgeLang is never trained on LIBERO, it improves performance across all four suites by **2.6–8.2%**, demonstrating strong cross-task generalization. Thanks to language-driven abstraction, BridgeLang remains robust to different instruction styles (natural-language vs. task-specific language). While our experiments focus on OpenVLA, BridgeLang is architecturally agnostic and interfaces via text; hence, its structured outputs are, in principle, applicable to other VLA systems, suggesting promising model-agnostic transfer and system-level generality.

(d) Discussion and Summary In summary, BridgeLang delivers explicit, interpretable reasoning, cross-task generalization, and system reusability at *minimal* computational cost. The proposed External Structured Reasoning Layer demonstrates that lightweight, language-based reasoning can significantly improve task understanding and decision consistency in VLA *without* re-training or architectural modifications—advancing a practical *Think-before-Act* paradigm for embodied AI.

5. Conclusion and Future Work

The primary objective of this work was not merely to introduce a new reasoning module, but to systematically examine whether structured Chain-of-Thought (CoT) reasoning can substantially enhance task understanding and decision consistency in existing Vision–Language–Action (VLA) systems without any VLA retraining. Our experiments verify this hypothesis: by injecting semantically accurate and logically cleaned subgoal supplements prior to task execution, the VLA model achieves an average improvement of +5.45% in success rate.

Limitations and Future Work. We conclude by noting key limitations that open clear avenues for future research. First, BridgeLang's reasoning is currently static and open-loop; future studies should explore dynamic, closed-loop replanning triggered by environmental feedback or execution failures. Second, our "0.0% failure" finding reveals the VLA's fragility to structured input. This motivates moving beyond our natural language workaround to develop lightweight structure-aware adapters that could leverage the full `<objects>` and `<relations>` information. Third, our validation is limited to OpenVLA in simulation; testing generality on real-world robots and broader architectures (*e.g.* RT series, Octo) is a critical next step. Finally, our offline hard-label distillation could be enhanced by exploring feature-level or distribution-level knowledge transfer to better capture the teacher's reasoning.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pages 39–48, 2016.

[3] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

[4] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauzá, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

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

[6] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023.

[7] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14953–14962, 2023.

[8] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[9] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[10] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2(3):6, 2022.

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

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

[13] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR, 2021.

[14] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327, 2016.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[17] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[18] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

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

[20] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

[21] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[22] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431, 2024.

[23] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[24] Georgios Pantazopoulos, Malvina Nikandrou, Amit Parekh, Bhathiya Hemanthage, Arash Eshghi, Ioannis Konstas, Verena Rieser, Oliver Lemon, and Alessandro Suglia. Multitask multimodal prompted training for interactive embodied task completion. *arXiv preprint arXiv:2311.04067*, 2023.

[25] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[26] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning

with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

[27] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[28] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[29] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

[30] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[31] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

[32] Jingchen Sun, Jiayu Qin, Zihao Lin, and Changyou Chen. Prompt tuning based adapter for vision-language model adaption. *arXiv preprint arXiv:2303.15234*, 2023.

[33] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[34] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[35] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[37] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2020.

[38] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024.

[39] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

[40] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

[41] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[42] Yi Zhang, Qiang Zhang, Xiaozhu Ju, Zhaoyang Liu, Jilei Mao, Jingkai Sun, Jintao Wu, Shixiong Gao, Shihan Cai, Zhiyuan Qin, et al. Embodiedvsr: Dynamic scene graph-guided chain-of-thought reasoning for visual spatial tasks. *arXiv preprint arXiv:2503.11089*, 2025.

[43] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[44] Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191, 2023.

[45] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.