Multimodal Causal Subtask Modeling for Scalable VLA Pipelines in Long-Horizon Manipulation

Yina Jian
Department of Computer Science, Columbia University
New York, NY, USA
yj2713@columbia.edu

Abstract

We propose Multimodal Causal Subtask Modeling (MCSM), a framework that structures long-horizon robot manipulation as causally grounded sequences of reusable short-horizon subtasks. The full framework reasons over visual observations, object geometry, affordance-based interaction predictions, and language goals to infer subtask sequences and dispatch pretrained policies. As an initial instantiation, we implement a data-efficient pipeline using LLM-guided decomposition with human-in-the-loop feedback, collecting subtask demonstrations independently and composing them into long-horizon trajectories. Evaluated on a real-world setup using the Universal Manipulation Interface (UMI), our approach achieves up to 85.6% reduction in training time compared to direct long-horizon training while preserving trajectory structure. These results validate subtask-level data reuse as a scalable strategy for long-horizon VLA systems and establish a modular foundation for the full MCSM framework.

1. Introduction

Vision-Language-Action (VLA) models have emerged as a promising paradigm for building generally capable robot agents [8], demonstrating strong generalization and instruction-following ability across diverse manipulation tasks. These advances are largely driven by large-scale data collection and multimodal pretraining. However, despite this progress, the real-world impact of VLA systems remains limited by the scalability and quality of the underlying data engine. In particular, existing pipelines have limited support for systematic data curation and efficient demonstration reuse, particularly for long-horizon tasks.

Long-horizon tasks present a fundamental challenge: as complexity increases, full-sequence demonstrations become costly to collect and prone to compounding errors across timesteps, yet current approaches have no systematic mechanism for data reuse across tasks.

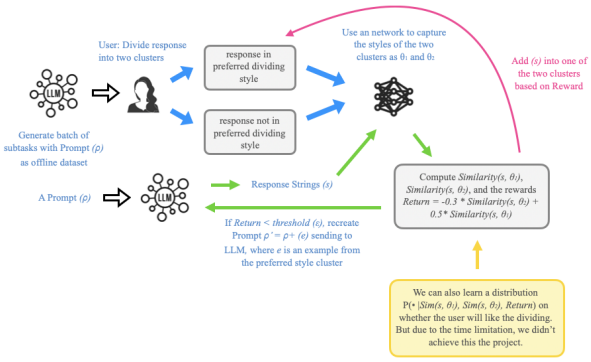

Figure 1. Method overview. The LLM generates subtask guidelines from a task prompt. A human expert clusters responses into preferred and non-preferred styles. A feature network captures style patterns and computes rewards to guide the LLM toward preferred-style outputs.

In this work, we propose *Multimodal Causal Subtask Modeling* (MCSM) as a principled framework for addressing these challenges. The core insight is that long-horizon tasks can be structured as causally grounded subtask sequences, where each subtask transition is conditioned on multimodal context — including visual observations, object geometry, affordance predictions, and language-specified goals. Under this framework, a learned causal model reasons over subtask dependencies and dispatches reusable short-horizon policies, enabling structured data reuse across tasks without requiring full-sequence demonstrations.

Contributions. This work makes the following contributions:

- We propose MCSM, a framework that structures long-horizon manipulation as causally grounded subtask sequences conditioned on visual, geometric, affordance, and language inputs, enabling scalable data reuse across VLA pipelines.
- We present an initial instantiation of MCSM via a

human-in-the-loop LLM steering mechanism for generating structured, action-grounded decomposition guidelines.
- We provide empirical evidence that subtask reuse significantly reduces data collection effort and training time, validating the feasibility of the proposed framework.

2. Related Work

Large Language Models in Robotics. With the increasing capability of LLMs, many frameworks attempt to adapt them for vision problems in robot manipulation [1]. However, most focus on subtasks within long-horizon problems or short-horizon tasks such as pick-and-place, and cannot adequately address compounding errors in long-horizon tasks.

Long-horizon Imitation Learning. Imitation learning has become popular for solving long-horizon tasks [4, 6]. These methods provide expert demonstrations of complete trajectories, allowing robot agents to learn and reproduce the entire process. Many demonstrate robustness to variations in object appearance, but require separate modules for recovery from unexpected situations, complicating data collection. Interactive approaches [2] incorporate human feedback during execution but remain focused on single-task settings without cross-task data reuse.

Reinforcement Learning with Probabilistic Models. Probabilistic control models [5] provide reliable performance but require redesigning reward functions for each new task, making cross-task adaptation difficult.

Affordance and Multimodal Task Planning. RT-Affordance [7] demonstrates that affordances serve as versatile intermediate representations that improve generalization in robot manipulation. Recent work has also explored grounding language instructions in visual affordances [1]. However, these approaches operate at the level of individual actions and do not model causal dependencies between subtasks. We propose to integrate affordance predictions into a broader causal reasoning framework for structured long-horizon subtask planning.

3. Method

The proposed MCSM framework consists of four modules: (1) a multimodal causal reasoning module for subtask decomposition, (2) a data collection module for subtask demonstrations, (3) a trajectory composition module for combining subtask outputs, and (4) a policy learning module for subtask execution. In this initial instantiation, module (1) is implemented via LLM-guided decomposition with human-in-the-loop feedback; modules (2)–(4) are implemented using UMI-based data collection, linear trajectory interpolation, and diffusion policy training respectively. Future iterations will replace each module with learned, multimodal-aware components as described in Sec. 6.

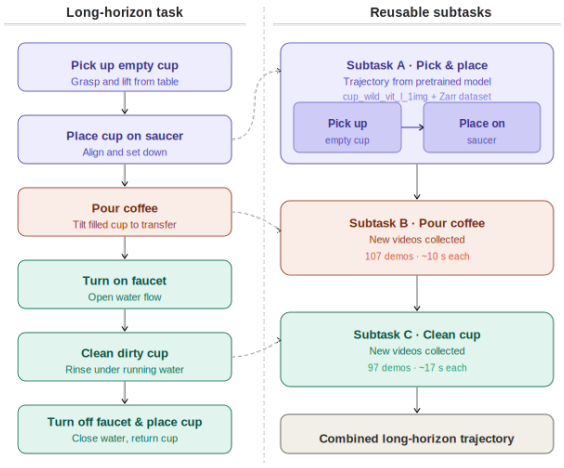

Figure 2. Task decomposition overview. The long-horizon task (left) is automatically divided into three reusable subtasks (right) by the LLM-guided pipeline.

3.1. Task Design

To test our hypothesis, we designed a long-horizon task that can be reasonably decomposed into short-horizon subtasks, ensuring all manipulations are achievable by UMI [4]. The long-horizon task consists of three stages: (1) pick and place an empty cup on a saucer, (2) pour coffee from one cup into the empty cup, and (3) clean the original cup with running water and return it. The full method overview is shown in Fig. 1.

Our method uses an LLM to divide this task into text guidelines for three subtasks: (1) pick-and-place cup on saucer, (2) pour coffee into the empty cup, and (3) turn on water, clean the dirty cup, place it on the saucer, and turn off the water. Since an existing dataset and pretrained model are available for the pick-and-place subtask [4], data collection is only required for two subtasks: pouring coffee and cleaning the cup.

3.2. Feature Matrix Extraction

To capture the preferred style of dividing guidelines, we used a prompt describing the environmental context, objects, and manipulation goal. We collected 447 responses from DeepSeek, and a human expert labeled 174 as preferred and 273 as non-preferred, forming our raw dataset.

We extracted features from both groups, including average sentence length, tokens (via `bert-base-uncased` from BERT), vectorizers, and style patterns (via `en_core_web_md` from spaCy and `textattack/roberta-base-imdb` from Transformers). These are combined into a feature matrix F, and

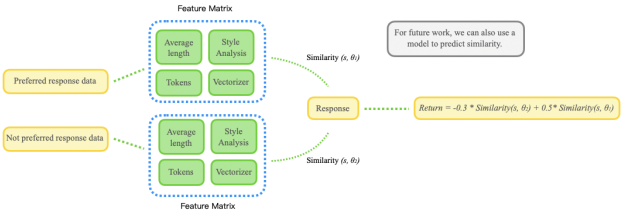

Figure 3. Feature matrix extraction pipeline. Cosine similarity between a new response and each cluster determines whether the response satisfies the reward threshold.

Figure 4. Sample video data frames captured by UMI fisheye cameras during the long-horizon task.

similarity is measured as:

$$\text{CosineSim}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{1}$$

The reward for a new response s is defined as:

$$R = -0.3 \cdot \text{Sim}(s, F_{np}) + 0.5 \cdot \text{Sim}(s, F_p) \tag{2}$$

If the reward does not satisfy the threshold, the LLM is prompted to regenerate.

3.3. Trajectory Extraction

After guidelines are generated, we used UMI to collect videos and extract trajectories for both the long-horizon task and each subtask. A sample of the raw fisheye video data is shown in Fig. 4. Trajectories consist of 3D location coordinates and quaternion orientations at each timestep. To combine subtask trajectories, we connect the endpoint of one trajectory to the start of the next:

$$p_i = p_e + \Delta p \cdot i, \quad \Delta p = \text{step\_size}, \quad i \in \mathbb{N} \tag{3}$$

where p_e is the last coordinate of trajectory τ_1 and p_s is the first coordinate of τ_2.

4. Experimental Setup

Data Collection. We collected 119 MP4 videos for the long-horizon task, 107 for pouring coffee, and 97 for cleaning the cup. All data are first processed into 3D positions at each timestep in CSV format, then passed to the diffusion policy model [3] for policy training. Feature extraction and prompt processing were implemented on Google Colab.

Training Environment. Since UMI requires Ubuntu, we configured an AWS EC2 instance on the `us-west-1` domain and set up the environment using Miniforge to run all data processing and model training with GPU support.

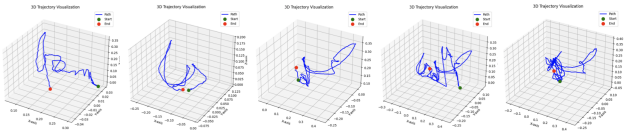

Figure 5. 3D trajectory visualization. Left to right: subtask 1 (pick-and-place), subtask 2 (pour coffee), subtask 3 (clean cup), combined trajectory, and trajectory from long-horizon demonstrations.

5. Results

5.1. Guideline Generation

The proposed pipeline consistently generates structured, preferred-style guidelines following a stable structure. A representative example is shown below:

- Locate the empty cup that is not on its saucer.
- Pour coffee from the original cup into the empty cup.
- Clean the original cup with water and place it back on its saucer.

Each sentence begins with an action verb and maintains a uniform, concise tone — a style we consider preferred because such sentences translate directly into robot manipulation action sequences.

5.2. Quantitative Evaluation

Similarity Comparison. Our approach improves average similarity to the preferred style by $+0.060$ (0.122 to 0.182) and the reward by $+0.019$ over randomly generated responses (see Appendix, Table 1).

Video Data Length. Average demonstration duration reduced from >40 s for the full sequence to $\sim$10 s and $\sim$17 s for the two new subtasks, demonstrating substantially reduced per-demonstration collection effort (see Appendix, Table 2).

Training Time. Training time was reduced from 16h 57m for the full sequence to 1h 40m and 45m for the two subtasks respectively — an overall reduction of approximately 85.6% (see Appendix, Table 3).

5.3. Trajectory Visualization

Fig. 5 shows the 3D trajectories extracted from videos. The composed trajectories exhibit similar spatial patterns and task phases compared to trajectories extracted from full demonstrations, suggesting that subtask composition preserves key structural properties of long-horizon behavior. Differences between individual trajectories are expected, as the demonstrations are not identical.

5.4. Training Loss Analysis

The long-horizon task converges smoothly to a low loss over 500 steps. Subtasks converge faster in wall-clock time

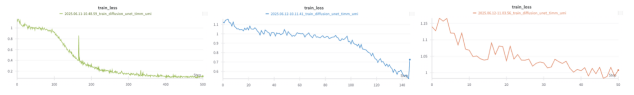

Figure 6. Training loss curves. Left (green): long-horizon task (500 steps). Center (blue): subtask 1 (147 steps). Right (orange): subtask 2 (50 steps). Subtasks train faster but with higher residual loss.

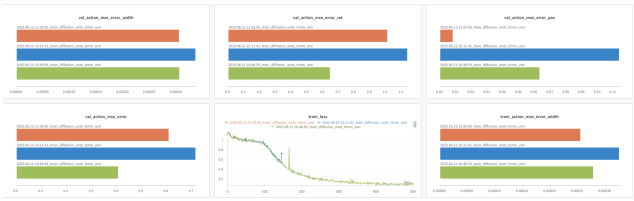

Figure 7. Final validation metrics (W&B). Red: subtask 2; blue: subtask 1; green: long-horizon task.

but reach higher final loss values, likely due to dropped frames during UMI preprocessing.

6. Discussion

The current implementation serves as an initial instantiation of the broader Multimodal Causal Subtask Modeling (MCSM) framework. In this instantiation, the LLM acts as a surrogate for the causal reasoning module, producing action-grounded subtask guidelines that approximate the structured decompositions a full causal model would infer. The modular design of the pipeline is intentional: each component — decomposition, data collection, trajectory composition, and policy learning — is designed to be replaceable with more principled implementations as the framework matures.

Each module in the current pipeline represents a deliberately minimal implementation chosen to validate end-to-end feasibility. The decomposition module (currently LLM-based) can be replaced with learned causal models that jointly reason over multimodal inputs; the trajectory composition module can be upgraded to learned transition policies that handle smooth inter-subtask handoffs; and the policy learning module can incorporate more expressive architectures such as diffusion transformers [3] or cross-embodiment pretrained models. Critically, the framework is designed such that each module can be improved independently without modifying the overall pipeline structure, providing a stable foundation for progressive integration of state-of-the-art components.

The full MCSM framework envisions replacing the LLM decomposition module with a learned causal model that reasons jointly over visual observations, object geometry, affordance-based interaction predictions, and language-specified task goals. Specifically, subtask boundaries and dependencies will be modeled as causal relationships over token sequences, where each token encodes the state of the manipulation scene at a key transition point. A diffusion-based planner will sample over the space of possible next subtasks conditioned on the current multimodal context, and the most probable subtask sequence will be selected to dispatch the corresponding pretrained or fine-tuned policy.

A current limitation is that we have not yet implemented extraction of action sequences from the trained diffusion model or deployment on a real robot arm. Based on UMI results [4], we believe trained subtask policies can generate valid trajectories connectable via path-finding algorithms.

Open Questions. This work surfaces several open questions we believe are valuable for the community. First, when subtask boundaries are ambiguous — for example, the transition between "pour coffee" and "return cup to saucer" — should the segmentation point be determined by affordance signals, language context, or a learned boundary detector? Second, is linear trajectory interpolation sufficient for real-robot deployment, or does reliable subtask composition require a learned transition policy that accounts for dynamics and contact? Third, under what conditions does cross-embodiment subtask reuse hold — for instance, can a subtask policy trained on UMI transfer to a standard robot arm without re-collection? These questions are natural next steps for the MCSM framework and we hope they invite discussion at the workshop.

Future work will focus on: (1) developing the full multimodal causal reasoning module that integrates visual, geometric, affordance, and language inputs for subtask inference; (2) replacing the current LLM decomposition with learned token-based causal subtask models; (3) collecting more diverse data across object categories and task types to improve model robustness; and (4) deploying the full MCSM pipeline on a physical robot arm for end-to-end evaluation.

7. Conclusion

We introduced MCSM, a modular framework that structures long-horizon robot manipulation as causally grounded sequences of reusable short-horizon subtasks conditioned on visual, geometric, affordance, and language inputs. Our results show that the adapted LLM generates consistent preferred-style guidelines, that combined subtask trajectories resemble those from long-horizon demonstrations, and that data collection and training time are substantially reduced (approximately 85.6%). This suggests subtask-level data reuse can serve as a scalable foundation for the MCSM framework and a practical alternative to directly collecting long-horizon demonstrations in data-driven manipulation pipelines.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, et al. Do as i can and not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[2] Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, et al. Interactive imitation learning in robotics: A survey. pages 1–197, 2022. 2

[3] 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*, 2024. 3, 4

[4] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proc. Robotics: Science and Systems (RSS)*, 2024. 2, 4

[5] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018. 2

[6] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conf. on Robot Learning (CoRL)*, 2024. 2

[7] Soroush Nasiriany, Homer Walke, Lerrel Shi, Antonio Loquercio, and Sergey Levine. RT-Affordance: Affordances are versatile intermediate representations for robot manipulation. *arXiv preprint arXiv:2411.02704*, 2024. 2

[8] Brianna Zitkovich, Tianhe Yu, Sichun Xu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conf. on Robot Learning (CoRL)*, 2023. 1

Appendix

Table 1 shows similarity and reward comparison between randomly generated and our approach. Table 2 shows video demonstration duration comparison. Table 3 shows training time comparison across tasks. Table 4 shows full convergence metrics for the pick-and-place diffusion model trained to epoch 102 (2776 global steps), confirming the model's capacity to fully learn a subtask policy given sufficient training. Table 5 shows a sample of the raw camera trajectory data extracted from UMI video, which serves as the direct input to the diffusion model.

Table 1. Similarity Comparison (50 responses each)

Method	$\text{Sim}(s, \theta_p)$	$\text{Sim}(s, \theta_{np})$	**Reward**
Random Generated	0.122	0.110	0.028
Our Approach	0.182	0.147	0.047

Table 2. Video Data Length Comparison

Task	**Duration**
long-horizon task (full sequence)	>40 s
Subtask 1 (pour coffee)	∼10 s
Subtask 2 (clean cup)	∼17 s

Table 3. Training Time Comparison (7 videos, 2 epochs)

Task	**Training Time**
long-horizon task (full sequence)	16 h 57 m 5 s
Subtask 1 (pour coffee)	1 h 40 m 17 s
Subtask 2 (clean cup)	45 m 53 s

Table 4. Full Convergence Metrics: Pick-and-Place Diffusion Model (Epoch 102)

Metric	**Value**
Final epoch	102
Global steps	2776
Learning rate	9×10^{-5}
train_loss	0.02991
train_action_mse_error	0.00060
train_action_mse_error_pos	0.00019
train_action_mse_error_rot	0.00090
val_action_mse_error	0.00275
val_action_mse_error_pos	0.00082
val_action_mse_error_rot	0.00418

Table 5. Camera Trajectory Data Sample (First 5 rows)

t (s)	**x**	**y**	**z**	q_x	q_y	q_z
0.000	0.2977	0.0215	−0.0055	0.0848	0.0160	−0.1014
0.017	0.2967	0.0217	−0.0056	−0.0849	−0.0164	0.1019
0.033	0.2917	0.0225	−0.0075	−0.0847	−0.0168	0.1021
0.050	0.2916	0.0225	−0.0073	−0.0849	−0.0166	0.1025
0.067	0.2914	0.0224	−0.0071	−0.0849	−0.0162	0.1029