

Appendix

A Details of Experimental Setup

A.1 SP-RT-1 Dataset

As described in Sec. 4.1, we constructed the SP-RT-1 dataset from the RT-1 dataset [1] for our task. The details are described below. We collected the first and last images of each episode. The dataset was preprocessed by modifying the instruction sentences. In the RT-1 dataset, 43.6% of the negative samples were incorrectly labeled as negative, despite the manipulator having successfully executed the manipulation. We replaced the instruction sentences for the incorrectly annotated samples with alternative sentences that were randomly selected to create negative samples. This strategy was chosen instead of converting them to positive samples, because the original dataset contained fewer negative samples than positive samples, and converting negative samples to positive samples would further reduce the proportion of negative samples.

The SP-RT-1 dataset consisted of a total of 13,915 samples, with a vocabulary size of 49, a total word count of 78,790, and an average sentence length of 5.66. The dataset contains 10,000 positive samples and 3,915 negative samples. The SP-RT-1 dataset contained 11,915, 1,000, and 1,000 samples in the training, validation, and test sets, respectively. We used the training, validation, and test sets to estimate parameters, tune hyperparameters, and evaluate models, respectively. We computed the accuracy on the validation set every epoch. The performance on the test set was evaluated using the model that achieved the highest accuracy on the validation set. The dataset is publicly available at <https://contrastive-lambda-repformer.s3.amazonaws.com/dataset/SP-RT-1.tar.gz>.

Other related datasets and benchmarks. For multimodal language understanding tasks for robotics, various datasets and benchmarks are used in both real-world [2, 3, 4] and simulation [5, 6, 7, 8] settings. Among them, the RT-1 dataset is the most relevant to our target task of success prediction for object manipulation. Additionally, VLMbench [9] is a standard benchmark for object manipulation tasks on a tabletop. It provides natural language instructions, labels indicating the success or failure of each manipulation, and images captured from five camera views.

A.2 Zero-Shot Transfer Experiment

For a comprehensive evaluation, we validated the proposed method in a physical environment using a mobile manipulator with zero-shot transfer settings. We collected the data in the environment described in Sec. 4.1. In this experiment, we used a subset of the YCB objects [10], which are standard objects for manipulation research. These selections were based on their suitability for grasping by the HSR end-effector.

In the experiment, we randomly selected up to four objects and arranged them on the table. Then, executable open-vocabulary instruction sentences were created and assigned to the episodes. The manipulations were performed by remote controlling the robot. The images of the scene before and after the manipulations were taken using the head-mounted camera of the robot. In total, 112 episodes were collected, with 56 episodes for both positive and negative samples. The dataset is also available at <https://contrastive-lambda-repformer.s3.amazonaws.com/dataset/zero-shot.tar.gz>.

A.3 Implementation Detail

Table A1 shows the experimental settings for the proposed method. Our model had approximately 64M trainable parameters and 7.25G multiply-add operations. We trained our model on a GeForce RTX 4090 with 24 GB of GPU memory and an Intel Core i9-13900KF with 64 GB of RAM. It took approximately 1.5 hours to train our model on the SP-RT-1 dataset. The inference time was approximately 1.6 ms/sample.

In Narrative Representation Module in λ -Representation Encoder, we used following prompt to generate descriptions: “Give a clear, comprehensive and detailed description of the state of the objects shown in this image. For each object, mention their colors, sizes, shapes, how they are placed (upright, etc.), position within the image and relative position to other objects. Begin with the phrase ‘In the image,’. Only use information that can be gained from the image. Mention the objects that appear in the sentence string below. If the objects in the sentence string are not present in the image, mention that they are not present. Sentence string: ‘instruction’.” Here, we inserted the instruction sentence for each episode into ‘instruction’.

Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Learning rate	1.0×10^{-6}
Weight decay	1.0×10^{-1}
Batch size	32
Epoch	150

Table A1: Experimental settings for Contrastive λ -Repformer.

A.4 Baselines

For comparative experiments, five baseline methods were used. We used the following experimental settings for each baseline. For each multimodal large language model (MLLM)-based method—InstructBLIP [11], Gemini [12], GPT-4V [13]—, we tested more than ten prompts and adopted the one with the best results.

UNITER-base/large [14]. We performed fine-tuning according to the hyperparameter settings described in [14].

InstructBLIP. InstructBLIP assumes a single image as the image input. Therefore, we concatenated x_{before} and x_{after} as shown in Fig. A1, handling them as a single input image. The prompt used is as follows: “These two images show the robot executing the instruction ‘instruction’. Based on them, please predict whether the robot has successfully completed the task and answer with ‘success’ or ‘failure’.” Here, we inserted the instruction sentence for each episode into ‘instruction’. This approach was applied similarly across all MLLM-based model prompts.



Figure A1: An example of the image input to InstructBLIP. The left and right parts show the images before and after manipulation, respectively.

Gemini. Gemini is capable of handling multiple images as input [12]. Therefore, during inference, we provided x_{before} , x_{after} , and the following prompt as input: “These images show the robot executing the instruction ‘instruction’. The first image shows the scene before the object manipulation by the robot and the second image shows the scene after. Based on the two images and the instruction, determine whether the robot has successfully completed the task and answer with ‘true’ or ‘false’.”

GPT-4V. Similarly, GPT-4V can also process multiple images [13]. Thus, in the experiments, we inputted x_{before} , x_{after} , and the following prompt: “These images, taken from a single viewpoint camera, show the robot executing the instruction ‘instruction’. Based on these images and the instruction, please determine whether the robot has successfully completed the task and answer with ‘true’ or ‘false’.”

B Additional Ablation Study

We conducted an additional ablation study to investigate the contribution of the cross-attention operation in Contrastive λ -

Model	Attention Mechanism	Accuracy [%]
(i)	Self-Attention	78.88 ± 1.05
(ii)	Cross-Attention	80.80 ± 0.86

Table A2: Results of additional ablation study. Bold indicates the highest value.

Representation Decoder. This operation was used to create a representation of the difference between two λ -Representations. Table A2 presents the results.

In this experiment, we changed the cross-attention operation to a self-attention operation to investigate its contributions. From the table, it can be observed that the accuracy of Model (i) was 78.88%, which was 1.92 points lower than that of Model (ii). This indicates that the cross-attention operation is suitable for identifying the differences between images.

C Error Analysis

The confusion matrix of Contrastive λ -Repformer on the test set of the SP-RT-1 dataset includes 431, 114, 386, and 69 samples that are true positive, false positive, true negative, and false negative cases, respectively.

Thus, there were a total of 183 samples where the proposed method failed on the test set of the SP-RT-1 dataset. Table A3 shows the results of the error analysis, where we randomly selected 100 samples of failed cases. We classified them into the following six categories:

Multimodal Language Comprehension Error: This refers to cases where the model incorrectly interpreted visual information and instruction sentences, such as misunderstanding the target object and misinterpretation of referring expressions.	Error type	#Errors
	Multimodal Language Comprehension Error	63
	Partial Visibility	14
	Narrative Deficiency	11
	Ambiguous Instruction	8
	Erroneous Data Sample	4
	Total	100

Table A3: Error analysis on failure cases.

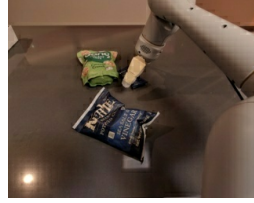
Partial Visibility: This category includes cases where the target object or area is only partially visible, making it difficult to make appropriate predictions. This can occur when the target object is more than half occluded by the manipulator or other objects, or when more than half of the target object is outside the photographed scene.

Narrative Deficiency: This addresses cases in which the narrative from the MLLM is missing.

Ambiguous Instruction: This involves cases where interpretations of success or failure may vary depending on the criteria for success. Fig. A2 shows a sample included in this category. In this example, the instruction given was “move rxbar blueberry near blue chip bag.” As shown in the figure, the ‘rxbar blueberry’ moved closer to the ‘blue chip bag’ before and after the object manipulation. However, the ground truth label for this example was false. In this case, the success or failure of the task depends on the definition of ‘near.’

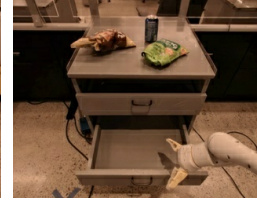
Erroneous Data Sample: This category covers cases where the input images of the sample are inadequate for the SPOM task, making it difficult to perform the task. For instance, a case where the instruction given is “pick a green can” and the manipulator is already grasping a green can in the x_{before} applies to this category.

As shown in Table A3, the main bottleneck was the Multimodal Language Comprehension Error. This issue is mainly due to the fact that the MLLM in the Narrative Representation Module generated incorrect sentences that could directly affect the success of the SPOM task. Fig. A3 shows a sample categorized as a Multimodal Language Comprehension Error. The left and right image in Fig. A3 show x_{before} and x_{after} , respectively. The captions created by the MLLM for x_{before} was “In the image, there is an open middle drawer on a metal table. Inside the drawer, there are two objects: a sandwich and a can of soda. The sandwich is upright, while the can of soda is on its side.” The captions for x_{after} was “In the image, there is an open middle drawer on a metal table. Inside the drawer, there are two objects: a sandwich and a can of soda. The sandwich is upright, while the can of soda is on its side.” and “In the image, there is an open middle drawer with a robotic arm reaching into it. The robotic arm appears to be picking up something from the drawer. Additionally,



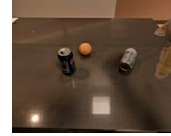
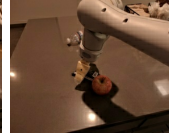
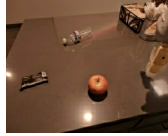
“move rxbar blueberry near blue chip bag”

Figure A2: A sample of Ambiguous Instruction. In this case, the given instruction was “move rxbar blueberry near blue chip bag.” The ground truth label was false. The success or failure of the manipulation depends on the definition of ‘near.’



“move rxbar blueberry near blue chip bag”

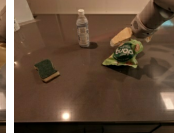
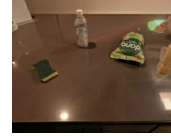
Figure A3: An example of a sample in the Multi-modal Language Comprehension Error category. The instruction for this sample was “open middle drawer.”



(i) “pick apple from white bowl”

(ii) “move rxbar chocolate near apple”

(iii) “place 7up can upright”



(iv) “knock 7up can over”

(v) “pick coke can from middle shelf of fridge”

(vi) “move green rice chip bag near sponge”

Figure A4: Additional qualitative results on the SP-RT-1 dataset. In this figure, (i)-(iii) represent true positive cases, and (iv)-(vi) are true negative cases. These are visualized in the similar manner to Fig. A2.

there is a can of soda sitting on top of the drawer.” The former caption states that the middle drawer was already open before the manipulation. This makes it difficult for the model to make appropriate predictions based on the information.

This issue may be due to the difficulty of designing prompts for large language models (LLMs). Despite experimenting with many prompts and selecting the best one, erroneous generations still occurred. Indeed, object hallucination is a known challenge in image captioning by LLMs [15]. Therefore, a possible solution could investigate prompt designs that reduce the likelihood of such errors. For example, instead of describing everything at once, several elements could be defined in advance and short responses could be obtained for each of them.

D Additional Qualitative Results

Figs. A4 and A5 provide additional success examples of Contrastive λ -Repformer on the SP-RT-1 dataset and in the zero-shot transfer experiment, respectively. For the sample shown in Fig. A4 (iii), all baseline methods except InstructBLIP made incorrect predictions. Likewise, for the sample displayed in Fig. A4 (vi), all baseline methods except UNITER-base made incorrect predictions. It was found that for episodes with only a subtle difference between the images before and after the manipulation, the baseline methods had difficulty in making accurate predictions, whereas Contrastive λ -Repformer was able to predict appropriately.

Furthermore, all MLLM-based methods except Gemini made incorrect predictions for Fig. A4 (ii), and all MLLM-based methods made incorrect predictions for Fig. A5 (ii). This indicates that even MLLM-based methods can struggle with referring expression comprehension and aligning images with natural language.

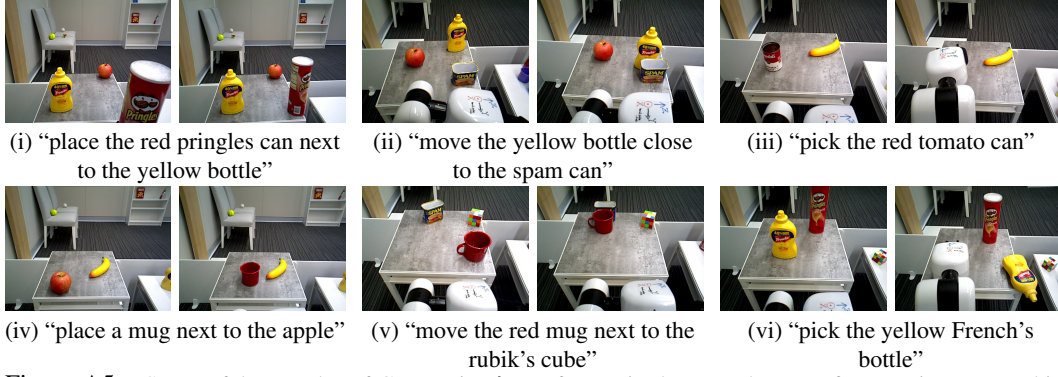


Figure A5: Successful examples of Contrastive λ -Repformer in the zero-shot transfer experiments. In this figure, examples (i)-(iii) show true positive cases, and (iv)-(vi) depict true negative cases. The examples are similarly visualized in the same manner in Fig. A2.

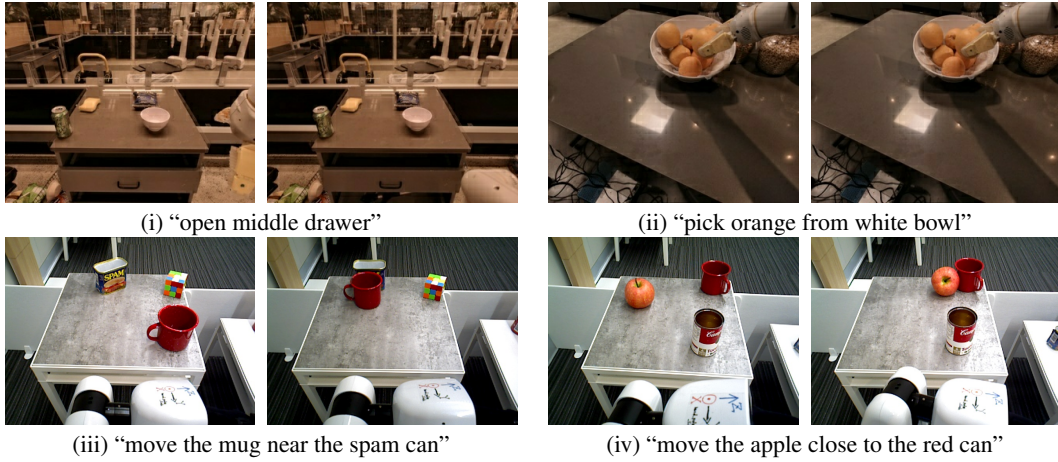


Figure A6: Failed cases of the proposed method. These are visualized in the same manner in Fig. A2 as well.

Fig. A6 shows failed cases of the proposed method. Fig. A6 (i) and (ii) show the failed examples on the SP-RT-1 dataset, and Fig. A6 (iii) and (iv) exhibit the failed examples in the zero-shot transfer experiment.

Fig. A6 (i) shows an example with the instruction of "open middle drawer." The ground truth label for this example was success, because the robot opened the middle drawer. Nonetheless, our method predicted that the robot failed in carrying out the instruction. This error can be explained by the fact that most of the middle drawer lies outside the photographed area, making it hard even for humans to deduce correctly.

The instruction for the instance displayed in Fig. A6 (ii) is "pick orange from white bowl" and the ground truth label was failure. This result is most likely because the bottom of the orange is still touching the other oranges. Meanwhile, all the baseline and proposed methods predicted success. This error arises from the ambiguity of the situation, where predictions would likely be divided even among humans.

Fig. A6 (iii) presents a failed example in the zero-shot transfer experiment. In this example, the instruction sentence was "move the mug near the spam can." This sample was labeled success, whereas Contrastive λ -Repformer predicted this sample as failure. To predict appropriately, the model needs to appropriately understand both the 'mug' and the 'spam can'. In particular, to understand 'spam', approaches such as optical character recognition are required, which makes it challenging.

Finally, Fig. A6 (iv) exhibits a failed case with the instruction of "move the apple close to the red can." Contrastive λ -Repformer predicted that the manipulator succeeded in following the instruction, while the ground truth label was failure. In this sample, there are three red objects: an apple, a red can, and a red mug. The manipulator brought the apple close to the red mug. Therefore, it is

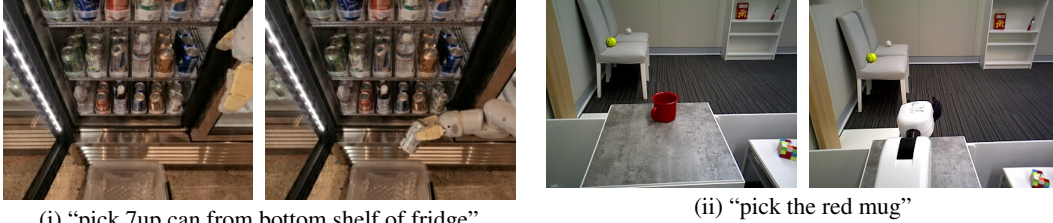


Figure A7: Samples of human errors. These are visualized in the same way in Fig. A2.

possible that the model judged the success of the manipulation based solely on the characteristic of being 'red'.

E Human Errors in Subject Experiment

Fig. A7 depicts examples where the human predictions were incorrect. In Fig. A7 (i), the instruction sentence for this sample was "pick 7up can from bottom shelf of fridge." Although the ground truth for this sample was success, the human prediction was failure. In this example, it is difficult to identify the label of the can that the manipulator grasped, as well as to determine where the can was retrieved from.

In Fig. A7 (ii), "pick the red mug" was the instruction. In this example, the mug was successfully grasped by the manipulator. However, the mug was mostly occluded, making it difficult to judge. As shown in the example, the SPOM task can be difficult even for humans.

References

- [1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, et al. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv preprint arXiv:2212.06817*, 2022.
- [2] Y. Qi, Q. Wu, P. Anderson, X. Wang, Y. Wang, C. Shen, and A. Hengel. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments. In *CVPR*, pages 9982–9991, 2020.
- [3] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. DROID: A Large-Scale In-the-Wild Robot Manipulation Dataset. In *RSS*, 2024.
- [4] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *ICRA*, 2024.
- [5] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, et al. BEHAVIOR-1K: A Benchmark for Embodied AI with 1,000 Everyday Activities and Realistic Simulation. In *CoRL*, pages 80–93, 2023.
- [6] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. Chang, L. Guibas, and H. Su. SAPIEN: A SimULATED Part-based Interactive ENvironment. In *CVPR*, pages 11097–11107, 2020.
- [7] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *ICLR*, 2022.
- [8] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning. In *CoRL*, pages 1025–1037, 2020.
- [9] K. Zheng, X. Chen, C. Jenkins, and X. Wang. VLMbench: A Benchmark for Vision-and-Language Manipulation. *NeurIPS*, 35:665–678, 2022.
- [10] B. Calli, A. Walsman, A. Singh, S. Srinivasa, et al. Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set. *IEEE RAM*, 22(3):36–52, 2015.
- [11] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In *NeurIPS*, 2023.
- [12] G. GeminiTeam, R. Anil, S. Borgeaud, Y. Wu, B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*, 2023.
- [13] J. Achiam, S. Adler, S. Agarwal, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.
- [14] Y. Chen, L. Li, L. Yu, E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, pages 104–120, 2020.
- [15] P. Manakul, A. Liusie, and M. Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In *EMNLP*, pages 9004–9017, 2023.