19th International Conference on Intelligent Autonomous Systems (IAS), Genoa, Italy, 2025.

LIAM: Multimodal Transformer for Language Instructions, Images, Actions and Semantic Maps

Yihao Wang, Raphael Memmesheimer, Sven Behnke

Autonomous Intelligent Systems, Computer Science Institute VI, Center for Robotics,
University of Bonn, Germany
https://ais.uni-bonn.de

Abstract. The availability of large language models and open-vocabulary object perception methods enables more flexibility for domestic service robots. The large variability of domestic tasks can be addressed without implementing each task individually by providing the robot with a task description along with appropriate environment information. In this work, we propose LIAM — an end-to-end model that predicts action transcripts based on language, image, action, and map inputs. Language and image inputs are encoded with a CLIP backbone, for which we designed two pre-training tasks to fine-tune its weights and pre-align the latent spaces. We evaluate our method on the ALFRED dataset, a simulator-generated benchmark for domestic tasks. Our results demonstrate the importance of pre-aligning embedding spaces from different modalities and the efficacy of incorporating semantic maps.

1 Introduction

With the rapid evolution of deep learning research, particularly in the natural language domain, we have witnessed the emergence of enormous Transformer-based models capable of generating high-quality texts and synthetic images [1, 2]. They hold much potential for open-ended robotic applications, like domestic service tasks, but cannot be directly applied because robotics involves additional modalities, such as images and actions.

In 2021, OpenAI released the foundation model CLIP, which uses a contrastive learning paradigm to connect text and images [3]. Representations of textual and visual inputs are learned by contrasting similar and dissimilar pairs. CLIP features are suitable for open-vocabulary zero-shot image categorization.

Many robotics applications soon utilized CLIP-based models for different open-vocabulary tasks [4]. Vision-Language-Action models like OpenVLA [5] predict the next action based on visual inputs. Research in models predicting entire action transcript sequences based on language instructions and visual inputs is still in its early phase. Training a model that understands all relevant modalities and has strong generalizability is challenging and requires a considerable amount of data.

In this work, we introduce the end-to-end model LIAM, which receives inputs from the modalities **L**anguage, **I**mages, **A**ctions, and **M**aps and predicts action sequences. Two CLIP-like self-supervised training paradigms are designed to pre-align two or more different embedding spaces during the pre-training stage, enhancing the model's ability to understand and process multimodal inputs. We utilized semantic maps as an additional modality directly in addition to textual and visual input, where the map modality is designed to serve as a knowledge base for the model. We evaluate our model with the ALFRED Challenge [6], a benchmark task that requires a robot to understand and execute complex natural language instructions in a simulated environment. The code and pre-trained models to reproduce the results are made publicly available: `https://github.com/AIS-Bonn/LIAM`.

2 Related Work

Open-vocabulary approaches have garnered significant attention in the field of computer vision. In the past decade, most vision downstream tasks were trained via datasets with a fixed number of classes. The CLIP model, with its contrastive learning paradigm, brings a promising solution for inferencing open vocabulary vision tasks. After embedding the text and image using separate encoders, CLIP is trained by maximizing the similarity score of the representation of corresponding image and text pairs [3].

A massive amount of work is built on CLIP models for different downstreaming tasks, e.g., object detection [7, 8], semantic segmentation [9, 10], action prediction [11, 12]. These approaches have designed similar methods adapted to their specific vision downstream tasks. For example, GroupViT first uses a group-based approach for semantic segmentation by introducing learnable group tokens to the ViT architecture ([10]). During inference time, each group is assigned the semantic label with the highest similarity score representing the text input.
To tackle the Vision-Language Navigation (VLN) and Vision-Language Manipulation (VLM) tasks, there are two main approaches. I. An End-to-End model trains a model that can accept all modal inputs simultaneously and make predictions of current action for the agent [6, 13]. For example, Episodic Transformer (E.T.) is an end-to-end attention-based transformer model that consists of three modality-specific encoders and one multi-modality fusion encoder [13]. II. Modular Methods propose a pipeline of learned modules for tackling the challenge of understanding the environment and the tasks [14, 15]. FILM proposes a language processing module, which converts the task description into an action transcript using pre-defined templates. A semantic mapping module processes egocentric RGB images into a 2D semantic map. Furthermore, a semantic search policy predicts the possible distribution of objects' locations. A deterministic policy is designed to predict the action decision based on the predicted action transcript and the predicted semantic map.

Most end-to-end models for VLM tasks primarily use language and images as inputs. In this paper, we also generate a semantic map following the FILM approach and incorporate it as an additional modality input.

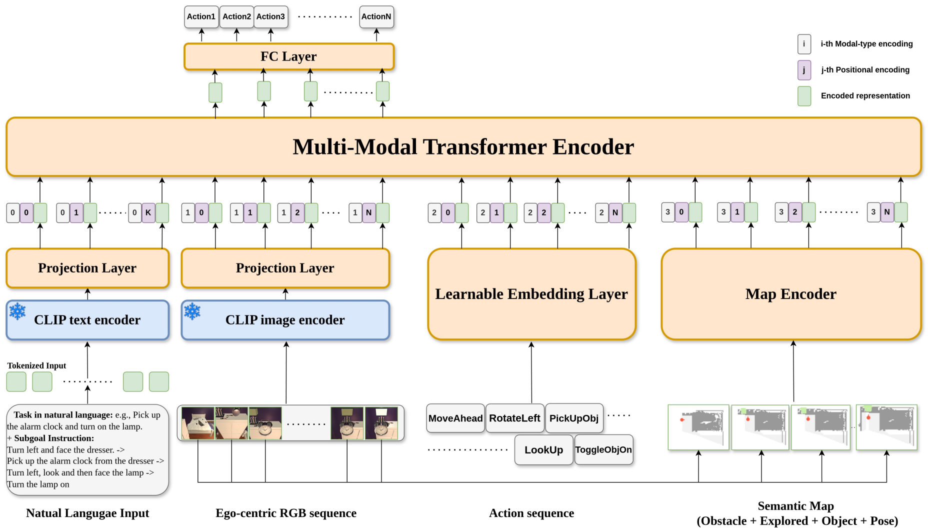

Fig. 1. Model architecture of LIAM. The blue blocks are all layers that were frozen during the end-to-end model training; the orange blocks are the parts that were trained.

3 Main Approach

We now introduce the LIAM model, including the pre-training stage and the action generation. An overview of the model is given in Figure 1. The model architecture is adopted from the Episodic Transformer [13], but we replace the image and text encoders with CLIP encoders and employ a semantic map [15]. To manage the high computational cost, we freeze the weights and biases of the CLIP backbone during end-to-end model training, ensuring efficient resource utilization.

All experiments are conducted using ALFRED (Action Learning From Realistic Environments and Directives) [6], a recent simulator generated benchmark for learning a mapping task, from vision (egocentric RGB image) and natural language input to an action transcript for a domestic service robot. ALFRED's challenge provides the following annotations: an initial state of the simulated room, language instructions, and an expert demonstration trajectory. Language instructions are provided as global instructions, e.g., "Pick up the alarm clock and turn on the lamp. <<goal>>" and a list of sentences for sub-goal instructions. The annotation consists of a sequence of discrete actions, the object's mask, whenever the interaction with objects is involved.

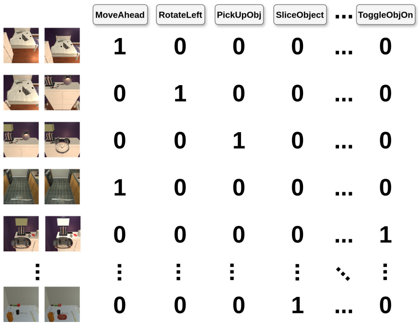

Fig. 2. An example of the ground truth from one mini-batch for alignment of visual and action embedding space. One action corresponds to two consecutive frames.

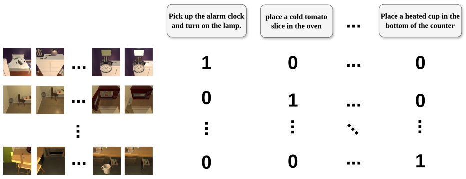
Pick up the alarm clock and turn on the lamp.
place a cold tomato slice in the oven
...
Place a heated cup in the bottom of the counter
1 0 ... 0
0 1 ... 0
0 0 ... 1

Fig. 3. Ground truth for alignment of visual and language embedding space. One frame sequence corresponds to one language instruction.

3.1 Contrastive Alignment Pre-training

CLIP was initially trained using data gathered from the Internet, resulting in bias and significant performance differences across different datasets in zero-shot inference [3]. Given that the ALFRED dataset is simulator-generated, it deviates slightly from reality. Hence, a pre-training stage to align the latent space of image and action becomes imperative.
We first pre-aligned the CLIP image encoder and action embedding via contrastive learning. The learning objective of the model is to match the correct pair of action embedding and image embedding. Unlike CLIP, where one image has one corresponding label, in our case, two consecutive images have one corresponding label, which denotes an action. Figure 2 shows an example of the ground truth of one mini-batch. Unlike the original CLIP, which forms an N × N affinity matrix (N denotes the batch size), our approach counts the number of unique actions that appear in each mini-batch instead and forms all the actions occurring as the column of the matrix. Each row denotes the representation of two consecutive frames. The matrix entry (i, j) has the value one if the i-th frame's representation corresponds to the action class j.

In our approach, we used f(·, ·), a 1D convolution layer along with global average pooling, to compute a fused representation of two consecutive images. We embedded actions with an embedding layer that embeds all 14 classes (12 classes + 2 special tokens) into 768-dimensional dense vectors. After having the normalized representations of both frames I_e and action classes A_e, we computed a cosine similarity score s of both vectors and fine-tuned the CLIP backbone following CLIP's approach. With $a_i \in A_e, I_t \in I_e$, the similarity matrix P is defined as:

$$P_{\mathrm{I2A}}(I) = \frac{\exp\Big(s\big(f(I_t, I_{t+1}), a_i\big)/\tau\Big)}{\sum\limits_{j=1}^{N-1} e^{s\big(f\left(I_t, I_{t+1}\right), a_j\big)/\tau}}, \quad P_{\mathrm{A2I}}(A) = \frac{\exp\Big(s\big(a, f(I_{t_i}, I_{t_i+1})\big)/\tau\Big)}{\sum\limits_{j=1}^{N-1} e^{s\big(a_j, f\left(I_{t_i}, I_{t_i+1}\right)\big)/\tau}}.$$

We notated text, action, and image with the letters T, A, and I. I2A denotes "image-to-action," and all later mentioned abbreviations (X2X) follow this pattern. Function f(·, ·) denotes the fusion function of two consecutive images at time step t and t + 1 as described above. τ is the temperature, a learnable parameter scales the similarity score. Following CLIP, we initialized the temperature with a value of 0.07 and use 100 as an upper threshold to clip the temperature value during the training stage, preventing the scaling factor from becoming too large [11] and 0.01 as a lower threshold bound to prevent dividing by zero error.

Following ActionCLIP [11], we used the KL divergence loss to fine-tune the CLIP backbone. The total loss of the contrastive alignment pre-training is the average of both image-to-action and action-to-image loss:

$$\mathcal{L}_{\text{Image-Action}} = \frac{1}{2}\mathbb{E}_{(x,y)\backsim\mathcal{D}}\Big[KL\big(P_{\mathrm{I2A}}(I), Q_{\mathrm{I2A}}(I)\big) + KL\big(P_{\mathrm{A2I}}(A), Q_{\mathrm{A2I}}(A)\big)\Big],$$

where $Q(\cdot)$ denotes the ground truth similarity matrix.

3.2 Triple Contrastive Pre-training

Our contrastive learning pre-aligns the latent space of the vision and action embedding. However, the alignment of the language embedding space with other latent spaces is neglected. To address this, we introduced a triple contrastive pre-training stage. This method not only aligns the vision and action embedding spaces but also aligns the language embedding space with them. This mutual training ensures that the loss can be back-propagated to all encoders. In addition to Figure 2, Figure 3 shows the example of the ground truth of aligning the CLIP text encoder and image encoder. The training objective is straightforward: the representation of the language instruction and its corresponding frame sequence should be similar to each other. We first computed the representation of every single frame (N frames) using the CLIP image encoder and further processed this sequence of representations in two ways.

Considering the computing resource for the image sequence representation, we chose the parameter-free approach to compute the mean value of all CLIP embeddings of all frames to get the sequence representation. For every two consecutive frames in this list, we used the 1D convolution layer to get the representation of both frames as the same methodology described above in Section 3.1. Thus, the visual embedding space is aligned simultaneously with both text and action embedding space. As shown below, we define cross-entropy loss $\mathcal{L}_{\text{Text-Image}}$ for learning the bidirectional pairing of visual and text sequences following the original CLIP. For aligning action and image, we continued using KL divergence loss $\mathcal{L}_{\text{Image-Action}}$discussed in Section 3.1:

$$\mathcal{L}_{\text{Text-Image}} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\left(\frac{\exp(S_{ii}/\tau)}{\sum_{j=1}^{N}\exp(S_{ij}/\tau)}\right) + \sum_{j=1}^{N}\log\left(\frac{\exp(S_{jj}/\tau)}{\sum_{j=1}^{N}\exp(S_{ij}/\tau)}\right)\right),$$

where S is the similarity matrix. The total loss of triple contrastive alignment is defined as:

$$\mathcal{L}_{\text{total}} = (1-\alpha)\times\mathcal{L}_{\text{Text-Image}} + \alpha\times\mathcal{L}_{\text{Image-Action}}.$$

We introduced α as a hyperparameter for weighing both losses. This hyperparameter gives us more flexibility. We assigned ($\alpha = 0.8$) in the total loss.

3.3 Action Generation

We embedded text and visual input using independent CLIP encoders, which are now pre-aligned. The map is embedded using a trainable projection layer (FC layer + GELU). We first applied positional encoding and modal-type encoding to each representation. The learnable modal-type encoding, first introduced by VILT [16], brings extra information and clarifies different modality types to the Transformer model; it also avoids the need for explicit separation tokens like "[SEP]."
After all the representations are concatenated, we fed it further to a multimodality fusing layer (2-layer Transformer encoder) to learn a general representation. The multimodal Transformer is a comprehensive model that aims to integrate all different modalities into one global representation, namely language, visual frames, action sequence, and semantic map in our case. We use causal attention following E.T. [13]. The attention mask follows these rules: Language tokens should attend to the language itself. Visual frames are allowed to attend to all language tokens, but they can only attend to all the frames, actions, and semantic maps before the current time step t. The same rules are applied to action tokens and semantic maps. Sinusoidal positional encoding is also applied.

Ultimately, we employed a fully connected layer to predict an output sequence based on all the tokens of visual representations, which have the same number of mappings of actions so that we can train the model with the ground truth action sequence.

3.4 LIAM Pipeline

In this section, we define our input and output mathematically to explain our approach more deeply. Given the following inputs: Tokenized language instructions $[L_1, \dots L_m]$, image sequence $[I_1, ..., I_n]$, action sequence $[A_1, \dots, A_n]$, semantic map sequence $[M_1, \dots, M_n]$, where language tokens have length m, and all other modalities have length n. For an image sequence $I_1, \dots, I_n$, only n - 1 actions are involved. We append "«stop»" as the last action. Four different encoders encode all input to their latent space and output F_l, F_v, F_a and F_m as the feature representation:

$$\begin{aligned}
&\overline{F}_{mod} = F_{mod} + mod^{Type} + mod^{Pos}, \quad \text{for mod in L, I, A, M.} \\
&F^0_{total} = [\overline{F_L}; \overline{F_I}; \overline{F_A}; \overline{F_M}] \\
&\overline{F}^l_{total} = LN(MA(\overline{F}^{l-1}_{total})) + \overline{F}^{l-1}_{total}, \qquad l = 1, ..., \mathbb{L} \\
&\overline{F}^l_I = \overline{F}^d_{total}[m+1 : m+1+n] \\
&O = \overline{F}^d_I D.
\end{aligned}$$

After applying modal-type and positional encoding to each feature representation, we concatenate them into F^0_{total}. The multi-modal Transformer encoder has $\mathbb{L}$ layers, each with Multi-headed Attention (MA), Layer Normalization (LN), and residual connections. We then extract the visual part of the total representation (from index m + 1 to m + 1 + n) for action sequence prediction. The final action sequence is obtained by multiplying this sliced matrix with a trained linear layer D. The end-to-end model is trained using cross-entropy loss between the generated action sequence and the ground truth action sequence. For predicted action sequence P = $[\hat{a_1} \dots \hat{a_n}]$ and ground truth Q = $[a_1 \dots a_n]$, the cross-entropy loss is defined as:

$$\mathcal{L}_{\text{action}} = -\sum_{i=1}^{n}\sum_{c=1}^{13} a_{i,c} log(\hat{a_{i,c}}),$$

where c represents the action class out of 14 classes (12 classes + «stop» + «pad»). However, the «pad» token is masked out during the computation of the loss value.

Following E.T. [13], we also use auxiliary losses in the training. First, we predict the object class for each frame ("NoObject" is also a class). The predicted object class is a list with the same length as the action. Similary to the action, for predicted object sequence $P_o = [\hat{o_1} \dots \hat{o_n}]$ and ground truth Q = $[o_1 \dots o_n]$, the cross-entropy loss is defined as:

$$\mathcal{L}_{\text{object}} = -\sum_{i=1}^{n}\sum_{c=1}^{85} o_{i,c} log(\hat{o_{i,c}}),$$

where c represents the object class out of 85 classes (84 object classes + NoObject).

We also predict the goal progress with the initial thoughts for giving the model a better understanding of the current progress of the current trial. The ground truth Q^{gp} of the goal progress is computed by

$$Q^{gp} = \left\{ \frac{i+1}{n} \mid i \in \{0, 1, 2, \dots, n-1\} \right\},$$

Table 1. Matching accuracy of image-to-text (I2A) and text-to-image (T2I) pairs.

Models	**Validation**		**Test**	
	ACC (I2A)	ACC (T2I)	ACC (I2A)	ACC (T2I)
Contrastive alignment of image and action				
MobileCLIP S0 (Zero-shot)	7.51	-	7.32	-
MobileCLIP S0	38.34	-	43.21	-
OpenCLIP-RN50 (Zero-shot)	5.63	-	5.51	-
OpenCLIP-RN50	61.41	-	58.78	-
OpenCLIP-ViT-B-32 (Zero-shot)	5.84	-	8.26	-
OpenCLIP-ViT-B-32	**65.45**	-	**61.58**	-
Triple contrastive alignment				
MobileCLIP S0 (Zero-shot)	3.21	35.04	3.34	33.03
MobileCLIP S0	80.28	60.22	**68.97**	35.95
OpenCLIP-RN50 (Zero-shot)	3.76	31.51	4.78	34.61
OpenCLIP-RN50	70.56	32.67	65.54	**51.82**
OpenCLIP-ViT-B-32 (Zero-shot)	3.89	45.62	5.67	37.65
OpenCLIP-ViT-B-32	**88.22**	**75.43**	65.98	40.69

where n is the length of the ground truth action for the current trial. For predicted goal progress P^{gp}, we use Mean Square Error (MSE) loss, defined as:

$$\mathcal{L}_{gp} = \frac{1}{n}\sum_{i=1}^{n}(Q_i^{gp} - P_i^{gp})^2.$$

The total loss is then defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{action} + \alpha\mathcal{L}_{object} + \beta\mathcal{L}_{gp},$$

where α, β are hyperparameters for weighing the auxiliary loss. We set all weights to 0.1 in our experiments.

4 Experiments

We now present results from the pre-training stage, and report quantitative and qualitative results.

4.1 Pre-training Stage

We pick three different CLIP backbones: MobileCLIP-S0 (54.7 million parameters), OpenCLIP with Resnet-50 backbone (114.1 million parameters), and OpenCLIP with the vision transformer backbone (152.7 million parameters).

Table 1 presents the accuracy score of correct matching pairs among the mini-batches. Firstly, we can clearly see that no CLIP model can correctly match two consecutive visual images with their corresponding actions in a zero-shot manner. As discussed before, CLIP is a model that is sensitive to the dataset.

Table 2. Results for the end-to-end model predicted sequence and the ground truth.

Models	**Valid Seen**		**Valid Unseen**	
	Accuracy	F1 score	Accuracy	F1 score
Zero-shot CLIP encoder	73.00	58.64	71.87	49.31
+ Semantic Map	78.08	71.69	76.49	59.54
Contrastive-aligned CLIP	76.97	64.17	79.56	44.02
+ Semantic Map	**79.30**	56.94	**81.51**	39.52
Triple-contrastive-aligned CLIP	77.52	72.15	76.89	64.57
+ Semantic Map	77.98	**72.63**	75.79	**67.02**

However, both image-action contrastive and triple contrastive alignment can be adequately trained and show significant accuracy gains in matching corresponding pairs of two consecutive images and action or pairs of the image sequence and text sequence. We use batch size 64 for image-action contrastive alignment training, batch size three, and additionally, constraint each sequence to length 21 (during validation) for triple-contrastive alignment to keep a fair comparison between both accuracy scores. We observe that using zero-shot inference, the accuracy for text sequence and image sequence matching is around 33%, and the accuracy for two image frames and action matching lies between 3% and 8%. Both scores denote the random guessing from the model. After fine-tuning, the accuracy of predicting the correct image and text sequence pair is increased using MobileCLIP backbone and CLIP VIT-B-32 backbone to 60.22% and 45.62%, however still not entirely correct out of the mini-batch. We observe that triple-contrastive alignment indeed benefits accuracy in matching the correct action label of the consecutive images using all three backbones. Overall, based on its performance in the action-image matching task, we decided to use pre-aligned CLIP VIT-B-32 as the backbone to extract the features of our input.

4.2 Quantitative Results

In our experiments, we used OpenCLIP with ViT B-32 as our backbone, which we found to be the most suitable in our experiments. From Table 2, first, we observe that the fine-tuning of the CLIP backbone brought benefits in increasing the accuracy and the F1 score. Although the improvement in accuracy is not significant, e.g., 73.00%, 76.97%, 77.52% in the seen data, 71.87%, 79.56%, 76.89% in the unseen data, respectively. When the triple contrastive pre-alignment is applied, the F1 score improves by 8% on seen data and by around 20% on unseen data, compared to the contrastive alignment approach. Second, the semantic map improved the model's performance for the zero-shot backbone and triple-aligned backbone, showing the potential of using the map as an extra modality to enhance the model's spatial understanding rather than solely given the ego-centric RGB images. The backbone fine-tuned with image-to-action alignment

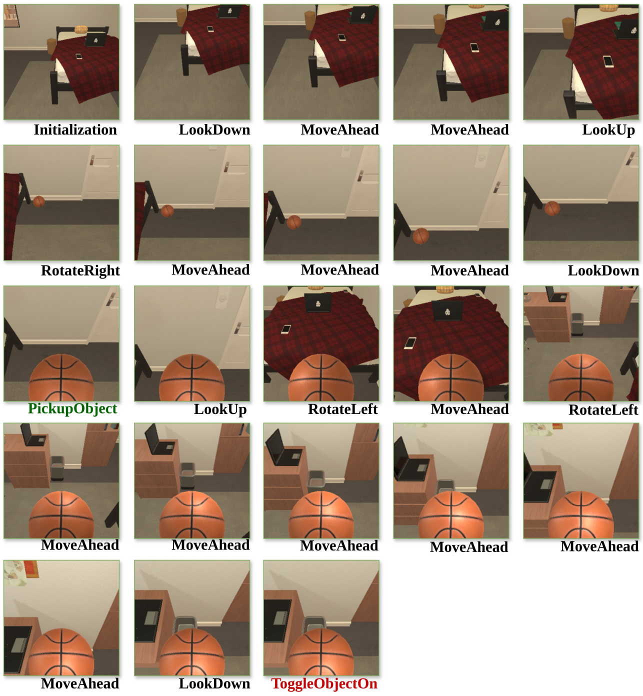

Fig. 4. Example task: "Look at the basketball in the light from the lamp". The given step-by-step instructions are as follows: 'Walk to the foot of the bed.' -> 'Pick up the basketball from the floor.' -> 'Go to the desk to your left.' ->'Turn on the lamp.'

has an acceptable accuracy score but a relatively low F1 score; this indicates that the model still suffers from the imbalanced data problem.

4.3 Qualitative Results

Figure 4 shows the decision of our agent for the task, "Pick up the basketball and turn on the desk lamp in the bedroom." The agent can comprehend most of the sub-goal instructions and correctly find and pick up the basketball beneath the bed. Furthermore, the agent rotates left twice and goes to the desk. However, this task eventually failed because of the model's wrong prediction of the lamp's location (which should be on the other side of the desk).

5 Conclusion

In this paper, we introduced LIAM, a multimodal model incorporating natural language input, egocentric RGB image input, action history, and an accumulated semantic map. Because all the modality-specific encoders encode inputs in their own latent space, we designed two pre-training tasks to pre-align the embedding spaces. This pre-training objective aims to maximize the agreement of the action representation and the global representation of two consecutive images. However, the pre-training task does not consider the alignment of the text encoding backbone. We introduced a triple contrastive alignment to address this issue. The agreement between the image sequence representation and the text sequence representation, the two-frame representation and the action representation are expected to be maximized.

Our model outperforms our baseline, using the OpenAI-released CLIP model and no map as an additional modality. We showed the importance of pre-aligning the embedding spaces from different modalities. In addition, using semantic maps as a modality to the end-to-end model brought benefits as well.

Future work holds immense potential for further enhancing our model. One avenue could be integrating the segmentation model inside our training stage and the semantic map generation stage. Furthermore, the methodology for encoding the map spatial information is another area that needs to be further investigated. We are particularly intrigued by the potential of considering maps as a knowledge base modality to the vision-language models.

Acknowledgements: This work has been funded by the German Ministry of Education and Research under the grant no. 16SV8683, project: Transferzentrum Roboter im Alltag (RimA).

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 1877–1901.

[2] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with CLIP latents". In: *arXiv:2204.06125* (2022).

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

[4] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. "CLIP-Fields: Weakly supervised semantic fields for robotic memory". In: *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*. 2023.

[5] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. "OpenVLA: An open-source vision-language-action model". In: *CoRR* abs/2406.09246 (2024).

[6] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. "ALFRED: A benchmark for interpreting grounded instructions for everyday tasks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10740–10749.

[7] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. "Detecting twenty-thousand classes using image-level supervision". In: *European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 350–368.

[8] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. "Open-vocabulary object detection via vision and language knowledge distillation". In: *International Conference on Learning Representations (ICLR)*.

[9] Timo Lüddecke and Alexander Ecker. "Image segmentation using text and image prompts". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 7086–7096.

[10] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. "GroupViT: Semantic segmentation emerges from text supervision". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 18134–18144.

[11] Mengmeng Wang, Jiazheng Xing, Jianbiao Mei, Yong Liu, and Yunliang Jiang. "ActionCLIP: Adapting language-image pretrained models for video action recognition". In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[12] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. "CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning". In: *Neurocomputing* 508 (2022), pp. 293–304.

[13] Alexander Pashevich, Cordelia Schmid, and Chen Sun. "Episodic transformer for vision-and-language navigation". In: *IEEE/CVF International Conference on Computer Vision (CVPR)*. 2021, pp. 15942–15952.

[14] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. "A persistent spatial semantic representation for high-level natural language instruction execution". In: *Conference on Robot Learning (CoRL)*. PMLR. 2022, pp. 706–717.

[15] So Yeon Min, Devendra Singh Chaplot, Pradeep Kumar Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. "FILM: Following instructions in language with modular methods". In: *International Conference on Learning Representations (ICLR)*. 2021.

[16] Wonjae Kim, Bokyung Son, and Ildoo Kim. "ViLT: Vision-and-language transformer without convolution or region supervision". In: *International Conference on Machine Learning (ICML)*. PMLR. 2021, pp. 5583–5594.