Pre-Manipulation Alignment Prediction for Open-Vocabulary Object Manipulation Based on End-Effector Trajectories

Motonari Kambara Komei Sugiura
Keio University
Yokohama, Japan
{motonari.k714, komei.sugiura}@keio.jp

Abstract

This study addresses a task designed to predict the alignment between a natural language instruction, a pre-manipulation image, and an end-effector trajectory. Conventional methods typically perform a success prediction only after the manipulation is executed, limiting their efficiency in executing the entire task sequence. We propose a novel approach that enables the prediction of success or failure by aligning the given trajectories and images with natural language instructions. We introduce Trajectory Encoder to apply learnable weighting to the input trajectories, allowing the model to consider temporal dynamics and interactions between objects and the end effector, improving the model's ability to predict manipulation outcomes accurately. We constructed a dataset based on the RT-1 dataset, a large-scale benchmark for open-vocabulary object manipulation tasks, to evaluate our method. The experimental results show that our method achieved a higher accuracy than the baselines.

1 Introduction

Object manipulation is essential in fields, such as household tasks [1] and agriculture [2, 3]. Gaining insight into the success of this manipulation can improve its efficiency and safety because it avoids potential dangers caused by task failures and the unnecessary execution of tasks following object manipulation failures.

This study focuses on predicting the success of open-vocabulary object manipulations. The task involves predicting the success of the manipulations based on a given end effector's trajectory, an egocentric view of the image before the manipulation, and a natural language instruction sentence. These task functions are challenging because they require considering future interactions between objects and the end effector based on the generated trajectory and their alignment with the natural language instructions.

A typical example of the task is depicted in Fig. 1. In this example, the instruction "Pick an apple from the white bowl." is given. Given that the manipulator successfully grasps the apple from the white bowl, the model should predict a successful object manipulation.

Most existing methods related to manipulation success prediction perform the success/failure prediction after the manipulation execution. A more desirable approach would be to validate planned trajectories a priori by assessing their consistency with the task context, which includes both visual observations and natural language instructions.

Figure 1. Task Example. "Pick an apple from the white bowl." is provided as a natural language instruction. In this example, the model is expected to predict 'Aligned' because the object manipulation is performed appropriately.

While Vision-Language-Action (VLA) models [4–6] are designed to address this limitation by performing alignment between visual observations, language instructions, and planned trajectories, they often fail to achieve sufficient alignment in practice. Indeed, for instance, OpenVLA [6], a state-of-the-art VLA model, achieved almost 0% success rate in zero-shot manipulation tasks [7].

This study proposes a model for pre-manipulation alignment prediction between natural language instructions, egocentric images, and planned trajectories. We introduce Trajectory Encoder to apply weighting to the trajectory using learnable parameters. This alignment enables the interactions between objects and the end effector to be determined from images and trajectories. Consequently, unlike many existing methods, our approach enables predicting alignment between a natural language instruction, an image, and a planned trajectory before the manipulation execution.

The contributions of this research are as follows:

- We propose a model that predicts whether the task specified by an instruction sentence can be appropriately executed by aligning the trajectory with the pre-manipulation image.
- We introduce Trajectory Encoder to apply weighting to the trajectory using learnable parameters.

2 Related Work

Collision prediction during object manipulation is highly relevant to our work. For instance, post-collision decision strategies have been proposed (e.g.,

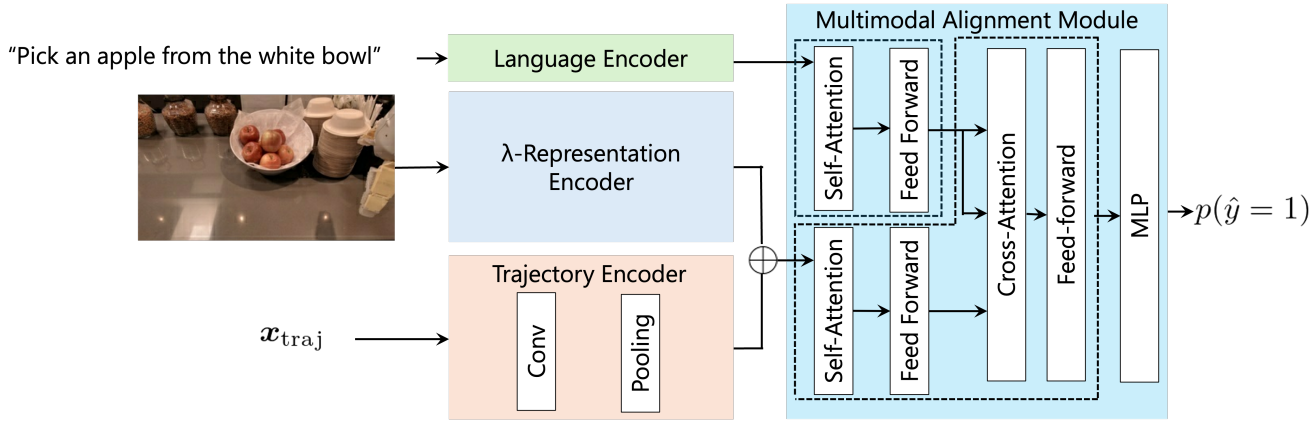

Figure 2. Network overview of the proposed method. 'Conv' and 'MLP' represent the convolutional layer and multi-layer perceptron, respectively.

[8]). Moreover, several methods predict collisions using images and placement policies [9–11]. Our method differs from these approaches by considering factors beyond collisions contributing to task failure. Liu et al. proposed a task failure prediction model [12]. They proposed Failure Classifier to detect erroneous trajectory generation based on images and predicted trajectories. However, this model cannot process natural language instructions, making it unsuitable for direct application to the task addressed in this paper.

The proposed method is also closely related to subtask planning methods in long-horizon tasks [5, 13, 14]. Some approaches determine the success or failure of a task after executing a subtask and replan subsequent subtasks based on that assessment [14–16]. These methods are similar to the proposed method because they evaluate task success. For example, REFLECT [17] verifies whether predefined states are achieved based on target states for each object class. However, unlike these methods that only detect failures during object manipulation, our proposed method predicts the alignment of inputs in a manipulation task without relying on predefined target states.

Our task shares significant similarities with the VQA task [18], which takes an image and a natural-language question as input and outputs an appropriate text-based answer. Existing VQA models exhibit sophisticated spatial understanding, reasoning capabilities, and the ability to follow instructions, thereby achieving favorable performance across a wide range of domains [19]. However, as shown in Sec. 5.2, even state-of-the-art VQA models find our task challenging, likely because it requires more advanced spatial reasoning for the appropriate alignment of the trajectory with the image. To address this, our proposed method introduces a cross-attention mechanism that explicitly accounts for the alignment between the trajectory and the image.

3 Problem Statement

This study focuses on Pre-Manipulation Alignment Prediction (PMAP) for open-vocabulary object manipulation. This task involves verifying the alignment between a natural language instruction sentence, a pre-manipulation egocentric image, and a planned trajectory. Given the pre-manipulation image, the trajectory, and the instruction sentence, the model should determine whether they are mutually aligned, indicating a likely successful manipulation. Fig. 1 illustrates a typical example of the PMAP task.

In this episode, the natural language instruction is given as "Pick an apple from the white bowl." In this example, models should predict 'Aligned' if the manipulator picks the apple successfully, and 'Hallucinated' if it falls the apple.

In the PMAP task, the input consists of an egocentric image, the trajectory of the end effector, and a natural language instruction sentence. The expected output is the predicted probability $p(\hat{y}=1)$. y and $\hat{y}$ denote a label and predicted label, respectively. The condition $y=1$ indicates that the planned trajectory, the egocentric image, and the given natural language instruction are aligned. The input image is exclusively the one captured just before the manipulation. Consequently, the model is required to infer the final state of the manipulation from the trajectory data and this single pre-manipulation image, making the task particularly challenging. In this study, we assume that trajectory generation is out of scope.

4 Proposed Method

Fig. 2 shows an overview of the proposed method. The proposed method consists of Trajectory Encoder and λ-Representation Encoder. The input to the model is denoted as $\boldsymbol{x} = \{\boldsymbol{x}_{\text{txt}}, \boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{traj}}\}$, where $\boldsymbol{x}_{\text{txt}}$, $\boldsymbol{x}_{\text{img}}$, and $\boldsymbol{x}_{\text{traj}}$ are the egocentric view image before object manipulation, the natural language instruction, and the end effector trajectory, respectively. $\boldsymbol{x}_{traj}$ has a time series of length T. In the proposed method, the

language feature $\boldsymbol{h}_{\text{txt}}$ is first obtained from $\boldsymbol{x}_{txt}$.

4.1 Trajectory Encoder

We introduce Trajectory Encoder, which is designed to filter the trajectory using learnable weights. This filtering process enables temporal trajectory compression for more efficient representation and processing.

The input to this module is $\boldsymbol{x}_{\text{traj}} \in \mathbb{R}^{D \times T}$, and the output is $\boldsymbol{h}_{\text{traj}}$, where D denotes the number of degrees of freedom of the manipulator. This module is composed predominantly of convolutional and pooling layers.

First, we obtain $\boldsymbol{x}_{\text{conv}} \in \mathbb{R}^{D \times T}$ by applying a convolutional layer along the time dimension, capturing temporal patterns. The convolution focuses on modeling temporal dependencies in the trajectory data. We then apply a pooling layer along the time dimension of $\boldsymbol{x}_{\text{conv}}$, reducing it from T to d_{trm}, resulting in a processed tensor $\boldsymbol{h}_{\text{traj}} \in \mathbb{R}^{D \times d_{\text{trm}}}$. This approach efficiently models the temporal dynamics of trajectories.

4.2 λ-Representation Encoder

Solving the PMAP task requires understanding the positional information of each object in the input image to predict whether the trajectory is interacting with the appropriate object.

In the PMAP task, models are required to predict the success or failure of object manipulation based on open-vocabulary manipulation instructions, images, and trajectories. Thus, it is essential to ensure proper alignment with natural language to predict whether the manipulation instruction can be executed based on the images.

Therefore, we employ the λ-representation [20] as a visual feature aligned with natural language. This representation has been reported to be more effective than visual features obtained from the image encoder of CLIP, which is also aligned with natural language.

This module extracts the λ-representation $\boldsymbol{h}_{\lambda}$ from the input image. The input to this module is $\boldsymbol{x}_{\text{img}}$, and the output is $\boldsymbol{h}_{\lambda}$. The process in this module is based on the λ-Representation Encoder module [20]. In contrast, for Narrative Representation, we use GPT-4o to generate a description about $\boldsymbol{x}_{\text{img}}$.

We then compute cross-attention between $[\boldsymbol{h}_{\lambda}; \boldsymbol{h}_{\text{traj}}]$ and $\boldsymbol{h}_{\text{txt}}$ using the N-layer transformer encoder-decoder. Finally, we obtain the predicted probability $p(\hat{y})$ of successful object manipulation.

5 Experiments

5.1 Experimental Settings

We constructed a dataset by extending the SP-RT-1 dataset [20]. The SP-RT-1 dataset was built based on the RT-1 dataset [5]. The SP-RT-1 dataset includes images before and after object manipulation, object manipulation instruction sentences, and 'Aligned'/'Hallucinated' labels for each episode of object manipulation. In contrast, the PMAP task requires a dataset that includes egocentric view images before object manipulation, end effector trajectories, natural language instructions, and task outcome labels indicating 'Aligned' or 'Hallucinated'. Therefore, because the SP-RT-1 dataset generally contains the information required for the PMAP task, we constructed our dataset based on the SP-RT-1 dataset.

Method	Accuracy [%]
Qwen2-VL [19]	52.1 ± 0
GPT-4o [21]	69.4 ± 0.64
Contrastive λ-Repformer [20]	77.7 ± 0.65
Ours w/o Traj. Enc.	83.2 ± 0.48
Ours	**83.4** ± 0.65

Table 1. Quantitative results. 'Ours w/o Traj. Enc.' refers to the proposed method where Trajectory Encoder is replaced with a linear function. Bold indicates the accuracy with the highest value.

The end effector trajectories in each episode were lacking when using the SP-RT-1 dataset for the PMAP task. We adapted the trajectories included in the RT-1 dataset for the PMAP task by collecting the trajectories. We split the dataset as follows in [20].

The constructed dataset includes 13,915 samples, split into 11,915 training samples, 1,000 validation samples, and 1,000 test samples. The manipulator used for data collection had a seven-dimensional action space [5]. An additional dimension was related to the opening and closing of the end effector. Therefore, the trajectory at each timestep has eight dimensions.

We used an NVIDIA GeForce RTX 4090 with 24GB of VRAM, 64GB of RAM, and an Intel Core i9–13900KF. The training time for the proposed method was approximately 1.5 hours, and the inference time, excluding feature extraction, was 0.78 ms per sample.

5.2 Quantitative Results

Table 1 presents the quantitative results, including the means and standard deviations. We conducted the experiments five times. In our experiments, In our experiments, we used GPT-4o [21], Qwen2-VL [19], and Contrastive λ-Repformer [20] as the baseline methods. Accuracy was used as the evaluation metric.

We selected these baseline methods for the following reasons: GPT-4o is a representative multimodal large language model that has been pre-trained on large-scale datasets and has exhibited remarkable performance in a variety of tasks, including VQA. Additionally, Qwen2-VL is a state-of-the-art VQA model. We chose Contrastive λ-Repformer because it has demonstrated remarkable results on the Success Prediction for Open-vocabulary Manipulation (SPOM) task [20], which is closely related to the PMAP task.

Contrastive λ-Repformer was trained on our dataset, whereas experiments for other baseline models were conducted in zero/few-shot settings. Post-manipulation images were not provided to Contrastive

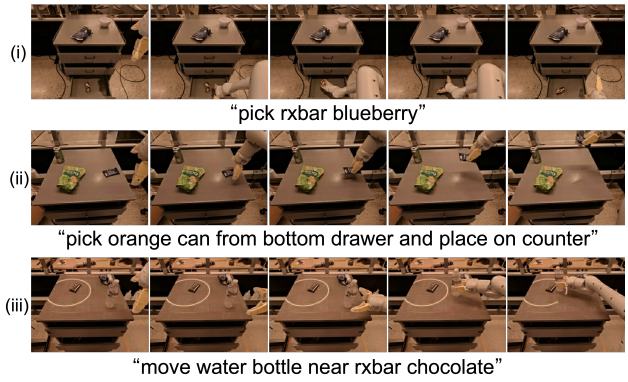

Figure 3. Qualitative results. Panels (i), (ii), and (iii) represent the True Positive, True Negative, and False Negative examples, respectively. The leftmost image in each panel illustrates the scene before manipulation.

λ-Repformer to prevent leakage of the object manipulation outcomes. These baselines cannot directly accept trajectories as inputs; therefore, we used images with the trajectory overlaid onto the pre-manipulation image as the input. For GPT-4o, two samples were included in the prompt as examples. That is, GPT-4o was used in a few-shot setting. On the other hand, since Qwen2-VL has difficulty handling multiple images as input, no sample examples were included in the prompt.

The baseline methods Qwen2-VL, GPT-4o, and Contrastive λ-Repformer achieved accuracies of 52.1%, 69.4%, and 77.7%, respectively. In contrast, our proposed method attained an accuracy of 83.4%, outperformed the best baseline method, Contrastive λ-Repformer, by 5.7 points. We attribute these results to the proposed method's ability to predict future situations based on trajectory information more effectively than the other methods.

5.3 Qualitative Results

Fig. 3 illustrates the qualitative results. Panels (i), (ii), and (iii) are examples of True Positive, True Negative, and False Negative, respectively.

In Fig. 3(i), the instruction was "Pick rxbar blueberry." and the manipulator successfully grasped the protein bar. Therefore, the label for this episode was 'Aligned'. The proposed method correctly predicted 'Aligned', whereas the baseline method incorrectly predicted 'Hallucinated'.

In the episode depicted in Fig. 3(ii), the instruction was "Pick orange can from bottom drawer and place on counter." However, the manipulator failed to grasp the can and could not retrieve it from the drawer. Therefore, the label for this episode was 'Hallucinated'. The proposed method correctly predicted 'Hallucinated' for this episode. The baseline method incorrectly predicted 'Aligned'. These results indicate that the proposed method could appropriately predict the alignment of object manipulation by considering the end effector's trajectory and the positions of objects within the image.

In contrast, Fig. 3(iii) illustrates an example where the proposed method made an incorrect prediction. In this episode, the instruction was "Move water bottle near rxbar chocolate." The manipulator moved the chocolate bar close to the water bottle, so the label was 'Aligned'. However, the proposed method predicted 'Hallucinated'. This episode was likely difficult because it necessitated accurately interpreting the spatial positions of multiple objects and aligning the trajectory with those positions.

5.4 Ablation Study

As an ablation study, we investigated the effectiveness of performance when Trajectory Encoder was removed. Table 1 also presents the results. In the table, 'Ours w/o Traj. Enc.' represents the model where Trajectory Encoder was replaced with a linear function. The accuracies of Models (i) and (ii) were 83.2% and 83.4%, indicating a 0.2-point improvement using Trajectory Encoder. The simple-weighting approach applied within Trajectory Encoder produced a slight improvement.

6 Conclusions

In this study, we focused on a task to predict the alignment between natural language instructions, egocentric view images before manipulation, and end effector trajectories.

Our contributions are as follows:

- We proposed a model that predicts whether an instruction sentence, a trajectory, and a pre-manipulation image are appropriately aligned or not.
- We introduced Trajectory Encoder, which applies weighting to the trajectory using learnable parameters.
- The experimental results demonstrated that our method achieved higher prediction accuracy than the baseline method.

As a potential future direction, it is conceivable to combine the proposed approach with VLA models [4–6]. As discussed in Sec. 1, many current VLA models exhibit insufficient alignment between modalities, which can lead to the generation of inappropriate trajectories. Moreover, in most cases, those models lack mechanisms to evaluate the validity of the trajectories they generate, making it challenging to prevent the execution of erroneous trajectories. By applying the proposed method to these methods, it becomes possible to select and execute the suitable trajectory from a set of candidates generated by VLA models, thereby enabling more effective execution of object manipulation tasks.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number 23H03478, JST Moonshot, and JSPS Fellows Grant Number JP23KJ1917.

References

[1] J. Wu, R. Antonova, *et al.*, "Tidybot: Personalized Robot Assistance with Large Language Models," *Autonomous Robots*, vol. 47, no. 8, pp. 1087–1102, 2023.

[2] C. Lehnert, A. English, C. McCool, *et al.*, "Autonomous Sweet Pepper Harvesting for Protected Cropping Systems," *IEEE RA-L*, vol. 2, no. 2, pp. 872–879, 2017.

[3] J. Jun, J. Kim, J. Seol, J. Kim, and H. Son, "Towards an Efficient Tomato Harvesting Robot: 3D Perception, Manipulation, and End-Effector," *IEEE Access*, vol. 9, pp. 17 631–17 640, 2021.

[4] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An Open-Source Generalist Robot Policy," in *RSS*, 2024.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, *et al.*, "RT-1: Robotics Transformer for Real-World Control at Scale," *arXiv preprint arXiv:2212.06817*, 2022.

[6] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "OpenVLA: An Open-Source Vision-Language-Action Model," in *CoRL*, 2024.

[7] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," *arXiv preprint arXiv:2410.24164*, 2024.

[8] S. Haddadin, A. Luca, and A. Albu-Schäffer, "Robot Collisions: A Survey on Detection, Isolation, and Identification," *T-RO*, vol. 33, no. 6, pp. 1292–1312, 2017.

[9] R. Mottaghi, M. Rastegari, A. Gupta, and A. Farhadi, ""What Happens If..." Learning to Predict the Effect of Forces in Images," in *ECCV*, 2016, pp. 269–285.

[10] A. Magassouba, K. Sugiura, A. Nakayama, *et al.*, "Predicting and Attending to Damaging Collisions for Placing Everyday Objects in Photo-Realistic Simulations," *Advanced Robotics*, vol. 35, no. 12, pp. 787–799, 2021.

[11] M. Kambara and K. Sugiura, "Relational Future Captioning Model for Explaining Likely Collisions in Daily Tasks," in *ICIP*, 2022, pp. 2601–2605.

[12] H. Liu, S. Dass, R. Martín-Martín, and Y. Zhu, "Model-Based Runtime Monitoring with Interactive Imitation Learning," in *ICRA*, 2024, pp. 4154–4161.

[13] S. Ishikawa and K. Sugiura, "Moment-Based Adversarial Training for Embodied Language Comprehension," in *ICPR*, 2022, pp. 4139–4145.

[14] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, *et al.*, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *CoRL*, 2023, pp. 287–318.

[15] M. Shirasaka, T. Matsushima, S. Tsunashima, Y. Ikeda, *et al.*, "Self-Recovery Prompting: Promptable General Purpose Service Robot System with Foundation Models and Self-Recovery," in *ICRA*, 2024.

[16] D. Driess, F. Xia, M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, *et al.*, "PaLM-E: An Embodied Multimodal Language Model," in *ICML*, 2023, pp. 8469–8488.

[17] Z. Liu, A. Bahety, and S. Song, "REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction," in *CoRL*, 2023, pp. 3468–3484.

[18] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *ICCV*, 2015, pp. 2425–2433.

[19] P. Wang, S. Bai, *et al.*, "Qwen2-VL: Enhancing Vision-Language Model's Perception of The World at Any Resolution," *arXiv preprint arXiv:2409.12191*, 2024.

[20] M. Goko, M. Kambara, *et al.*, "Task Success Prediction for Open-Vocabulary Manipulation Based on Multi-Level Aligned Representations," in *CoRL*, 2024.

[21] OpenAI. (Accessed: Nov. 2024.) GPT-4o. [Online]. Available: https://platform.openai.com/docs/models/gpt-4o