TEMPORAL ACTION-REASONING ALIGNMENT FOR STEERABLE VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent research increasingly explores generalist Vision-Language-Action (VLA) policies that leverage large-scale vision-language pretraining to generalize from only a few demonstrations. A promising direction is Chain-of-Thought reasoning, which enables models to reason step-by-step about the environment state and next subgoals before generating low-level actions. However, current VLAs tend to ignore intermediate thoughts when executing actions as they over-rely on vision. Prior work has tried to address this with information-dense intermediate representations, but such approaches struggle in low-data settings. We therefore propose a data-efficient framework TARA (Temporal Action-Reasoning Alignment) that improves policy steerability through precise temporal alignment between reasoning and actions. We heuristically decompose demonstrations into meaningful subgoals using implicit visual and behavioral information and train flow-matching policies to predict the actions required for each subgoal. We show that this temporal alignment increases attention to reasoning traces while preserving learned fine-grained behavioral knowledge. Moreover, aligned policies learn latent representations that better support compositional generalization and demonstrate robustness on LIBERO-Plus (+3.7%), generalizability through improved policy steerability on specifically constructed benchmark (+4.8%) across different subgoal representations, and significant improvements on a real-world Franka robot.

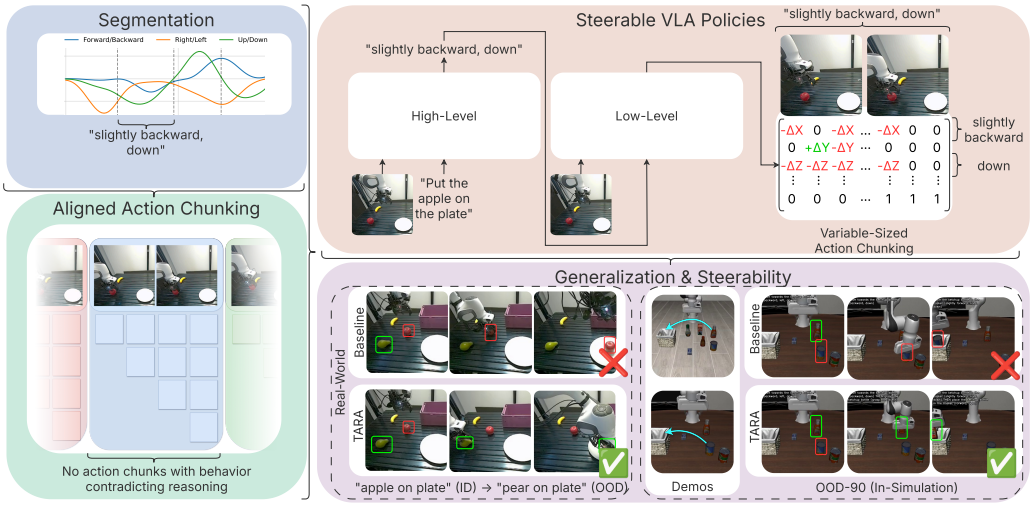


Figure 1: **Temporal Action-Reasoning Alignment (TARA)**, a data-efficient framework for steerable VLA policies via temporal alignment between reasoning and actions. Using heuristics such as translational velocities, we segment robot data into meaningful subgoals and construct action sequences such that their behavior is exactly explained by the subgoal. We employ a hierarchical architecture where a Vision-Language model (VLM) predicts the next subgoal and a flow-matching policy generates variable-sized action chunks grounded in the subgoal. Our approach improves generalizability from a few demonstrations through policy steerability.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

A central direction of current research in robot learning is the development of generalist policies that can generalize to unseen robot embodiments, tasks or environments from little demonstration data. While embodied datasets like OXE (Collaboration et al., 2023) exist, they are nowhere near the scale of language datasets, making large-scale pretraining difficult. With the advancements of VLMs, researchers started to utilize the large-scale vision-language understanding of VLMs by finetuning them to predict actions. These Vision-Language-Action Models (VLAs) (Zitkovich et al., 2023) serve as robot policies by generating actions grounded in specific robot embodiments, given an overall task instruction and an observation of the current environment state. Given a natural language instruction (e.g., "pick up the apple") and observation of the environment, VLAs predict the next actions, such as end effector movements. Typically, instead of predicting a single action, they predict a series of actions, also known as an *action chunk*. The technique of using fixed-sized chunks is known as Action Chunking (Zhao et al., 2023; Chi et al., 2023) and it has been shown to help better capture the temporal dependencies.

Inspired by the effectiveness of Chain-of-Thought (CoT) reasoning (Wei et al., 2023) in Large Language Models (LLMs), where models reason step-by-step before producing an answer (Kojima et al., 2023; Guo et al., 2025), recent research aims to endow VLAs with a similar ability to improve generalizability. However, do reasoning traces actually impact the agent's executed behaviors? The extent to which the policy is influenced by an (intermediate) instruction is referred to as *policy steerability* and poses a major limitation of current VLAs because they tend to over-rely on vision (Wanna et al., 2026a; Zhou et al., 2025; Fei et al., 2025), especially in low-data settings. This is further amplified by using Action Chunking (see Appendix B) because fixed-horizon action sequences extend past the completion of the current subgoal and thus contain steps that are unexplained by, or even contradict, the stated reasoning. As a result, the subgoal becomes an unreliable signal and policies learn to ignore it in favor of the more consistently informative visual state.

In this work, we propose **Temporal Action-Reasoning Alignment (TARA)**, a framework that improves generalization of VLAs through increased policy steerability by temporally aligning actions with reasoning. Specifically, we segment existing robot data into meaningful intermediate subgoals using rule-based heuristics based on translational shifts and train policies to predict variable-sized action chunks that exactly cover the remaining steps for completing the current subgoal. This way, the policy has to actively infer from the subgoal which and how many steps are required, thus ensuring the policy pays more attention to the CoT.

We show that policy steerability can be improved not only by using information-dense representations but also by aligning actions with meaningful subgoals while requiring less data. Additionally, we find that aligned policies learn latent representations that better support compositional generalization and pay more attention to subgoals while preserving learned fine-grained behavioral knowledge. Our approach improves generalization and robustness, achieving a 3.7% gain on LIBERO-Plus and a 4.8% gain on OOD-90 across different subgoal representations, along with significant improvements on a real-world Franka robot.

In short, we make the following contributions:

1. We introduce a framework for Action-Reasoning Alignment in flow-matching policies using variable-sized Action Chunking.
2. We propose rule-based heuristics based on translational movements or visual gripper poses for segmenting robot trajectories into precise, meaningful subgoals.
3. We show that alignment in robot data serves as a new branch for optimizing policy steerability in VLAs.

2 RELATED WORK

Variable-Sized Action Generation. Existing diffusion or flow-matching VLAs typically rely on fixed, empirically derived action chunk sizes that may vary across benchmarks (Intelligence et al., 2025). A common approach to enable variable-sized action sequences is to generate a chunk of fixed horizon and execute part of it open-loop (Chi et al., 2023) based on heuristics such as model

uncertainty (He et al., 2024; Liang et al., 2026) or fusing multiple chunks using attention gates (Jing et al., 2025). While these approaches optimize the trade-off between reactivity and foresight, they are not trained to predict variable action horizons. Pfrommer et al. (2025) propose a linear interpolation of action chunks to a horizon $\max_H$, however, the horizon is chosen from experience gained using Reinforcement Learning. Moreover, there are a handful of autoregressive approaches employing variable-sized action generation like Zhang et al. (2026b), where action generation is terminated using a distance-based termination condition. Lin et al. (2026) teach VLAs to switch from acting to reasoning when detecting an error or requiring human input, but also at language subtask boundaries. We focus on variable-sized action generation in flow-matching policies for action-reasoning alignment.

Policy Steerability. While natural language task instructions build the basis for most VLAs, they tend to be ignored due to their repetitive, formulaic structure, narrow diversity in current robot datasets, and low information density (Hou & Zhao, 2026; Wanna et al., 2026a), especially compared to observations which encode more dense information such as object locations, obstacles, the progress towards task completion, and the current pose of the robot. One branch of research addresses this modality imbalance with counterfactual information (Peng et al., 2025; Fang et al., 2026; Glossop et al., 2026), however, the augmentation pipelines are time-intensive and require separate policies for synthetic trajectory generation. In contrast, our approach works as a supplement to other methods and requires little additional augmentation, making the most of human demonstrations. Alternatively, instructions can be "densified" by incorporating implicit information encoded in the other modalities of robot datasets and reducing the task horizon by decomposing them into subgaols. The most straightforward way to decompose tasks are natural language subtasks (Ahn et al., 2022; Shi et al., 2025; Intelligence et al., 2025), however, they do not provide enough information to generalize to new behaviors. Other language representations include object-centric instructions (Smith et al., 2025) or per-step atomic motions (Belkhale et al., 2024; Zawalski et al., 2025; Chen et al., 2025). Atomic motions are essentially language translations of end-effector movements, although in practice they seem to prompt policies to sample from a distribution of contextually reasonable actions rather than executing blind, literal movements (Chen et al., 2026). Inspired by atomic motions, we introduce Direction, a representation designed to be compatible with Action Chunking and work as a supplement for subtasks by abstracting translational movements into magnitude-free vectors, e.g., "backward, slightly up".

As the density of language commands is restricted by the ambiguity of natural language, related work also investigates non-language abstractions such as code (Liang et al., 2022), goal images (Zhao et al., 2025a), goal videos (Zhou et al., 2024), action priors (Zhong et al., 2026; Yang et al., 2025b; Song et al., 2025) and other representations (Liu et al., 2026; Intelligence et al., 2026). The most popular non-language abstraction is to either point at a target location (Yuan et al., 2025; Chen et al., 2026) or to propose a visual trace of the end-effector movement (Zheng et al., 2024; Huang et al., 2025; Gu et al., 2023; Li et al., 2025; Niu et al., 2024). ThinkAct (Huang et al., 2025) uses an MLLM to predict 2D gripper end-effector poses over the visual scene until the end of the task by training to reason via Reinforcement Learning similar to Guo et al. (2025). While these abstractions provide denser information, VLMs typically struggle to plan using these abstractions (Smith et al., 2025) and to generalize with high dataset heterogeneity.

Current research seems to steer towards the utilization of multiple representations at once (Hirose et al., 2025; Lee et al., 2025; Chen et al., 2026; Intelligence et al., 2026). Embodied Chain-of-Thought (ECoT) (Zawalski et al., 2025) reasons step-by-step about object locations, gripper position, and atomic motions, but also high-level plans and subtasks before arriving at the action. ECoT-Lite (Chen et al., 2025) directly builds upon ECoT by incorporating Action Chunking and demonstrates that implicit CoT reasoning can be effective while reducing latency. Chen et al. (2026) train a policy on diverse abstractions and let the planner choose the most appropriate, as they argue that different representations complement each other with respect to policy steerability, while Intelligence et al. (2026) prompt their policy using subtasks, subgoal images, and episode metadata at each step. However, to demonstrate reasonable generalization, these approaches require large-scale datasets and tend to fail when demonstration data is scarce (Huang et al., 2026; Chen et al., 2026). We propose a framework that improves policy steerability more efficiently, demonstrating generalization capabilities even when data is scarce.

Action-Reasoning Alignment. Most methods utilizing language representations such as subtasks or atomic motions (Belkhale et al., 2024; Zawalski et al., 2025; Sun et al., 2025; Smith et al., 2025), including recent (Chen et al., 2026), seem not to employ Action Chunking at all, despite its shown advantages. while Chen et al. (2025); Intelligence et al. (2025; 2026) use fixed action horizons independent of subgoal. While Action Chunking is common in approaches employing non-language representations their choice of action horizon is empircal and unrelated from the subgoal e.g. Zhao et al. (2025b) predict subgoal images with a horizon of one second while predicting actions of a much smaller horizon. Methods that use visual traces (Huang et al., 2025; Li et al., 2025; Niu et al., 2024; Belkhale et al., 2024) predict a full trace from the current state to the end of the episode, despite using shorter action horizons. Alternatively, action-reasoning alignment can be realized by sampling multiple action chunks and letting a trained verifier select action sequences based on the subtask (Wu et al., 2026b). While verification is an effective method for scaling test-time compute, the pool of sampled actions needs to include at least a single correct action which, especially for Out-of-Distribution (OOD) tasks, is not always given (Singhi et al., 2026). Our approach drastically increases the likelihood of correct actions, which verification methods may benefit from.

3 METHODOLOGY

3.1 SEGMENTATION & ANNOTATION

Most existing approaches (Zawalski et al., 2025; Chen et al., 2026; Lin et al., 2026) rely entirely on proprietary MLLMs to segment and annotate trajectories with only little guidance. This approach often leads to suboptimal segmentations, such as misaligned boundaries. This is especially problematic given the noisy nature of imperfect human teleoperation. Because the quality of segments is crucial for our purposes, we opted for a more deterministic, rule-based segmentation strategy.

3.1.1 SUBTASK+DIRECTION

Inspired by Sun et al. (2025), we initially segment trajectories into three phases based on the state of the gripper: Open, Closing, and Closed. Specifically, we define Closing segments starting from the first gripper action switch until the robot makes a significant change in direction, signaling a successful grasp.

To assign semantic labels to these boundaries, we use Qwen-3-VL-32B-Instruct (Yang et al., 2025a). We extract up to 25 frames per trajectory, distribute them proportionally based on the length of each segment. The VLM is provided with these images alongside the segment boundaries and instructed to output high-level subtasks. See Appendix A.2 for details.

We then independently subdivide non-Closing subtasks into directions, which represent magnitude-free translational movements. In contrast to atomic motions (Belkhale et al., 2024), directions are less reliant on specific embodiments and pose an intuitive heuristic for trajectory segmentation based on spatial shifts while minimizing conflicting intents between labels and actual actions. As illustrated in Figure 1 (Segmentation), we set segment boundaries at zero-crossings of translational velocities i.e. whenever the robot changes direction in X, Y, or Z. We label segments rule-based with natural language text e.g. +X/-X as "forward"/"backward" and append the modifier "slightly" when this direction contributes between 12.5% and 25% to the overall shift in this segment.

3.2 TEMPORAL ACTION-REASONING ALIGNMENT

Recent approaches typically predict actions using a fixed chunk size H. However, as illustrated in Figure 2, completing a subgoal may require $L_{\text{subgoal}} < H$ steps and the remaining actions in the predicted chunk will inherently extend beyond the completion of that objective. In the worst case, a subgoal may only require a single final step for completion. As a result, the action sequence contains behaviors not explained by the subgoal or may even include contradicting behaviors.

To address this problem, we propose Temporal Action-Reasoning Alignment (TARA). Instead of predicting a fixed number of actions regardless of subgoal progress, the policy predicts only the actions necessary to complete the current subgoal. Consequently, the policy must infer what should be achieved from the reasoning process and learn how to accomplish it through the generated actions.

This encourages the model to pay more attention to the reasoning and to treat subgoals as atomic manipulation primitives rather than arbitrary segments of a demonstration trajectory.

Formally, for every step in a demonstration trajectory, we define the action horizon as $H = L_S$, where L_S denotes the number of remaining action steps until the current subgoal, or the overall task if no further subgoal exists, is completed. The horizon therefore satisfies $L_S \in [1, \max_H]$, where $\max_H$ is the maximum supported action horizon. We set the hyperparameter $\max_H$ such that only a minority of subgoals are truncated. The policy consequently predicts action sequences of variable length equal to the number of remaining steps for completing the current subgoal.

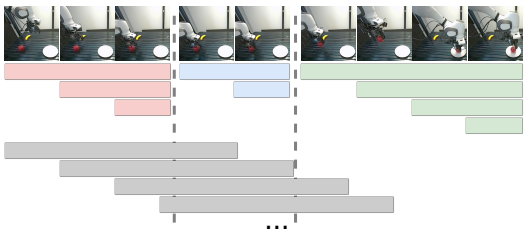


Figure 2: **Comparison between TARA (top) and standard fixed-horizon approaches (bottom) in the context of subgoals.** Conceptual subgoal boundaries are indicated with vertical lines.

3.3 Steerable VLA Policies

We adopt a hierarchical architecture inspired by Shi et al. (2025), where either PaliGemma-VLM (Beyer et al., 2024) or LLARVA (Niu et al., 2024) acts as a high-level planner, taking task instructions and observations to generate subgoals in various representations, and $\pi_{0.5}$ (Intelligence et al., 2025) functions as the low-level VLA policy which generates executable action sequences conditioned on the subgoal, the current observation, and optionally the original instruction. PaliGemma is a 2B-parameter VLM, which integrates Gemma (Team et al., 2024) with the SigLIP (Tschannen et al., 2025) image encoder, and we use it for Subtask+Direction subgoals. To explore non-language representations, we experiment with visual traces using LLARVA (Niu et al., 2024) as our planner. This model pretrains LLaVA (Liu et al., 2024), a VLM bridging a CLIP visual encoder (Radford et al., 2021) and the 7B-parameter LLM Vicuna (Peng et al., 2023), on a subset of the OXE dataset annotated with gripper traces. As our low-level policy, we use $\pi_{0.5}$, which pairs PaliGemma-VLM with a 300M-parameter action-head that produces continuous action chunks via flow-matching.

To implement variable-sized action generation in flow-matching policies, we use all-zero (no-op) actions to pad action chunks to $H = \max_H$ and train policies to actively predict no-ops. We find this design to be superior for our purposes compared to linearly interpolating action chunks (Pfrommer et al., 2025). During inference, we filter trailing no-ops. Our planners additionally predict the following subgoal if the current subgoal is near completion. See Appendix A.3 for ablations.

4 Benchmarks

For our in-simulation experiments, we evaluate on benchmarks built on LIBERO (Liu et al., 2023), which simulates a Franka Emika Panda arm and has been widely adopted for evaluating VLAs, as it provides a diverse collection of language-conditioned tabletop manipulation tasks with varying object arrangements and environments. The combination of popularity and narrow training data makes it an ideal platform for evaluating policy steerability in low-data settings, reflecting limited data availability common in real-world deployment. LIBERO tasks are organized into four suites, each consisting of 10 tasks and evaluating different aspects of VLA understanding. Additionally, LIBERO provides a rarely used subset, LIBERO-90, consisting of short-horizon tasks and environments that vary in their similarity to those in the other suites, including several completely unseen tasks, objects, and layouts.

OOD-90. As recent VLAs achieve near-saturated performance on LIBERO and most tasks can be solved without substantial reasoning or generalization, making it insufficient for evaluating policy steerability, we introduce OOD-90, a benchmark that tests generalization of hierarchical VLAs to the held-out set LIBERO-90 through policy steerability. We assign each task in LIBERO-90 an OOD score based on visual, lingual, and behavioral similarity to any task in the in-distribution base LIBERO dataset and quantize them into four equally sized OOD levels. Compared to existing generalization benchmarks, we focus on policy steerability in hierarchical VLAs by training high-

level planners on demonstrations for LIBERO-90, serving expert subgoals while we evaluate how well low-level policies leverage these subgoals to generalize. See Appendix A.1 for more details.

LIBERO-Plus. Additionally, we evaluate on LIBERO-Plus (Fei et al., 2025), which expands LIBERO by injecting perturbations across seven distinct dimensions, including visual shifts such as unseen camera perspectives (Camera), backgrounds (Background), lighting (Light), and noise (Noise), linguistic shifts such as modified task instructions (Language), and perturbations to initial object (Layout) and robot poses (Robot).

Real-world data. To conduct real-world experiments, we collect human teleoperated demonstrations for four Pick&Place tasks across two layouts, including placing a banana or an apple on a plate or a basket using a real-world Franka Emika Panda arm. Specifically, we record around 50 demonstrations for each of the 4 tasks, see Fig. 4 and Appendix A.4.

5 Experiments

TARA is designed to improve the effectiveness of Chain-of-Thought reasoning in VLAs by temporally aligning actions and reasoning to achieve increased policy steerability. We conduct our experiments primarily on derivatives of LIBERO, which evaluate the robustness and generalization capabilities of VLAs in low-data settings. Our experiments focus on the following questions: (1) Can TARA improve generalization and robustness? (2) Does CoT reasoning matter more with TARA? (3) Does TARA lead to better representation learning? (4) Is TARA effective with non-language subgoal representations?

5.1 Experimental Setup

Baselines. We compare our approach to a base policy without reasoning ($\pi_{0.5}$), an ECoT-Lite-like (Chen et al., 2025) method adapted for hierarchical inference which, for fairness, receives our Subtask+Direction annotations as the intermediate subgoal, and a ThinkAct-like (Huang et al., 2025) method which receives a simplified visual trace from the current state to the end of the task. Similar to Niu et al. (2024), we use an off-the-shelf gripper detector based on Detectron2 (Wu et al., 2019), trained on manually annotated images from a subset of OXE, to extract end-effector bounding boxes and compute the center coordinates to represent the gripper's position. Following Huang et al. (2025), we reduce these 2D paths to $K = 8$ representative points using the Ramer–Douglas–Peucker (RDP) algorithm (Ramer, 1972; DOUGLAS & PEUCKER, 1973), and train the planner and the policy on these visual traces for our ThinkAct-like baseline.

All methods use $\pi_{0.5}$ as the low-level policy and generate $H = 50$ actions, but only the first $H = 10$ are executed open-loop. We use LeRobot (Cadéne et al., 2026; 2024), an open-source framework for end-to-end robot learning, along with its PyTorch implementation of $\pi_{0.5}$, PaliGemma-VLM, default hyperparameters, and evaluation protocols to conduct our experiments and initialize our policies with `lerobot/pi05_base`. For all our LIBERO experiments, we fully finetune on the LIBERO dataset for 30K steps, using a batch size of 256 across 8 A100 GPUs. Our planners are trained using LoRA (Hu et al., 2021) and batch sizes of 128 across 8 A100 GPUs. For our real-world experiments, we fully finetune on demonstration data for four tasks, each with 50 demonstrations, for 5K steps using a batch size of 64 across 2 A100 GPUs, use OpenVLA (Kim et al., 2024), pretrained on the Bridge dataset (Walke et al., 2024) annotated with explicit reasoning traces (Chen et al., 2026) and additionally finetuned using LoRA for 5K steps using a batch size of 32 across 2 A100 GPUs, as our planner and deploy on a real Franka Emika Panda arm. See Appendix A.4 for more details.

5.2 Can TARA improve generalization and robustness?

In Figure 3, we compare the performance of TARA and our baselines on OOD-90 and LIBERO-Plus.

Even though our baselines, ECoT-Lite-like and ThinkAct-like, receive expert subgoals, they show no notable improvement on OOD-90 compared to our non-reasoning $\pi_{0.5}$ policy. This shows that VLAs fail to effectively ground their physical movement in intermediate reasoning steps to generalize to OOD scenarios. VLAs over-rely on vision Fei et al. (2025); Zhou et al. (2025); Wanna et al. (2026b), causing policies to default to behaviors most familiar to the visual state. This failure mode

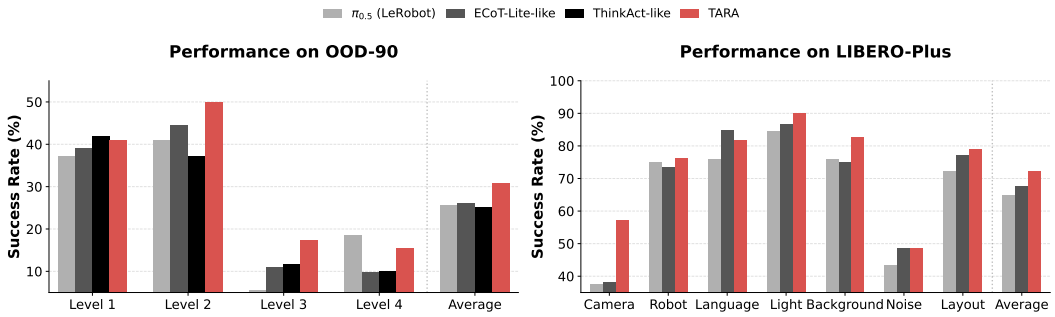


Figure 3: **Results on OOD-90 (left) and LIBERO-Plus (right).** For OOD-90 we test 10 episodes for each task. TARA achieves the best average on both benchmarks.

is reflected in our qualitative example in Figure 1, where baselines pick the object most strongly correlated with the visual state. In contrast, our approach significantly improves generalization to unseen tasks and environments through more effective utilization of intermediate reasoning steps. By closely aligning actions and reasoning, we reduce inconsistencies, thus subgoals become a more useful signal for action generation. Despite being notorious for weak policy steerability, we outperform baselines using language subgoals with TARA. Similarly, we outperform the baselines on the LIBERO-plus benchmark. TARA improves generalization to unseen poses/layouts and robustness to visual perturbations, showing both more effective use of the planner's reasoning and reduced reliance on vision. Although TARA seems to fall short on the Language category, language understanding is a known weakness of PaliGemma-VLM (Zhang et al., 2026a; Wu et al., 2026a), and subgoal errors naturally impact TARA more.

Figure 4 shows our real-world results. Compared to $\pi_{0.5}$ and our ECoT-Lite-like baseline, TARA can generalize to unseen objects and also demonstrates a slight improvement on unseen layouts. As in Figure 1, baseline policies default to familiar tasks like "put apple on plate" rather than the instructed "put pear on plate" as instructed, while TARA successfully conditions its movement on the given subgoal, which is consistent with our OOD-90 findings. Thus, our real-world experiments serve as an extension of OOD-90 and demonstrate that policy steerability in low-data settings is a real limitation of VLAs not restricted to LIBERO which TARA has shown to mitigate. Notably, most failures of TARA seem to be caused by reasoning mistakes of the upstream planner which in contrast to OOD-90 has not seen the tasks either, and not by limited policy steerability.

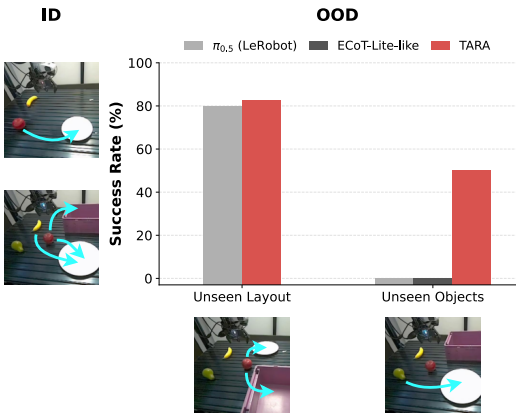


Figure 4: **Real-world experiments on a Franka robot.** We train our policies on 4 tasks, including placing an apple or banana on a plate or basket (ID) and test generalization to unseen layouts by swapping the positions of the basket and plate (2 tasks) and unseen objects by instructing it to place a pear on a plate (1 task) (OOD). We test each task for 5 episodes, where contact with the correct object contributes 50% and completing the task 100% success.

5.3 Does CoT Reasoning Matter More with TARA?

Next we analyze policy steerability in isolation and understand why TARA is helpful.

In Figure 5, we plot the Direction-following capability of our ECoT-Lite-like baseline and TARA in OOD-90 environments by task OOD score. Specifically, for each subgoal in our annotated LIBERO-90 dataset, we first reconstruct the state at the beginning of a subgoal by initializing the simulator with the exact robot and object poses used in the respective demonstrations and rollout ground-truth actions to this point, then pass ground-truth Subtask+Direction to the low-level policies, rollout the policies for the duration of the subgoal, and finally compute the cosine similarity between the Direction label and the physical path parsed as a Direction. We intentionally compare high-level Directions instead of low-level trajectories to

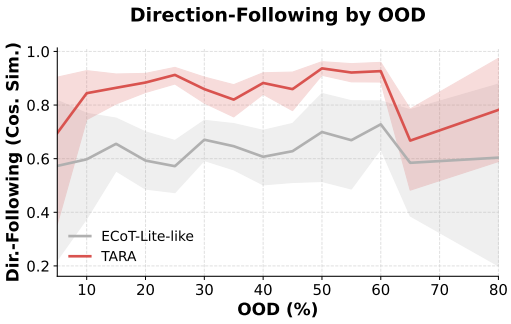


Figure 5: **Analyzing policy steerability in isolation on OOD-90.** Policy steerability as cosine similarity between the given and executed direction by OOD degree.

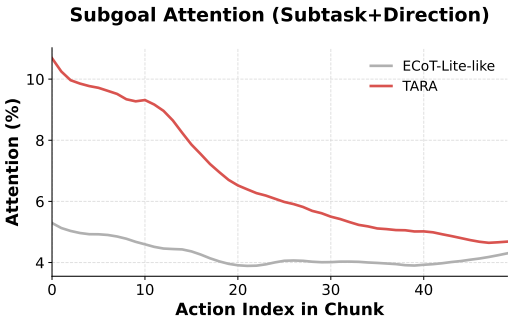


Figure 6: **Analyzing subgoal attention with OOD-90 inputs.** Subgoal attention as a percentage of total attention mass in TARA and our ECoT-Lite-like baseline.

measure policy steerability independent of fine-grained deviations that are not explained by the Direction.

Overall, our approach demonstrates superior policy steerability capabilities compared to our ECoT-Lite-like baseline. However, as shown by the discrepancy between our isolated results and OOD-90, policy steerability alone does not directly convert to generalizability. It requires careful balancing between multiple modalities and pretrained behavioral knowledge for practical generalizability.

To understand how this improved steerability is achieved, we analyze the policy's internal cross-attention mechanisms. For each step in the predicted action chunk, we isolate the attention mass allocated exclusively to the language tokens of the Subtask+Direction subgoal and plot this as a percentage of the total attention budget in Figure 6.

TARA allocates more than double its total attention budget to the subgoal at the initial action of the action chunk than the baseline. Over the course of the action horizon, TARA's subgoal attention exhibits a monotonic decay as the frequency of subgoals that require more steps decreases. In contrast, the baseline exhibits a flat attention profile throughout the chunk. This provides evidence that TARA supports the policy to attend to the subgoal and to ground movements not only in the observable state but also in the intermediate reasoning. In contrast, the baseline largely ignores the subgoal from the very beginning, which aligns with our earlier results. Due to the natural misalignment of actions and reasoning in current CoT approaches, policies deem reasoning traces not as useful during action-planning, confirming exactly why the baseline struggles to reliably follow intermediate reasoning steps.

5.4 Does TARA Lead to Better Internal Representations?

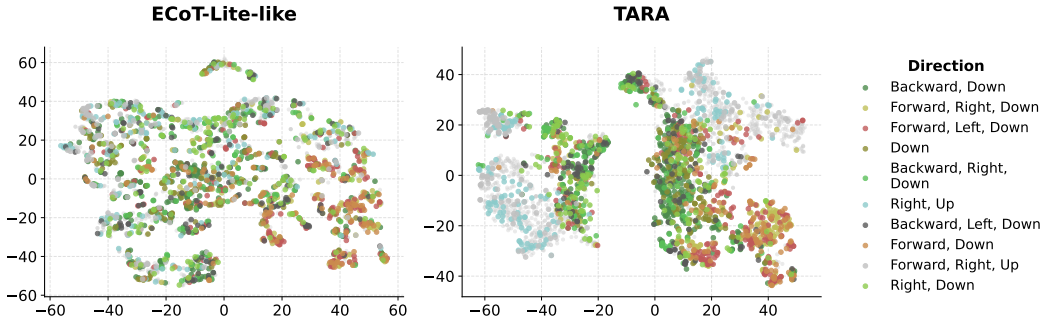


Figure 7: **Analyzing the latent space by Direction with OOD-90 inputs.** Unsupervised t-SNE projections for our ECoT-Lite-like baseline (left) and TARA (right), colored by Direction.

In addition to following intermediate reasoning steps, another motivation behind TARA is to regard subgoals as atomic motion primitives that ideally can be composed in novel orders to generalize to unseen tasks. In Figure 7, we compare the latent space of TARA and our ECoT-Lite-like baseline. We pass LIBERO-90 observation-subgoal pairs to each policy, extract and reduce the latent vector $\mathbf{z}$ using t-Distributed Stochastic Neighbor Embedding (t-SNE), and overlay Direction labels post-hoc.

While our baseline displays near-complete entanglement, TARA learns to organize its latent space into two separate clusters corresponding to either navigation or object manipulation subtasks. Each cluster shows horizontal gradients from downward to upward movements, and the navigational cluster additionally demonstrates a diagonal gradient from backward to forward motions.

Our TARA policy learns universal concepts of directional movement, which allows the composition of individual learned behaviors into arbitrary sequences to solve novel tasks.

5.5 Is TARA effective with non-language subgoal representations?

In contrast to our ThinkAct-like baseline, we use the RDP algorithm to determine subgoal boundaries such that the average segment length is similar to Subtask+Direction. Afterwards, we annotate each segment separately with $K = 8$ points. Table 1 compares two strategies for aligning visual traces to our previous baselines and TARA using Subtask+Direction. First, we follow ThinkAct by only including the remaining part of the visual trace from the current timestep until the end of the subgoal and second, similar to our Subtask+Direction representation, we predict subgoals from beginning to end.

Table 1: **Performance comparison of visual trace variations on OOD-90 and LIBERO-Plus (spatial suite).** The results are shown in success rate (%).

Method	OOD-90	L-Plus (S)
$\pi_{0.5}$ (LeRobot)	25.6	74.9
ECot-Lite-like	26.1	77.6
TARA	30.9	**80.8**
ThinkAct-like	25.2	71.3
From current	28.2	-
TARA (Visual Trace)	**31.8**	75.7

TARA (Visual Trace) outperforms all other methods on OOD-90 and improves upon its non-aligned counterpart ThinkAct-like on the spatial subset of LIBERO-Plus. While overall not as effective as Subtask+Direction, we show that applying TARA with denser non-language representations is still beneficial. Given the high OOD-90 score alongside ThinkAct-like's low performance, LLARVA or visual traces in general may simply be a poor fit for LIBERO-Plus. Interestingly, using full subgoals exceeds visual traces from the current state.

6 Conclusion

In short, our goal is to develop generalist VLA policies that can generalize from a few demonstrations using CoT reasoning. However, a prominent limitation of VLAs is their policy steerability, reducing the impact of reasoning traces on executed behaviors, especially when data is scarce. Thus, we propose TARA, a data-efficient framework that heuristically segments existing data into meaningful subgoals and temporally aligns actions and reasoning via variable-sized Action Chunking in flow-matching policies. Experiments and analysis demonstrate that policies pay more attention to the CoT, learn better representations, and improve generalizability through policy steerability. A crucial limitation of TARA is its sensitivity to the quality and segmentation strategy of subgoals. Policies rely more on the accuracy of reasoning traces and can quickly learn to over-rely on subgoals at the cost of fine-grained behaviors. It requires careful balancing between all modalities for practical generalizability. As high-level embodied reasoning poses a bottleneck for TARA, future work may explore planner pretraining or test-time scaling approaches, for example, subgoal verification. Additionally, our Direction representation enables alternative alignment strategies such as a bottom-up formulation where action and reasoning horizon are constant, which could serve as a basis to fully understand the relationship between Action Chunking and policy steerability. Finally, TARA could be a synergistic complement to prior policy steerability approaches such as Chen et al. (2026), where TARA may benefit from multiple subgoal representations of different horizons while premature policy subgoal-completion could serve as a signal for switching subgoal representations.

AI USE STATEMENT

In this work, we used generative AI tools for annotating segmented trajectories with subtasks, and assisting with the design and implementation of experiments to analyze underlying attention allocations and latent representations. We have not used generative AI tools for developing theoretical models or conceptual frameworks, proposing or refining hypotheses, assisting with translation, implementing methods, or interpreting results, and formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools for suggesting training hyperparameters for our real-world experiments, creating or editing software code, searching and summarizing relevant literature, brainstorming, and editing the text to improve readability. We have reviewed all AI-assisted work. Specifically, critical code was tested for correctness, LLM-generated text was treated as suggestions and manually revised, and relevant literature was manually cross-checked. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the source code in the supplementary material, including scripts and instructions for segmenting and annotating LIBERO datasets, training policies and planners using the hyperparameters mentioned in this work, and evaluating our approach on OOD-90.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint, 2024. URL `https://github.com/Stanford-ILIAD/openvla-mini`.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. In *https://arxiv.org/abs/2403.01823*, 2024.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL `https://arxiv.org/abs/2407.07726`.

Rémi Cadéne, Simon Alibert, Alexander Soare, Quentin Gallouédec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Khalil Meftah, Maxime Ellerbach, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. `https://github.com/huggingface/lerobot`, 2024.

Rémi Cadéne, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. Lerobot: An open-source library for end-to-end robot learning. In *The Fourteenth International*

Conference on Learning Representations (ICLR), 2026. URL https://arxiv.org/abs/2602.22818.

William Chen, Suneel Belkhale, Suvir Mirchandani, Oier Mees, Danny Driess, Karl Pertsch, and Sergey Levine. Training strategies for efficient embodied reasoning, 2025. URL https://arxiv.org/abs/2505.08243.

William Chen, Jagdeep Bhatia, Catherine Glossop, Nikhil Mathihalli, Ria Doshi, Andy Tang, Danny Driess, Karl Pertsch, and Sergey Levine. Steerable vision-language-action policies for embodied reasoning and hierarchical control. 2026.

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

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan

Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

DAVID H DOUGLAS and THOMAS K PEUCKER. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10(2):112–122, 1973. doi: 10.3138/FM57-6770-U75U-7727. URL https://utppublishing.com/doi/abs/10.3138/FM57-6770-U75U-7727.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in vlas, 2026. URL https://arxiv.org/abs/2602.17659.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Catherine Glossop, William Chen, Arjun Bhorkar, Dhruv Shah, and Sergey Levine. Cast: Counterfactual labels improve instruction following in vision-language-action models, 2026. URL https://arxiv.org/abs/2508.13446.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Nantian He, Shaohui Li, Zhi Li, Yu Liu, and You He. ReDiffuser: Reliable decision-making using a diffuser with confidence estimation. In Ruslan Salakhutdinov, Zico Kolter, Kather-

ine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17921–17933. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/he24e.html.

Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation, 2025. URL https://arxiv.org/abs/2509.19480.

Yuchen Hou and Lin Zhao. Langgap: Diagnosing and closing the language gap in vision-language-action models, 2026. URL https://arxiv.org/abs/2603.00592.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning, 2025. URL https://arxiv.org/abs/2507.16815.

Suning Huang, Jiaqi Shao, Ke Wang, Qianzhong Chen, Jiankai Sun, Yanjiang Guo, Mac Schwager, and Jeannette Bohg. Breaking lock-in: Preserving steerability under low-data vla post-training. *arXiv preprint arXiv:2604.23121*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Dong Jing, Gang Wang, Jiaqi Liu, Weiliang Tang, Zelong Sun, Yunchao Yao, Zhenyu Wei, Yunhui Liu, Zhiwu Lu, and Mingyu Ding. Mixture of horizons in action chunking, 2025. URL https://arxiv.org/abs/2511.19433.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space, 2025. URL https://arxiv.org/abs/2508.07917.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. Hamster: Hierarchical action models for open-world robot manipulation, 2025. URL https://arxiv.org/abs/2502.05485.

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. In *arXiv preprint arXiv:2209.07753*, 2022.

Yuanchang Liang, Xiaobo Wang, Kai Wang, Shuo Wang, Xiaojiang Peng, Haoyu Chen, David Kim Huat Chua, and Prahlad Vadakkepat. Adaptive action chunking at inference-time for vision-language-action models, 2026. URL https://arxiv.org/abs/2604.04161.

Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning, 2026. URL https://arxiv.org/abs/2505.11917.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. URL https://arxiv.org/abs/2310.03744.

Shuo Liu, Ishneet Sukhvinder Singh, Yiqing Xu, Jiafei Duan, and Ranjay Krishna. Vls: Steering pretrained robot policies via vision-language models. *arXiv preprint arXiv:2602.03973*, 2026.

Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4, 2023. URL https://arxiv.org/abs/2304.03277.

Zhenghao "Mark" Peng, Wenhao Ding, Yurong You, Yuxiao Chen, Wenjie Luo, Thomas Tian, Yulong Cao, Apoorva Sharma, Danfei Xu, Boris Ivanovic, Boyi Li, Bolei Zhou, Yan Wang, and Marco Pavone. Counterfactual vla: Self-reflective vision-language-action model with adaptive reasoning, 2025. URL https://arxiv.org/abs/2512.24426.

Samuel Pfrommer, Yixiao Huang, and Somayeh Sojoudi. Reinforcement learning for flow-matching policies, 2025. URL https://arxiv.org/abs/2507.15073.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763. PMLR, 2021.

Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972. ISSN 0146-664X. doi: https://doi.org/10.1016/S0146-664X(72)80017-0. URL https://www.sciencedirect.com/science/article/pii/S0146664X72800170.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL https://arxiv.org/abs/2502.19417.

Nishad Singhi, Christian Bialas, Snehal Jauhri, Vignesh Prasad, Georgia Chalvatzaki, Marcus Rohrbach, and Anna Rohrbach. Think twice, act once: Verifier-guided action selection for embodied agents, 2026.

Laura Smith, Alex Irpan, Montserrat Gonzalez Arenas, Sean Kirmani, Dmitry Kalashnikov, Dhruv Shah, and Ted Xiao. Steer: Flexible robotic manipulation via dense language grounding. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16517–16524. IEEE, 2025.

Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, Dong Wang, and Xuelong Li. Hume: Introducing system-2 thinking in visual-language-action model. *arXiv preprint arXiv:2505.21432*, 2025.

Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. Emma-X: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14199–14214, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.695. URL https://aclanthology.org/2025.acl-long.695/.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

Selma Wanna, Agnes Luhtaru, Jonathan Salfity, Ryan Barron, Juston Moore, Cynthia Matuszek, and Mitch Pryor. Limited linguistic diversity in embodied ai datasets, 2026a. URL https://arxiv.org/abs/2601.03136.

Selma Liliane Wanna, Agnes Luhtaru, Jonathan Salfity, Ryan Barron, Juston Moore, Cynthia Matuszek, and Mitch Pryor. Limited linguistic diversity in embodied AI datasets. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pp. 11058–11086, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.507. URL https://aclanthology.org/2026.acl-long.507/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Xiao-Ming Wu, Bin Fan, Kang Liao, Jian-Jian Jiang, Runze Yang, Yihang Luo, Zhonghua Wu, Wei-Shi Zheng, and Chen Change Loy. Vlanext: Recipes for building strong vla models, 2026a. URL https://arxiv.org/abs/2602.18532.

Yilin Wu, Anqi Li, Tucker Hermans, Fabio Ramos, Andrea Bajcsy, and Claudia PÃŠrez-D'Arpino. Do what you say: Steering vision-language-action models via runtime reasoning-action alignment verification. *2026 IEEE International Conference on Robotics & Automation (ICRA)*, 2026b.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. *arXiv preprint arXiv:2507.17520*, 2025b.

Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, and Jianye Hao. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation, 2025. URL https://arxiv.org/abs/2508.13998.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning, 2025. URL https://arxiv.org/abs/2407.08693.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models, 2026a. URL https://arxiv.org/abs/2601.03309.

Wenbo Zhang, Tianrun Hu, Hanbo Zhang, Yanyuan Qiao, Yuchu Qin, Yang Li, Jiajun Liu, Tao Kong, Lingqiao Liu, and Xiao Ma. Chain-of-action: Trajectory autoregressive modeling for robotic manipulation, 2026b. URL https://arxiv.org/abs/2506.09990.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713, 2025a. doi: 10.1109/CVPR52734.2025.00166.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models, 2025b. URL https://arxiv.org/abs/2503.22020.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models, 2026. URL https://arxiv.org/abs/2601.11404.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination, 2024. URL https://arxiv.org/abs/2404.12377.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization, 2025. URL https://arxiv.org/abs/2510.03827.

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 Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Appendix

A.1 OOD-90

To evaluate generalization through policy steerability, we use the LIBERO hold-out set LIBERO-90, consisting of 90 short-horizon tasks that exhibit varying degrees of similarity to the base LIBERO training data. We quantify how OOD each task is with respect to the training set along three complementary dimensions:

- **Language and Vision:** The task instruction and the first third-person observation are embedded using CLIP. Cosine similarity between embeddings of hold-out and training tasks is used to measure semantic and visual similarity.
- **Action:** We compare demonstration trajectories using Dynamic Time Warping (DTW), providing a measure of similarity between the required motion primitives.

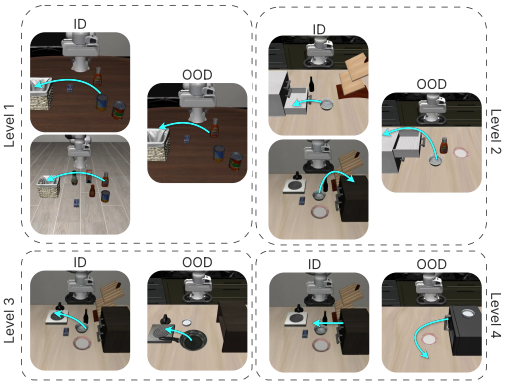


Figure 8: **Examplary ID-OOD pairs of OOD-90 for each OOD level.**

The language, vision, and action similarities are combined with equal weight to obtain an overall OOD score. The tasks are then sorted according to this score and divided into four equally sized OOD levels, where Level 1 contains tasks that are most similar to the training distribution, and Level 4 contains those that differ the most.

Our evaluation framework is designed specifically for hierarchical VLAs. While the low-level policy is trained only on the standard LIBERO dataset, the high-level planner is finetuned on demonstrations for LIBERO-90. During inference, the planner provides expert subgoals for unseen tasks, allowing us to evaluate not only policy steerability but also whether the policy can successfully leverage these expert subgoals to generalize beyond its training distribution. In Figure 8, we illustrate exemplary ID-OOD pairs for each OOD level.

A.2 Dataset Annotation Details

Table 2 shows the prompt for annotating our segments with subtasks using Qwen3-VL-32-Instruct.

A.3 Ablations

A.3.1 Representing variable-sized action chunks

Table 3 compares two approaches for representing varying action horizons: ILFM (Pfrommer et al., 2025), where we linearly interpolate action chunks with varying horizons to a fixed horizon of $H' = 25$ and add the original horizon as an additional dimension. During inference, the predicted action chunk is interpolated back to its original horizon. For padding with no-ops, action chunks are padded with no-op actions up to $\max_H = 50$, while longer action chunks are truncated to this maximum horizon.

Padding using no-ops consistently outperforms ILFM across all four OOD difficulty levels.

Table 3: **Comparing methods for representing dynamic action horizons**. OOD_90 results using Subtask+Direction shown in success rate (%). We compare ILFM and padding with no-ops.

Method	L1	L2	L3	L4	Avg
ILFM	26.4	28.2	12.7	8.2	18.9
Pad (no-ops)	**34.6**	**45.5**	**19.1**	**17.3**	**29.1**

Table 2: Exemplary prompt passed to Qwen3-VL-32B-Instruct formatted for task "pick up the ketchup and place it in the basket" with 3 segments.

Segment 1 (1.34 seconds):
`<image><image><image>`

Segment 2 (3.76 seconds):
`<image><image><image><image><image><image><image>`

Segment 3 (2.50 seconds):
`<image><image><image><image><image>`

The above segments show a robot manipulation trajectory that completes the task: 'Pick up the ketchup and place it in the basket'. Decompose the task into 3 subtasks based on the segments provided in sequential order. Pay attention to the robot hand and identify which subtask it is performing in each segment.

You should output strictly in a Python dictionary format: {{`segment_number: 'subtask', ...`}}, where `segment_number` is an integer in [1, 3].

Note that there are only a few long segments, so do NOT include any micro-subtasks like 'lift'. Instead, subtasks should capture the whole segment and focus on logical steps crucial for completing the task.

Hint: Segments [1] involve grasping. Start the corresponding subtasks with 'grasp'. Remember to approach/move towards the object in subtasks [0].

Example 1: Task 'put the white mug on the plate and put the chocolate pudding to the right of the plate' with 7 segments, Output: {`1: 'approach the white mug', 2: 'grasp the white mug', 3: 'place the white mug on the plate', 4: 'move towards the chocolate pudding', 5: 'grasp the chocolate pudding', 6: 'place the chocolate pudding to the right of the plate', 7: 'push the chocolate pudding to the right of the plate'`}

Example 2: Task 'open the top drawer and put the bowl inside' with 3 segments, Output: {`1: 'move towards the top drawer, open the top drawer and approach the bowl', 2: 'grasp the bowl', 3: 'put the bowl inside the top drawer'`}

While ILFM preserves a fixed output length, the interpolation process appears to distort the temporal structure of the action sequence, making it more difficult for the policy to learn accurate movements. In contrast, padding with no-ops preserves the original action sequence and only extends shorter horizons with explicit idle actions. Although more sophisticated methods for handling variable action horizons may exist, padding with no-ops has shown to be a simple and effective solution.

A.3.2 Drawbacks of improved policy steerability

As policy steerability increases, new challenges arise for hierarchical VLAs, particularly when the planner struggles at subgoal boundaries. These boundary issues were less apparent in unaligned hierarchical VLAs, as policies would often generate actions past the current subgoal, and practically steer the planner instead. While action-reasoning alignment ensures that the policy faithfully executes the planner's intent, it exposes the flawed state estimation of current high-level reasoners. To bridge minor state estimation discrepancies at subgoal boundaries, we train the planner to predict both the current and next subgoal whenever the current subgoal can be completed within 10 steps.

Table 4: **Impact of chaining current and next subgoal in TARA (Subtask+Direction).** Success rates (%) on the LIBERO benchmark. Success rates on individual suites are rounded to full integers.

Method	S	O	G	L/10	Avg
$\pi_{0.5}$ (LeRobot)	97	**99**	**98**	**96**	**97.5**
TARA	**99**	**99**	97	92	97.3
- chain Next	93	90	86	79	87.0

As shown in Table 4, this measure significantly impacts performance on the seen LIBERO benchmark. Without chaining both the current and next subgoal when the policy is close to subgoal completion, we increase the gap to the base policy. On the other hand, TARA reaches effectively equal performance on LIBERO.

A.4 Additional information on experimental setup

A.4.1 OOD-90

Policy. We initialize our $\pi_{0.5}$ baseline with the fully finetuned `lerobot/pi05-libero` checkpoint (finetuned for approximately 30K steps from `lerobot/pi05_base` on LIBERO) and use it without further finetuning. For all other baselines and for TARA, we initialize $\pi_{0.5}$ from `lerobot/pi05_base` and finetune with the same hyperparameters used for `lerobot/pi05-libero`: 30K training steps on the LIBERO dataset annotated with our subgoals, batch size 256 across 8 A100 GPUs, learning rate 2.5×10^{-5}, and 1K warmup steps. Baselines are trained only on the current subgoal, whereas TARA policies are additionally trained on the next subgoal, with probability 0.5 when the current subgoal completes within 15 steps and probability 1.0 when it is within 5 steps of completion.

Planner. We initialize PaliGemma-VLM with `lerobot/pi05_base` weights and finetune it on the held-out LIBERO-90 set using LoRA ($r = 16$, $\alpha = 32$, dropout $= 0.1$) for 30K steps, with batch size 128 across 8 A100 GPUs and learning rate 2.5×10^{-4}.

For our visual trace planners, we finetune LLARVA for 30K steps using LoRA ($r = 128$, $\alpha = 256$, dropout $= 0.05$) and learning rate 5×10^{-5}. Other settings are identical to PaliGemma-VLM.

A.4.2 LIBERO-Plus

We reuse the policies and further finetune PaliGemma-VLM for 10K steps and LLARVA for 20K steps on the base LIBERO dataset using the same hyperparameters described above.

A.4.3 REAL-WORLD EXPERIMENTS

Policy. As before, we initialize $\pi_{0.5}$ from `lerobot/pi05_base` and fully finetune it for 5K steps, with batch size 64 across 2 A100 GPUs, learning rate 2.5×10^{-5}, and 500 warmup steps. For TARA, we additionally apply the subgoal-chaining scheme described above.

Planner. We initialize OpenVLA with `Embodied-CoT/steerable-policy-embodied-reasoner-7b` weights, pretrained on the Bridge dataset annotated with explicit reasoning traces for a variety of subgoal representations. We finetune the planner using LoRA ($r = 16$, $\alpha = 32$, dropout $= 0.1$) for 5K steps, with batch size 32 across 2 A100 GPUs and learning rate 2.5×10^{-4}.

B IMPACT OF ACTION CHUNKING ON POLICY STEERABILITY

We use ECoT as an example to demonstrate the impact of action chunkin on policy steerability in VLAs.

B.0.1 EXPERIMENTAL SETUP

We finetune `Embodied-CoT/ecot-openvla-7b-oxe`, pretrained on OXE annotated with ECoT reasoning traces, on LIBERO-90 annotated with ECoT reasoning traces. We use LoRA with rank=64, $\alpha = 128$, learning rate $5e^{-4}$ with batch size of 64 across 8 A100 GPUs for 100K steps. We follow MiniVLA (Belkhale & Sadigh, 2024) and train an action tokenizer to compress action chunks with $H = 10$ into seven discrete tokens.

We evaluate the robustness to language perturbation of ECoT on a newly constructed suite based on tasks from LIBERO_90. We introduce language perturbations on three axis:

1. Referring Expressions: Referring target objects using a distinct visual property
2. Spatial Relationship: Referring target objects via their spatial relationship to other items
3. Context: Referring target objects via their semantic/functional meaning

We designed the suite such that for every task we paraphrase, at least two tasks with different target objects exist within the same scene in the training data, ensuring that simple memorization is ineffective.

B.0.2 RESULTS

Method	Setting	Referring Expressions	Spatial Relationship	Context	Average
ECoT (w/o AC)	Default	41.25	40.00	18.75	33.33
	w/ Plans	73.75	38.75	31.25	47.92
	Control	**85.00**	**85.00**	**67.50**	**79.17**
ECoT (w/ AC)	Default	67.50	33.75	32.50	44.58
	w/ Plans	75.00	37.50	33.75	48.75
	Control	**92.50**	**93.75**	**86.25**	**90.83**

Table 5: **Impact of ground truth plans and Action Chunking.** Comparison of ECoT performance with and without Action Chunking on our own generalization suite based on LIBERO. *w/ Plans* denotes providing the model with ground truth plans. Note that while adding plans improves performance, a significant gap remains compared to the *Control*. Furthermore, removing Action Chunking (*w/o AC*) results in a much larger relative gain when plans are introduced compared to the standard Action Chunking setting.

Table 5 shows that performance drops significantly when the target object is referenced via its spatial relationship to other items or its semantic/functional meaning. Contrary to expectations, providing ground-truth plans does not remotely recover performance to control levels.

Furthermore, the impact of ground truth plans seems even less significant when ECoT is trained with Action Chunking. This may be due to a larger misalignment between the reasoning and the actions during training.