EMMA-X: An Embodied Multimodal Action Model with Grounded Chain of Thought and Look-ahead Spatial Reasoning

Qi Sun[1*], **Pengfei Hong**[2*], **Tej Deep Pala**[1], **Vernon Y.H. Toh**[1],
U-Xuan Tan[1], **Deepanway Ghosal**[1 †], **Soujanya Poria**[3]

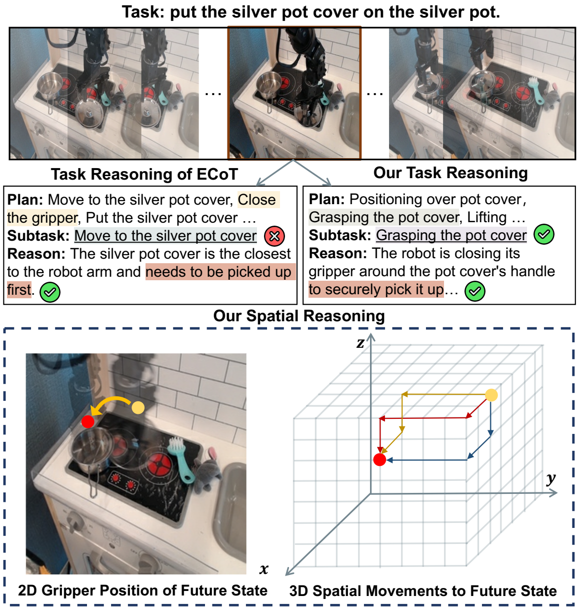
[1] Singapore University of Technology and Design
[2] HiThink Research
[3] Nanyang Technological University

Abstract

Traditional reinforcement learning-based robotic control methods are often task-specific and fail to generalize across diverse environments or unseen objects and instructions. Visual Language Models (VLMs) demonstrate strong scene understanding and planning capabilities but lack the ability to generate actionable policies tailored to specific robotic embodiments. To address this, Visual-Language-Action (VLA) models have emerged, yet they face challenges in long-horizon spatial reasoning and grounded task planning. In this work, we propose the Embodied Multimodal Action Model with Grounded Chain of Thought and Look-ahead Spatial Reasoning, EMMA-X. EMMA-X leverages our constructed hierarchical embodiment dataset based on BridgeV2, containing 60,000 robot manipulation trajectories auto-annotated with grounded task reasoning and spatial guidance. Additionally, we introduce a trajectory segmentation strategy based on gripper states and motion trajectories, which can help mitigate hallucination in grounding subtask reasoning generation. Experimental results demonstrate that EMMA-X achieves superior performance over competitive baselines, particularly in real-world robotic tasks requiring spatial reasoning. We make our codes, models, and datasets publicly available: `https://declare-lab.github.io/Emma-X/`.

1 Introduction

The robotic policy model aims to generate sequences of low-level action manipulation policies for robots. Traditional reinforcement learning-based robotic control methods often focus on narrowly defined tasks within fixed environments (Ma et al., 2024), hindering their ability to generalize beyond task-specific training data and limiting their applicability (Brohan et al., 2023b; Chi et al.).


Task: put the silver pot cover on the silver pot.
Task Reasoning of ECoT
Our Task Reasoning
Our Spatial Reasoning
z
y
x
2D Gripper Position of Future State
3D Spatial Movements to Future State

Plan: Move to the silver pot cover, Close the gripper, Put the silver pot cover …
Subtask: Move to the silver pot cover ☒
Reason: The silver pot cover is the closest to the robot arm and needs to be picked up first. ✓

Plan: Positioning over pot cover, Grasping the pot cover, Lifting …
Subtask: Grasping the pot cover ✓
Reason: The robot is closing its gripper around the pot cover's handle to securely pick it up… ✓

Figure 1: Comparison of our EMMA-X with ECoT in task reasoning. While both approaches utilize Gemini, our method also incorporates image sequence input, whereas ECoT relies solely on text input. We also illustrate an example of spatial reasoning.

Recent advancements in foundation models for vision and language have highlighted the remarkable scene-understanding and task-planning capabilities (Radford et al., 2021; Zhai et al., 2023; Touvron et al., 2023). These Visual-Language Models (VLMs) excel at breaking down complex tasks into manageable steps through chain-of-thought reasoning and demonstrate significant potential in planning. Despite their strengths, VLMs are not inherently designed to directly generate policies applicable to specific embodiment configurations in robotics. This limitation has spurred the emergence of Visual-Language-Action (VLA) models, which aim to bridge this gap by leveraging multimodal inputs to produce adaptive and generalized robotic actions for complex, multi-task scenarios

---

[*] Both authors contributed equally to this work.
[†] Now at Google Deepmind.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14199–14214
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

(Brohan et al., 2023a; Kim et al., 2024; Octo Model Team et al., 2024).

However, most of the existing VLA models often exhibit "muscle memory" response patterns, struggling to perceive scene variation and understand instructions as humans do when handling complex tasks or ambiguous commands. Zawalski et al. (2024) attempts to address this issue through visual and task reasoning, including the bounding box of the object, task segmentation, and the direction of predicted action, etc. Although they equip VLAs with an understanding of the current situation and task, they lack long-horizon spatial reasoning on how robots should move next. We hypothesize that the completion of subgoals or subtasks can be enhanced if the VLA incorporates look-ahead spatial reasoning, such as inferring the gripper's future 2D position and the 3D movement plans necessary for the gripper to reach that position. In particular, we train the VLA model to predict future position g_{t+k} of the gripper as checkpoints and use them to devise a high-level movement plan $\beta(g_t, g_{t+k})$. This plan informs the immediate action a_t at the current state s_t, ensuring decisions are both reactive to the present and aligned with long-term objectives. Similar to a delivery driver planning a route with key landmarks to make purposeful driving decisions, this approach optimizes task completion by balancing foresight and adaptability.

Additionally, another limitation in task reasoning provided by ECoT(Zawalski et al., 2024) is the absence of visual grounding when augmenting reasoning data using Gemini. We observe that Gemini frequently hallucinates due to a lack of holistic understanding of the setup and environment. As shown in Figure 1, the image shows that the robot already started to grasp the pot cover, while the task reasoning indicates the subtask is still "Move to silver pot cover", which conflicts with the following reasoning they provided.

In this work, we introduce the Embodied Multimodal Action Model with Grounded Chain of Thought Reasoning, **EMMA-X**. We develop a hierarchical embodiment dataset based on BridgeV2, consisting of 60,000 robot manipulation trajectories. For each state of a given trajectory, we generate detailed spatial reasoning grounded in the environment and task reasoning, such as the plans of how the robot should perform the subtask. As shown in Figure 1, we also generate the 2D gripper position, and 3D spatial movements of the gripper to transit to future states, which enable the VLA model to reason a long-horizon plan for accomplishing the task.

Furthermore, we utilize Gemini (Team et al., 2023) to generate grounded task reasoning for each observed state. To avoid the abovementioned reasoning conflict problem of task reasoning in ECoT, we propose a novel trajectory segmentation strategy, which leverages the opening and closing states of the gripper and the motion trajectory of the robot arm to segment the sequence of states into distinct segments. By grounding, we mean that, unlike ECoT, which prompts Gemini to generate subtask reasoning based solely on textual descriptions, our approach incorporates visual images segmented using the aforementioned strategy. As shown in Figure 1, our method can accurately provide the subtask "Grasping the pot cover" corresponding to the current robotic state. This illustrates that our strategy significantly reduces Gemini's hallucination issues by requiring it to construct a visual understanding of the environment, rather than relying solely on textual descriptions of the environment. Finally, we train our **EMMA-X** based on OpenVLA using our constructed hierarchical embodiment dataset.

The main contributions of our work are summarized as follows:

- We introduce a 7B-parameter embodied multimodal action model, **EMMA-X** created by fine-tuning OpenVLA with the grounded chain of thought (CoT) reasoning data.
- We synthetically construct a hierarchical embodiment dataset from the existing robot manipulation dataset, which includes the 3D spatial movements, 2D gripper position, and grounded reasoning.
- We propose a novel trajectory segmentation strategy that leverages the gripper's opening and closing states alongside the motion trajectory of the robot arm, facilitating both grounded task reasoning and look-ahead spatial reasoning.
- Our proposed **EMMA-X** achieves significant performance improvements over existing competitive baselines on various real-world robot tasks, especially in tasks where spatial reasoning is required.

2 Problem Formulation

2.1 Policy Imitation Learning

Given a set of expert demonstrations $\mathcal{D} = \{(\{s_t\}_{t=1}^{T}, \mathcal{T}_i, \{a_t\}_{t=1}^{T})\}_{i=1}^{N}$, where N is the number of demonstrations in the dataset, T is the number of states (image frames of the environment) for a data sample D_i, $s_i = \text{image}_i$ represents the state consisting of an image of the environment, $\mathcal{T}_i$ is a natural language task instruction, and a_i represents the action taken by the expert in that state, the goal is to learn a policy $\pi_\theta(a \mid s, \mathcal{T})$ that mimics the expert's behavior.

The policy π_θ is modeled by a Vision-Language-Action (VLA) model. In line with the OpenVLA setting, the policy outputs a generalized action as a 7-dimensional vector. This vector encodes the end-effector's (gripper's) velocity of Cartesian components (x, y, z), orientational components (roll, pitch, yaw), and the gripper's close-open action.

The goal is to find parameters θ that minimize the difference between predicted action and the expert's action.

2.2 Hierarchical Policy Imitation

We build on the above formulation by decomposing a general task $\mathcal{T}$ into a hierarchical structure consisting of finer-grained components: states, segments, and subtasks.

A **state** at timestep t, denoted s_t, represents the scene. The sequence of states for the i-th trajectory is $S_i = \{s_1, s_2, \ldots, s_T\}$, where T is the number of timesteps. An **action** a_t is taken at state s_t, and the corresponding sequence of actions is $A_i = \{a_1, a_2, \ldots, a_T\}$. A **segment** σ is a series of consecutive states, $\{s_t, s_{t+1}, \ldots, s_{t+k}\}$, contributing to a subgoal, with $\Sigma_i = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ representing the segment sequence for the i-th trajectory. In each segment, the robot performs similar actions. A **subtask** $\mathcal{S}$ consists of segments, $\{\sigma_1, \sigma_2, \ldots, \sigma_p\}$, to achieve a specific subgoal. Finally, a **task** $\mathcal{T}$ is a series of subtasks, $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m\}$, required to complete the overall objective.

Our Vision-Language-Action (VLA) model $\pi_\theta(a_t \mid s_t, \mathcal{T})$ predicts actions a_t for each state s_t by hierarchically decomposing tasks into subtasks. This ensures the end-effector's motion aligns with subgoal intents, enhancing the model's ability to execute complex tasks through manageable subtasks. We create a dataset $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{N}$, where $\mathcal{D}_i = \{S_i, \Sigma_i, \mathcal{T}_i\}$. Each state $s_t \in S_i$ is labeled with its subtask. Without such labeling, chain-of-thought training is infeasible. During inference, the model generates reasoning chains, including subtasks and relevant spatial information derived from visual scenes.

3 Methodology

In this section, we introduce our proposed framework in detail. Our EMMA-X encompasses three crucial designs: (1) Segmenting the trajectory based on the states of the gripper and the motion trajectory of the robotic arm. (2) Generating hierarchical planning including grounded task reasoning, 2D gripper positions, and 3D spatial movements. (3) Training the our EMMA-X based on OpenVLA with our constructed dataset.

3.1 Trajectory Segmentation

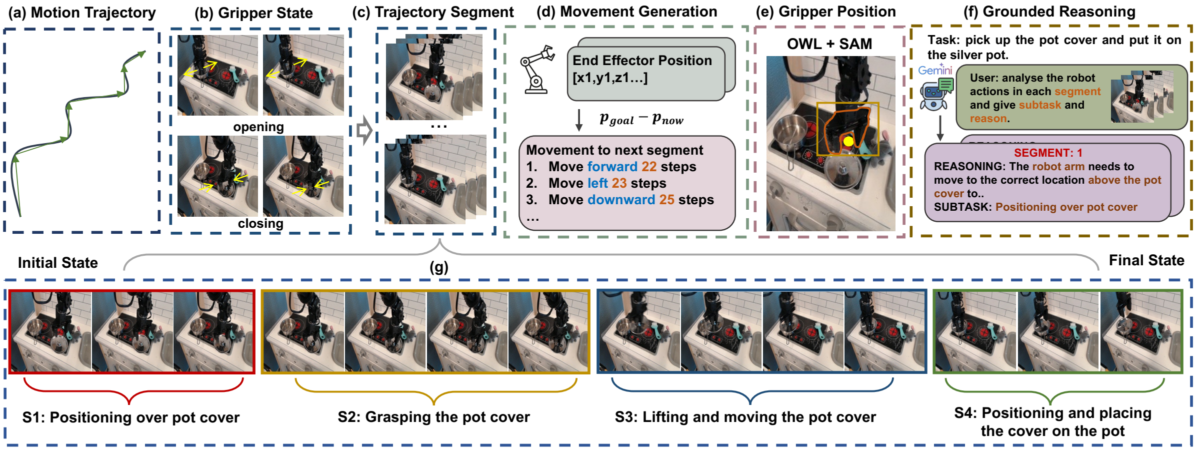
Why Segment Trajectories? The overarching goal of our work is to enhance Vision-Language-Action (VLA) models with grounded chain-of-thought (CoT) reasoning. We identified two key limitations in existing VLAs: 1) While existing VLAs improve task decomposition by breaking a task into subtasks and solving each using CoT (Zawalski et al., 2024), their CoT reasoning relies exclusively on textual scene descriptions [1]. This limits their reasoning capability for real-world scenarios. 2) They lack robust spatial reasoning abilities, essential for effective task planning and execution.

To address these limitations, we propose two key solutions: **Incorporating visual scene information:** Beyond textual prompts, we integrate visual inputs into Gemini to enable task decomposition into subtasks and generate high-level plans grounded in both visual and textual contexts. **Fine-grained movement plans:** We train the robot to determine *where to go* and *how to reach* a potential future state necessary for completing a subtask.

To implement these solutions, every state must be labeled with the subtask the robot is performing. However, our experiments revealed that directly annotating each individual frame via Gemini resulted in noisy labels, likely due to insufficient contextual information. To overcome this, we segment trajectories into sequences of consecutive states where the robot performs semantically similar actions. This segmentation provides richer context,

---

[1] We use "scene" and "environment" interchangeably throughout this paper.



Movement to next segment
1. Move forward 22 steps
2. Move left 23 steps
3. Move downward 25 steps
...

Task: pick up the pot cover and put it on the silver pot.

User: analyse the robot actions in each segment and give subtask and reason.

SEGMENT: 1
REASONING: The robot arm needs to move to the correct location above the pot cover to..
SUBTASK: Positioning over pot cover

Figure 2: Construction of our hierarchical embodied dataset. We first segment the trajectory. Then, we generate the 3D spatial movement that requires to transition to the end state of the segment. Based on segments, we recognize the 2D gripper position and generate the grounded task reasoning.

allowing Gemini to assign subtask labels more effectively.

Additionally, segmentation facilitates finding the gripper's position in a future state and planning its movement. At a given state s_t, the model predicts the movement plan required to reach the initial state of the next segment, s_{t+k}, before determining the policy a_t for s_t. Since $t + k > t$, this approach enables the model to perform look-ahead spatial reasoning, predicting the gripper's position at a likely future state, planning the motion trajectory, and generating a_t accordingly.

Our Segmentation Method. As shown in Figure 2(a) and Figure 2(b), we segment observation sequences by integrating the motion trajectory and the gripper states of the end effector. To achieve this, we utilize the **Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)** algorithm (McInnes et al., 2017), which effectively handles noise stemming from small fluctuations caused by imperfections in human demonstration. The flexibility of HDBSCAN enables the discovery of diverse trajectory patterns within the data.

We define a custom distance measurement to segment the end effector's trajectory, capturing both spatial and temporal information. Let $\mathbf{p}_i = (x_i, y_i, z_i)$ denote the 3D position, and $\mathbf{r}_i = (r_{ix}, r_{iy}, r_{iz})$ represent the 3D orientation of data point i. Additionally, let t_i represent the timestamp of this data point. The distance between two data points i and j is given by the following expression:

$$d(i,j) = \|\mathbf{p}_i - \mathbf{p}_j\|_2 + \lambda \|\mathbf{r}_i - \mathbf{r}_j\|_2 + \beta |t_i - t_j| \quad (1)$$

where λ is a weighting factor for the orientation component, and β controls the influence of the temporal distance [2]. This combined distance metric $d(i, j)$ ensures that both spatial movement and temporal separation contribute to the segmentation process. The inclusion of temporal information helps to distinguish trajectories that are spatially similar but occur at different times, while the orientation term captures changes in the end effector's rotation.

Applying the HDBSCAN algorithm with this distance metric allows us to segment the trajectory into meaningful clusters that reflect distinct movement patterns. However, the motion trajectory alone does not fully capture the interaction dynamics of the end effector with the environment. To address this, we incorporate the gripper state gs_i, which represents whether the gripper is in a *grip* (closed) or *loose* (open) position. A segmentation breakpoint occurs when the HDBSCAN algorithm detects a new cluster or when the gripper state changes between consecutive data points, formally defined as $gs_i \neq gs_{i+1}$.

This dual-segmentation approach effectively combines trajectory-based clustering with interaction-based segmentation, ensuring that the resulting segments capture both the motion patterns and the manipulation actions of the end effector. By integrating these two modalities, we achieve a richer and more accurate segmentation of the policy.

Finally, as a result of the segmentation process, we have a sequence of segments denoted as $\Sigma_i = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$, where n is the num-

---

[2] We use λ as 1 and β as 0.03 for best segmentation.

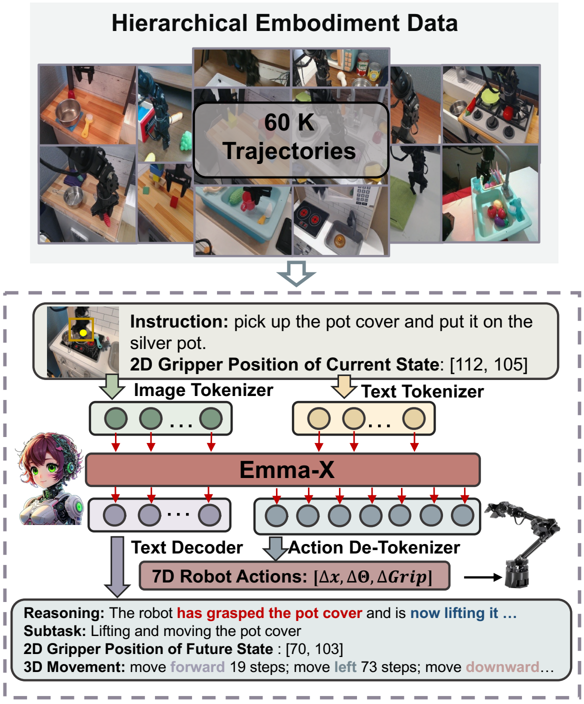

Hierarchical Embodiment Data
60 K Trajectories
Instruction: pick up the pot cover and put it on the silver pot.
2D Gripper Position of Current State: [112, 105]
Image Tokenizer
Text Tokenizer
Emma-X
Text Decoder
Action De-Tokenizer
7D Robot Actions: [Δx, ΔΘ, ΔGrip]
Reasoning: The robot has grasped the pot cover and is now lifting it ...
Subtask: Lifting and moving the pot cover
2D Gripper Position of Future State : [70, 103]
3D Movement: move forward 19 steps; move left 73 steps; move downward…

Figure 3: The overview of EMMA-X fine-tuned from OpenVLA using our hierarchical embodiment dataset.

ber of segments. Here, a segment is expressed as $\sigma = \{s_t, s_{t+1}, \ldots, s_{t+k}\}$, comprising k states.

3.2 Data Generation

After obtaining the segments, we generate hierarchical embodied planning data for each demonstration, as shown in Figure 2. For each segment of a demonstration, we produce the 2D end-effector position and 3D movements for the completion state of the current segment. Additionally, we generate grounded reasoning for the corresponding subtasks.

Why Look-ahead Spatial Reasoning? Consider the robot as a *delivery driver* tasked with delivering a package to a specific destination (the **goal**). The driver has access to a detailed high-level map of the city, which provides potential landmarks or checkpoints (s_{t+k}) along the way to the destination. To reach the goal efficiently, the driver performs two tasks: **Plans a high-level route**: The driver identifies likely landmarks and routes to guide them toward the destination, akin to predicting s_{t+k} and the movement plan $\beta(s_t, s_{t+k})$. **Executes immediate driving decisions**: While en route, the driver makes real-time decisions (a_t), such as turning left or stopping at a traffic signal, informed by the planned route and the current position s_t.

Without the ability to establish landmarks or checkpoints (future states) and plan routes based on them, the driver would rely solely on reactive decisions, leading to inefficiencies or incorrect paths. By integrating both the high-level plan and immediate feedback, the driver ensures purposeful and adaptive progress toward the goal. Following this analogy, we calculate the look-ahead gripper position and movement plan to reach there.

Look-ahead Gripper Position Generation. Following (Zawalski et al., 2024), we also use OWLv2 (Minderer et al., 2024) and SAM (Kirillov et al., 2023) to detect 2D gripper position, which can be seen in Figure 2(e). The difference is that they train the model to output only the gripper position for the current input state, whereas, in our data construction process, we use the current gripper position as input and predict the gripper position for the first state of the next segment. Thus, although both approaches utilize the gripper position, our model focuses more on predicting the gripper position in future states during training, rather than identifying its position in the current state. Let's consider for every state s_t, we obtain g_t, the gripper position of the first state of the next segment.

Look-ahead Movement Plan Generation. As shown in Figure 2(d), we infer the 3D spatial positions corresponding to the current state and the end state of the current segment using the state policy of the robot. Specifically, we calculate the displacement between these two positions to determine the direction and step size required for the manipulator to move from the current state to the end state. Following the motion language idea in RT-H (Belkhale et al., 2024a), we encode our high-level motion plans using a standardized template in Appendix E. By integrating look-ahead spatial reasoning, the model incorporates both reactive and proactive decision-making. It combines immediate context at the current state s_t with a high-level plan that predicts likely future states s_{t+k} and the corresponding movement strategy $\beta(s_t, s_{t+k})$. This dual focus enables the model to align immediate actions with the overarching goal, ensuring purposeful and adaptive task execution. Please note that this data is not directly executed as the robot's actions. Let's consider for every state s_t, we will obtain m_t, the movement plan to the first state of the next segment.

Category	Task	OpenVLA		ECoT		Emma-X (Ours)	
		h_Succ (%)	Succ (%)	h_Succ (%)	Succ (%)	h_Succ (%)	Succ (%)
Spatial Relation	Put the upper half of the carrot in the pot	30	10	35	20	**80**	**60**
Spatial Relation	Put the left half of the lemon in the pan	30	0	35	10	**55**	**20**
Spatial Relation	Put the blue cube on the left plate	25	20	5	0	**60**	**60**
Spatial Relation	Put the blue cube on the right plate	60	60	35	20	**90**	**90**
OOD Object	Put the banana in pot	70	50	45	40	**85**	**70**
OOD Object	Put the blue cube on the plate	**90**	**90**	20	10	85	70
OOD Object	Wipe the stove with towel	70	50	50	30	**90**	**90**
OOD Instruction	Pick up any object that is a kind of vegetable	40	30	15	0	**75**	**70**
OOD Instruction	Put the inedible object on the towel	0	0	25	0	**40**	**30**
OOD Instruction	Put the edible object on the towel	0	0	15	**10**	**35**	0
In Domain	Open microwave	50	30	25	0	**65**	**30**
In Domain	Close microwave	80	60	45	40	**100**	**100**
Average		45.41	33.33	28.75	15.00	**71.66**	**57.50**

Table 1: Experimental results of Emma-X and baselines on 12 real-world WidowX-250 robot manipulation tasks.

Grounded Chain-of-Thought Reasoning. As shown in Figure 2(f) and (g), we utilize Gemini [3] to derive the subtask corresponding to each segment, along with scene understanding and the reasoning behind the series of actions the robot needs to perform the subtask. Specifically, we take sequences of segmented images, and task descriptions as input to guide Gemini in generating the subtask and grounded reasoning for each segment. Compared to (Zawalski et al., 2024) that infer subtasks and their mapping to states solely from textual information, our approach first segments the sequence based on the robot's motion trajectory and gripper's state as explained in Section 3.1. After that, based on the given multimodal information, we generate the corresponding subtasks and the reasoning of each subtask. Note that each subtask can comprise multiple segments. For the i-th trajectory, we obtain the grounded reasoning from Gemini, defined as: $GR_i = \{(\sigma_k, \mathcal{S}_k, \mathcal{R}_k) \mid k = 1, \ldots, n\}$, where: - σ_k is the k-th segment, - $\mathcal{S}_k$ is the subtask label assigned to σ_k, - $\mathcal{R}_k$ is Gemini's justification for assigning subtask $\mathcal{S}_k$ to σ_k, and - n is the total number of segments in the trajectory. The prompt template can be seen in the Appendix B.

The Final Dataset. The final dataset for the i-th trajectory in the training dataset is defined as: $\{\mathcal{D}_i\}_{i=1}^N = \{X_i, Y_i\}_{i=1}^N = \{\{(s_t, \mathcal{T}_i), (m_t, g_t, GR_t, a_t)\}_{t=1}^T\}_{i=1}^N$, where $t = 1, 2, \ldots, T$, and T is the total number of timesteps in the trajectory.

3.3 Emma-X

In this section, we introduce the architecture of our proposed Emma-X which is a 7B-parameter VLA model fine-tuned from OpenVLA using our constructed hierarchical embodiment data. As shown in Figure 3, we adjust the text prompt with the current gripper position and add chain-of-thought training to enhance the ability of spatial reasoning and scene understanding before predicting the next robot action policy.

During the process of predicting for real robot testing, we input the task description, the current observation image, and the 2D gripper position detected in real-time by OWLv2 (Minderer et al., 2024) and SAM (Kirillov et al., 2023). Emma-X first outputs the subtask and a description of the current scene, including the spatial relationship between the target object in the image and the robotic arm, as well as the operational instructions required for the gripper to reach the goal of the current subtask. Additionally, Emma-X also predicts the target position the gripper needs to reach after completing the sub-task, including both the 2D location in the image and the 3D spatial movements. Finally, the model outputs the next 7D robot action policy for downstream manipulation.

4 Experiments

4.1 Implementation Details

To create the hierarchical reasoning dataset, we employed our data creation pipeline on full BridgeData-v2, which consists of approximately 60,000 trajectories paired with task instructions, resulting in an augmented dataset.

To train our VLA models, we employed OpenVLA, a 7B vision-language-action (VLA) model built upon the Prismatic vision-language framework and pretrained on the Open X-Embodiment dataset. For autoregressive training, we convert

---

[3] We used gemini-1.5-pro-latest for our data generation.

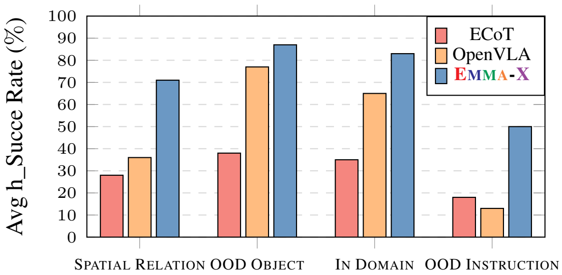


Figure 4: Experimental results on different categories of real-world robot tasks.

our continuous 7-dimensional action policy into discrete policy tokens, consistent with OpenVLA's methodology. We adhere to OpenVLA's training procedure and fine-tune the base model on our augmented dataset for 9 epochs until convergence. Once the reasoning and actions are converted into a unified string format, Emma-X is trained using a standard next-token prediction objective, optimizing the cross-entropy loss on predicted action tokens. Unlike OpenVLA, we use a smaller batch size of 64 to adapt to our specific hardware and experimental setup.

4.2 Robot Setup and Metrics

We evaluate our approach using the 6-DoF WidowX robot arm, as introduced in the Bridge V2 paper, which represents a standard benchmark for assessing generalizable robotic policies. The policy takes as input a single third-person camera feed and a natural language instruction, predicting end-effector velocity actions to control the robot.

To rigorously test the generalization capabilities of the policies, we develop a suite of challenging evaluation tasks that span multiple aspects: in-domain scenarios, out-of-domain (OOD) objects, spatial relationships, and OOD instructions. All policies are assessed on identical real-world setups to ensure consistency in camera angle, lighting conditions, and background. Each task is conducted over 10 trials, adhering to the methodology established by OpenVLA. If the robot can successfully achieve the task specified inside the prompts, it is counted as a success (**succ**) receiving a score of 1, otherwise, a score of 0 is assigned. Following OpenVLA, we also introduce a "half-success" (**h-succ**) metric that considers both the task goal and difficulty and assigns a 0.5 score only when the half-success criteria are met (Appendix C).

Models	SPATIAL RELATION		OOD OBJECT		OOD INSTRUCTION	
	h_Succ (%)	Succ (%)	h_Succ (%)	Succ (%)	h_Succ (%)	Succ (%)
EMMA-X	77	70	88	80	75	70
w/o m_t	42 (↓ 35)	37 (↓ 33)	63 (↓ 25)	55(↓ 25)	40 (↓ 35)	30 (↓ 40)
w/o g_t	32 (↓ 45)	30 (↓ 40)	45 (↓ 43)	35 (↓ 45)	45 (↓ 30)	30 (↓ 40)
w/o GR_t	22 (↓ 55)	10 (↓ 60)	45 (↓ 43)	40 (↓ 40)	25 (↓ 50)	20 (↓ 50)
w/o HDBSCAN	27 (↓ 50)	20 (↓ 50)	53 (↓ 35)	35 (↓ 45)	65 (↓ 10)	40 (↓ 30)
OpenVLA	38	30	70	50	40	30
w/ FT	28 (↓ 10)	23 (↓ 17)	65 (↓ 5)	50 (↓ 0)	15 (↓ 25)	0 (↓ 30)

Table 2: Models with different configurations.

4.3 Baselines

To comprehensively evaluate the performance of our proposed EMMA-X, we conduct extensive experiments across 12 different tasks on the real robot with several competitive methods.

OpenVLA (Kim et al., 2024): A VLA model based on large-scale VLM Prismatic-7b and pre-trained on the Open-X-Embodiment dataset(Collaboration et al., 2023).

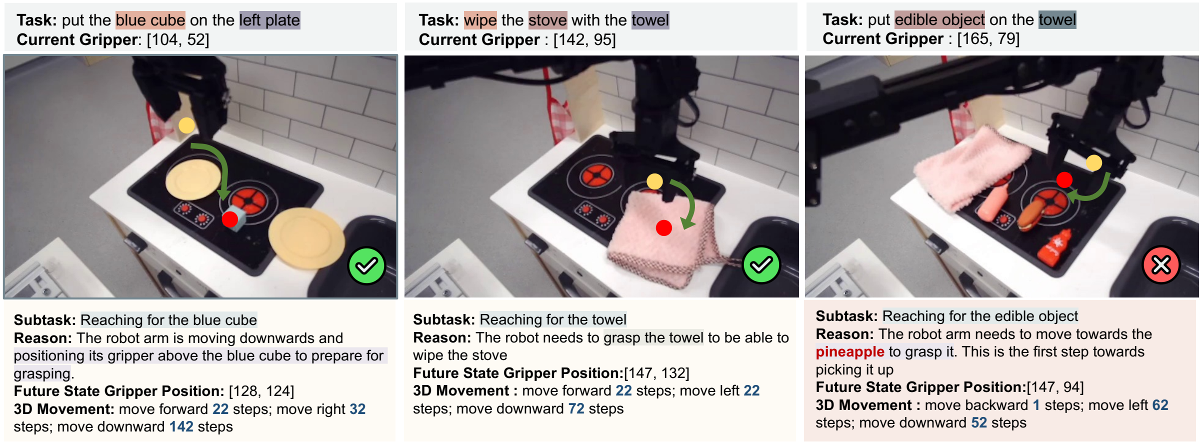
OpenVLA w/ FT: For a fair comparison, we finetuned the OpenVLA model on the BridgeV2 dataset for the same number of epochs following the same training setting in our method.

ECoT (Zawalski et al., 2024): A VLA model fine-tuned from OpenVLA on BridgeV2 dataset (Walke et al., 2023) with their generated chain-of-thought reasoning data.

4.4 EMMA-X Improves Policy Generalization

In this section, we compare EMMA-X with several baselines on 12 real-world robotic tasks. As shown in Table 1, our EMMA-X outperforms the strong baseline OpenVLA, with a 24.17% increase in task success rate and a 26.25% increase in half success rate. This demonstrates the effectiveness of our constructed hierarchical embodiment dataset. In addition, compared to ECOT, our EMMA-X shows significant gains, which can be caused by the following: 1) ECoT suffers from noisy training data, which causes hallucinations when faced with out-of-domain instructions or unfamiliar objects, leading to task failures. Interestingly, even for IN DOMAIN tasks, it performs poorly compared to other models, highlighting its limited reasoning capabilities. Our grounded task reasoning approach addresses this by incorporating the segmented visual images, ensuring more accurate task understanding. 2) EMMA-X enhances spatial reasoning by predicting the 2D gripper position of the end state of the current segment and 3D spatial movements to transit to it before predicting the next robot action policy.

As shown in Figure 4, we also compared the



Task: put the blue cube on the left plate
Current Gripper: [104, 52]

Subtask: Reaching for the blue cube
Reason: The robot arm is moving downwards and positioning its gripper above the blue cube to prepare for grasping.
Future State Gripper Position: [128, 124]
3D Movement: move forward 22 steps; move right 32 steps; move downward 142 steps

Task: wipe the stove with the towel
Current Gripper : [142, 95]

Subtask: Reaching for the towel
Reason: The robot needs to grasp the towel to be able to wipe the stove
Future State Gripper Position:[147, 132]
3D Movement : move forward 22 steps; move left 22 steps; move downward 72 steps

Task: put edible object on the towel
Current Gripper : [165, 79]

Subtask: Reaching for the edible object
Reason: The robot arm needs to move towards the **pineapple** to grasp it. This is the first step towards picking it up
Future State Gripper Position:[147, 94]
3D Movement : move backward 1 steps; move left 62 steps; move downward 52 steps

Figure 5: Qualitative examples of successful and failed cases with EMMA-X on real-world robot testing.

average performance across various categories of robotic tasks. Notably, our method achieved the most significant performance improvement in SPATIAL RELATION tasks, outperforming OpenVLA by 35% and ECoT by 29% in the h_Succ rate. These results strongly validate the effectiveness of our predicted 3D spatial movements. Furthermore, our method demonstrated substantial performance gains in OOD INSTRUCTION tasks, highlighting the efficacy of our grounded task reasoning.

4.5 Analysis

We trained several variants of EMMA-X to evaluate the roles of segmentation, look-ahead spatial reasoning, and grounded chain-of-thought (CoT) reasoning, which collectively constitute the core of EMMA-X. For this evaluation, we sampled 6 prompts across SPATIAL RELATION, OOD OBJECT, and OOD INSTRUCTION (prompts are indicated in magenta color in Section C). For each prompt, we conducted 10 rollouts under the same experimental setup as our main experiments.

Segmentation Greatly Helps the Policy. To evaluate the effectiveness of our segmentation technique, we conducted an experiment where sequences were segmented solely based on the gripper's (end effector) open and close positions. The results, reported in Table 2 under the *w/o HDBSCAN* condition, show a general performance drop of 10% to 50%. Notably, spatial reasoning performance experienced the most significant decline, with a drop of 50%. These findings demonstrate that the distance metric introduced in Eq. 1 is crucial for the segmentation process.

The Impact of Look-ahead Spatial Reasoning. To evaluate the importance of look-ahead spatial reasoning, we conducted two experiments: 1) EMMA-X was trained without explicitly predicting the gripper's position in the next segment, relying only on the predicted movement plan to reach the future gripper position of that segment (denoted as *w/o* g_t in Table 2). This assumes that EMMA-X implicitly infers the future gripper position. 2) We trained EMMA-X to predict the future end effector's position but without rolling out a movement plan to reach that position (denoted as *w/o* m_t in Table 2). The results reveal significant performance drops in both cases (25%-40% for "w/o m_t" and 30% to 45% for "w/o g_t"), with a more pronounced decline in spatial reasoning tasks (35% for "w/o m_t" and 45% for "w/o g_t"). Furthermore, the results suggest that predicting the future end effector's position is more critical, as the performance drop in the absence of 3D spatial movements to the next segment is less severe. We hypothesize that this may be due to OpenVLA's inherent spatial reasoning capabilities, which enable it to more easily transition between positions.

The Importance of Grounded CoT Reasoning. Grounded chain-of-thought (CoT) reasoning is a foundational element of EMMA-X. To assess its impact, we trained a variant of EMMA-X without grounded reasoning, while retaining look-ahead spatial reasoning in the data. The results show a marked performance drop by 43%-55%, highlighting that spatial reasoning alone is insufficient. Interestingly, the absence of grounded CoT reasoning resulted in a more severe decline in spatial reasoning performance compared to models where spatial reasoning capabilities were explicitly ablated. This underscores the critical role of grounded CoT in tackling complex reasoning tasks, including spatial reasoning. Therefore, we surmise that for enhancing the generalizable policies of Vision-Language-

Action (VLA) models, it is essential to improve their broader reasoning capabilities, encompassing object recognition, color understanding, abstraction, commonsense knowledge, and more.

Fine-tuning does not Improve OpenVLA. We sought to find whether fine-tuning OpenVLA on BridgeV2 could match the performance of EMMA-X. The results, shown in Table 2, reveal that OpenVLA's performance degrades by 5%-30% after fine-tuning with the worst performance observed for OOD INSTRUCTION. We hypothesize that this decline is due to overfitting, as BridgeV2 is already part of OpenVLA's pre-training dataset.

Qualitative Analysis on Real-world Robot Task. To qualitatively evaluate the effectiveness of our spatial and task reasoning in guiding robotic actions, we present two successful trajectories and one failed trajectory in Figure 5. From the left case, we find that the predicted gripper position corresponds to the end state of the subtask "reaching for the blue cube". The 3D movement provides a detailed path, clearly directed toward the "blue cube". We also include a failed trajectory where the "hotdog" is mistakenly identified as a "pineapple". This error propagates, impacting the prediction of the gripper's future position and preventing it from accurately picking up the "hot dog". More reasoning samples of our EMMA-X and ECoT can be seen in Appendix G and H.

5 Conclusion

We introduce EMMA-X, a 7B-parameter embodied multimodal action model designed to enhance spatial reasoning and task planning for robotic policy generation. We construct a hierarchical embodiment dataset enriched with grounded reasoning, including 2D gripper positions and 3D spatial movements. Furthermore, our proposed trajectory segmentation strategy reduces hallucination in task reasoning by grounding reasoning in visual images. The experimental results demonstrate the effectiveness of EMMA-X, showing significant improvements over existing baselines in tasks requiring long-horizon spatial reasoning.

Limitations

While EMMA-X shows promising performance, its latency remains higher compared to OpenVLA. This increased inference time primarily results from the additional tokens generated during the reasoning process. Specifically, EMMA-X generates approximately 10 times more tokens than OpenVLA. To mitigate this, a potential strategy is to predict all policies within a segment and only regenerate the policy if the predicted policy deviates significantly from the expected movement plan. Another limitation is the generalization capability of EMMA-X. Scaling the training process to incorporate a larger subset of the OXE dataset could enhance the model's ability to handle a broader range of tasks and robotic systems. Lastly, using SAM for detecting the gripper position can lead to inaccuracies. These errors may occur when the gripper is partially occluded by objects or positioned outside the image frame. Employing a more robust model for detecting and segmenting the robot hand could address these challenges and improve reliability.

Acknowledgements

Thanks to all co-authors for their hard work. The work is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-GV2023-010). Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. 2024a. Rt-h: Action hierarchies using language. *Preprint*, arXiv:2403.01823.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. 2024b. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023a. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *Preprint*, arXiv:2307.15818.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023b. Rt-1: Robotics transformer for real-world control at scale. *Preprint*, arXiv:2212.06817.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668.

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 SchÃűlkopf, 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 BÃijchler, 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, JoÃčo SilvÃľrio, 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, 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, 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. 2023. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`.

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. 2024. Racer: Rich language-guided failure recovery policies for imitation learning. *arXiv preprint arXiv:2409.14674*.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, and Feifei Feng. 2024a. Improving vision-language-action models via chain-of-affordance. *arXiv preprint arXiv:2412.20451*.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo S. Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, and Michael S. Ryoo. 2024b. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*.

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

Leland McInnes, John Healy, and S. Astels. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2:205.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. 2024. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36.

Vivek Myers, Bill C. Zheng, Oier Mees, Sergey Levine, and Kuan Fang. 2024. Policy adaptation via language optimization: Decomposing tasks for few-shot imitation. *arXiv preprint arXiv:2408.16228*. Conference on Robot Learning (CoRL) 2024.

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

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. 2024. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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

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, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. 2023. Bridgedata v2: A dataset for robot learning at scale. In *7th Annual Conference on Robot Learning*.

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

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952. IEEE.

A Related Work

Vision–Language–Action (VLA) Models. Early "generalist" robot policies such as RT-1 and RT-2 showed that conditioning Transformer backbones on paired vision, language, and action tokens yields robust multi-task control across embodiments (Brohan et al., 2023b,a). Scaling data breadth (Open X-Embodiment) and model size (RT-2-X) further improves generalisation (Collaboration et al., 2023), while Octo (Octo Model Team et al., 2024) and OpenVLA (Kim et al., 2024) demonstrate compositional and end-to-end finetuning strategies, respectively. Despite these advances, VLAs still under-utilise the inherent reasoning abilities of their language backbones when solving long-horizon manipulation tasks.

Spatial Reasoning with Chain-of-Thought. Recent work injects explicit spatial chain-of-thought (CoT) signals into policy generation. ECoT augments VLAs with textual rationales aligned to high- and low-level actions (Zawalski et al., 2024). LLaRA formulates control as visuo-textual conversations, aligning pixel-level gripper coordinates with actions (Li et al., 2024b). LLARVA leverages structured prompts and intermediate 2-D "visual traces" to couple perception and action (Niu et al., 2024). Chain-of-Affordance (CoA) further decomposes reasoning into sequential object, grasp, spatial, and motion affordances, boosting robustness in complex scenes (Li et al., 2024a). Our method extends this line by grounding CoT in predicted future states and hierarchical sub-goal selection.

Language-Enriched Robot Demonstrations. A complementary direction repurposes existing robot datasets with richer linguistic structure. Policy Adaptation via Language Optimisation (PALO) samples task decompositions from a vision-language model to enable few-shot adaptation (Myers et al., 2024). RACER augments demonstrations with failure-recovery trajectories and fine-grained language, pairing a VLM supervisor with a visuomotor actor (Dai et al., 2024). RT-H builds an action hierarchy by first predicting "language motions" as an intermediate abstraction, improving robustness across multi-task datasets (Belkhale et al., 2024b). These language-enriched corpora highlight promising paths for data efficiency and reliable long-horizon control, motivating our future work on scaling grounded CoT generation to open-world manipulation settings.

B Prompts and Sample Outputs

Template for prompting Gemini to generate subtasks and reasoning.

```
The robot successfully completed a task specified by the instruction: '{instruction
    }'. Here is a list of segments of images for the robot hand to perform the task
    specified by the instruction.

Based on the sequence of segments provided in sequential order, pay attention to
    the robot hand and identify which subtask it is performing in each segment.
    Provide the justification for why the subtask should be done based on the
    environment. You can assign the same subtask to multiple segments.

You should output in dictionary format: {segment_number: [subtask, reason for
    justification], ...}. The segment_number starts from 1 and must be an integer.
    The output dictionary keys correspond to each segment, and the output dictionary
     length should be the same as the number of segments.

{segment_count}

Segment 1: images in segment 1
Segment 2: images in segment 2
...
Segment n: images in segment n
```

We list the Gemini output of the example in Figure 1 as follows:

~~~
Here\'s a breakdown of the robot\'s subtasks in each segment:

```json{
   1: ["Positioning over pot cover", "The robot arm needs to move to the correct
       location above the pot cover to be able to grasp it. This involves adjusting
       its position and orientation."],
   2: ["Grasping the pot cover", "The robot is closing its gripper around the pot
       cover\'s handle to securely pick it up. This requires precision to ensure a
       stable grip."],
   3: ["Lifting and moving the pot cover", "The robot has grasped the pot cover and
        is now lifting it up and moving it towards the silver pot.  It must avoid
       collision with other objects in the environment."],
   4: ["Positioning and placing the cover on the pot", "The robot carefully
       positions the cover above the silver pot and lowers it into place. This
       requires aligning the cover with the pot\'s opening and releasing the grip
       gently to avoid dropping or damaging the cover or pot."]
   }
~~~

C Half Success Criteria

The half-success criteria to for the metric h_success are as follows. The prompts in magenta color are used to report the results in Table 2.

Task	**Half Success Criteria**
Put the banana in pot	Robot successfully grasps the banana.
Put the upper half of the carrot in the pot	Robot moves towards or grasps the correct target object.
Pick up any object that is a kind of vegetable	Robot moves towards the correct vegetable.
Put the left half of the lemon in the pan	Robot moves towards and correctly selects the left half of the lemon.
Put the blue cube on the plate	Robot grasps the blue cube.
Put the blue cube on the left plate	Robot grasps the blue cube and moves it towards the left plate.
Put the blue cube on the right plate	Robot grasps the blue cube and moves it towards the right plate.
Put the inedible object on the towel	Robot moves towards the correct inedible object.
Put the edible object on the towel	Robot moves towards the correct edible object.
Wipe the stove with a towel	Robot touches the towel but does not wipe the stove.
Open the microwave	Robot partially opens the microwave door.
Close the microwave	Robot partially closes the microwave door.

Table 3: The tasks used to evaluate the models and their half-success criteria.

D Segmentation Statistics

Average Number of frames per segment: 5.5
Average Number of segments per trajectory: 6.9
Average Number of frames per trajectory: 32.8

E Motion Plan Template

Translational Movements: *move (left/right)* x steps, *move (forward/backward)* y steps, *move (upward/-downward)* z steps.

Rotational Movements: *pitch (upward/downward)* α degrees, *yaw (left/right)* β degrees, *roll (clockwise/counterclockwise)* γ degrees.
Gripper Action: *(open/close)* gripper.

F Pseudo Code for Training EMMA-X and Running Inference

Notations defined in 2.2

Algorithm 1 Data Generation, Training, and Inference Process

For each sample i in Embodied Dataset, we have:
T: Number of time frames
$\mathbf{S} = \{s_t\}_{t=1}^{T}$: Images at each time frame t, where s_t is the image at time frame t
$\mathbf{G} = \{g_t\}_{t=1}^{T}$: Gripper poses at each time frame t (position, orientation, and open-or-close state)
$\mathcal{T}$: Task instruction in natural language format

Training Process:

```
while not converged do
    for each sample $i$ do
        $M_{\text{frames}\rightarrow\text{segments}} \leftarrow$ dual_segmentation$(G)$    ▷ Mapping from frame to segment
        $M_{\text{segments}\rightarrow\text{subtasks, reasons}} \leftarrow$ Gemini$(S, G, \mathcal{T})$    ▷ Mapping from segment to subtasks
        for each time frame $t \in \{1, 2, \ldots, T\}$ do
            $\sigma_t \leftarrow M_{\text{frames}\rightarrow\text{segments}}(t)$    ▷ Get segment for time $t$
            $\mathcal{GR}_t \leftarrow M_{\text{segments}\rightarrow\text{subtasks, reasons}}(\sigma_t)$    ▷ Get grounded reasoning from Gemini
            $g_t \leftarrow$ SAM$(s_t)$    ▷ Get 2D gripper position Using SAM model
            $g_{\text{end}} \leftarrow$ SAM$(S_{\text{end}})$    ▷ Get 2D gripper position at end of current segment
            $m_t \leftarrow$ Template$(g_t - g_{\text{end}})$    ▷ Translational change to movement plan in natural language
            $prediction \leftarrow$ EMMA-X$(\mathcal{T}, s_t, g_t, GR_t, g_{\text{end}}, m_t, a_t)$    ▷ Perform supervised fine-tuning (SFT) with label: $(\mathcal{T}, s_t, g_t, GR_t, g_{\text{end}}, m_t, a_t)$
        end for
    end for
end while
```

Inference Process:

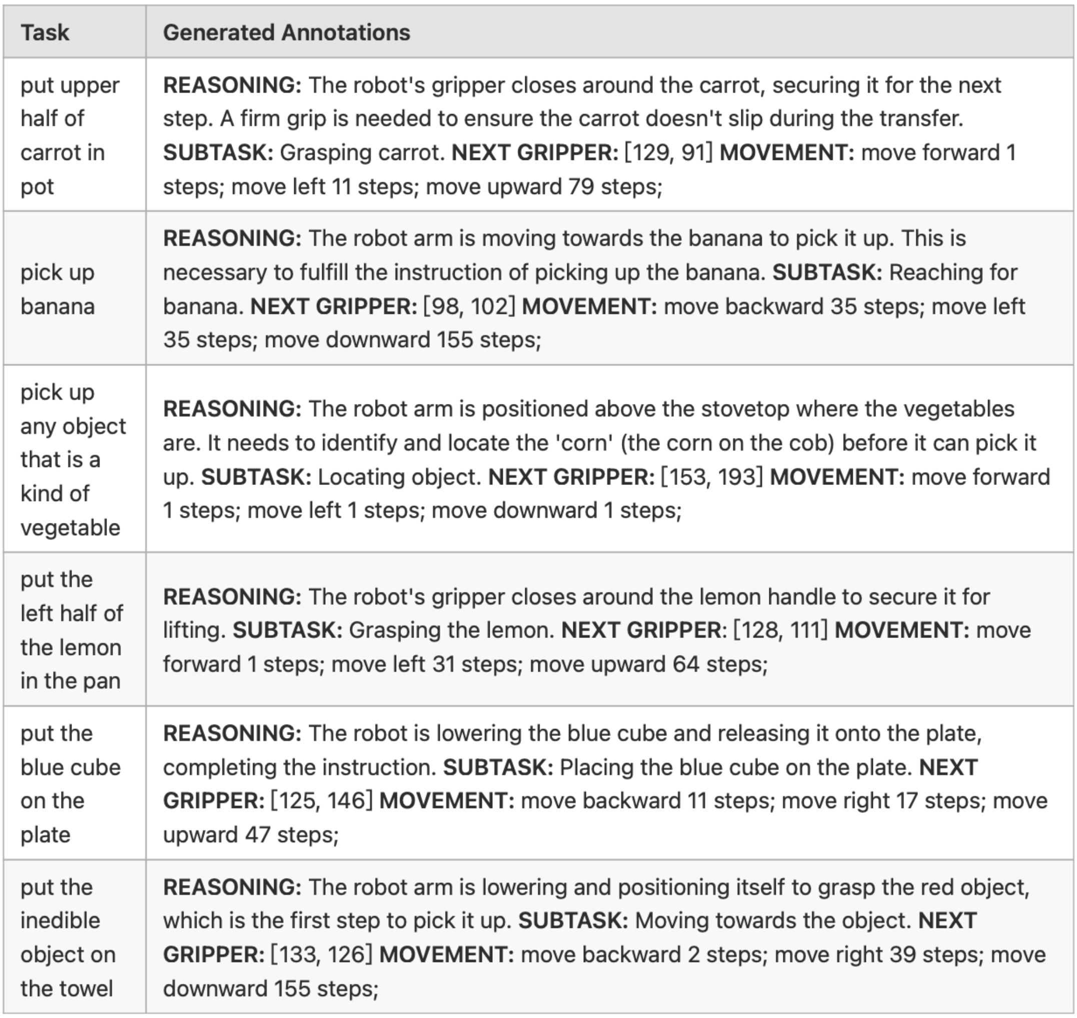
```
while Task not completed do
    $g_t \leftarrow$ SAM$(s_t)$
    $GR_t, g_{\text{end}}, m_t, a_t \leftarrow$ EMMA-X$(\mathcal{T}, s_t, g_t)$
    Control the robot using $a_t$, to get new $s_t, g_t$
end while
```

G Generated Annotations of our EMMA-X

As illustrated in the Figure 6, we present the model's reasoning across different real-world robot tasks, including its understanding of the current environment, task segmentation, and prediction of the robotic hand's 2D and 3D positional movement in the subsequent state.

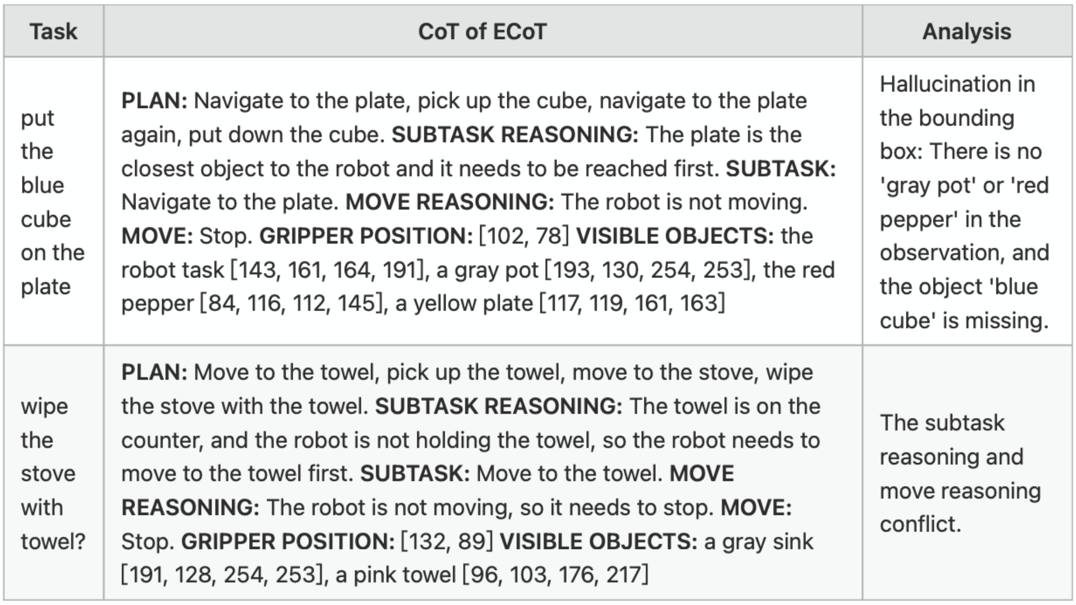
H Comparison of ECoT with our EMMA-X

We provide a comparison between the CoT data generated during our data generation phase and that of ECoT. As shown in the Figure 1, for the same frame in the BridgeV2 trajectory, ECOT's reasoning sequence tends to exhibit misaligned task segmentation. Specifically, even when the robotic hand has already grasped the silver cover in the current frame, ECoT often remains in the previous task phase, predicting actions toward reaching the silver cover.



Task	Generated Annotations
put upper half of carrot in pot	**REASONING:** The robot's gripper closes around the carrot, securing it for the next step. A firm grip is needed to ensure the carrot doesn't slip during the transfer. **SUBTASK:** Grasping carrot. **NEXT GRIPPER:** [129, 91] **MOVEMENT:** move forward 1 steps; move left 11 steps; move upward 79 steps;
pick up banana	**REASONING:** The robot arm is moving towards the banana to pick it up. This is necessary to fulfill the instruction of picking up the banana. **SUBTASK:** Reaching for banana. **NEXT GRIPPER:** [98, 102] **MOVEMENT:** move backward 35 steps; move left 35 steps; move downward 155 steps;
pick up any object that is a kind of vegetable	**REASONING:** The robot arm is positioned above the stovetop where the vegetables are. It needs to identify and locate the 'corn' (the corn on the cob) before it can pick it up. **SUBTASK:** Locating object. **NEXT GRIPPER:** [153, 193] **MOVEMENT:** move forward 1 steps; move left 1 steps; move downward 1 steps;
put the left half of the lemon in the pan	**REASONING:** The robot's gripper closes around the lemon handle to secure it for lifting. **SUBTASK:** Grasping the lemon. **NEXT GRIPPER**: [128, 111] **MOVEMENT:** move forward 1 steps; move left 31 steps; move upward 64 steps;
put the blue cube on the plate	**REASONING:** The robot is lowering the blue cube and releasing it onto the plate, completing the instruction. **SUBTASK:** Placing the blue cube on the plate. **NEXT GRIPPER:** [125, 146] **MOVEMENT:** move backward 11 steps; move right 17 steps; move upward 47 steps;
put the inedible object on the towel	**REASONING:** The robot arm is lowering and positioning itself to grasp the red object, which is the first step to pick it up. **SUBTASK:** Moving towards the object. **NEXT GRIPPER:** [133, 126] **MOVEMENT:** move backward 2 steps; move right 39 steps; move downward 155 steps;

Figure 6: The reasoning samples generated by EMMA-X during testing on real-world robot tasks.



Task	CoT of ECoT	Analysis
put the blue cube on the plate	**PLAN:** Navigate to the plate, pick up the cube, navigate to the plate again, put down the cube. **SUBTASK REASONING:** The plate is the closest object to the robot and it needs to be reached first. **SUBTASK:** Navigate to the plate. **MOVE REASONING:** The robot is not moving. **MOVE:** Stop. **GRIPPER POSITION:** [102, 78] **VISIBLE OBJECTS:** the robot task [143, 161, 164, 191], a gray pot [193, 130, 254, 253], the red pepper [84, 116, 112, 145], a yellow plate [117, 119, 161, 163]	Hallucination in the bounding box: There is no 'gray pot' or 'red pepper' in the observation, and the object 'blue cube' is missing.
wipe the stove with towel?	**PLAN:** Move to the towel, pick up the towel, move to the stove, wipe the stove with the towel. **SUBTASK REASONING:** The towel is on the counter, and the robot is not holding the towel, so the robot needs to move to the towel first. **SUBTASK:** Move to the towel. **MOVE REASONING:** The robot is not moving, so it needs to stop. **MOVE:** Stop. **GRIPPER POSITION:** [132, 89] **VISIBLE OBJECTS:** a gray sink [191, 128, 254, 253], a pink towel [96, 103, 176, 217]	The subtask reasoning and move reasoning conflict.

Figure 7: The reasoning of ECoT on our real-world robot tasks

Additionally, we also provide several ECoT reasoning outputs during real-world robot testing. From the first case shown in Figure 7, we observe that ECOT's reasoning in real-world tasks is prone to hallucination issues. For instance, hallucination in the bounding box: objects such as the "gray pot" or "red pepper" are not present in the observation, while the "blue cube" is missing. Such errors can severely mislead the robotic hand's subsequent actions. In the second example, we further observe conflicts between subtask reasoning and motion planning, indicating inconsistencies in its decision-making process.