Contrastive Language, Action, and State Pre-training for Robot Learning

Krishan Rana*, Andrew Melnik and Niko Sünderhauf

***Abstract*—In this paper, we introduce a method for unifying language, action, and state information in a shared embedding space to facilitate a range of downstream tasks in robot learning. Our method, Contrastive Language, Action, and State Pre-training (CLASP), extends the CLIP formulation by incorporating distributional learning, capturing the inherent complexities and one-to-many relationships in behaviour-text alignment. By employing distributional outputs for both text and behaviour encoders, our model effectively associates diverse textual commands with a single behaviour and vice-versa. We demonstrate the utility of our method for the following downstream tasks: zero-shot text-behaviour retrieval, captioning unseen robot behaviours, and learning a behaviour prior for language-conditioned reinforcement learning. Our distributional encoders exhibit superior retrieval and captioning performance on unseen datasets, and the ability to generate meaningful exploratory behaviours from textual commands, capturing the intricate relationships between language, action, and state. This work represents an initial step towards developing a unified pre-trained model for robotics, with the potential to generalise to a broad range of downstream tasks.**

***Index Terms*—robot learning, natural language, contrastive learning, behaviour captioning, behaviour generation**

I. Introduction

Recent advancements in the fields of natural language processing and computer vision have demonstrated the potential of large-scale pre-trained models in learning general representations for a wide range of downstream tasks [1, 2, 3, 4]. However, the robotics community is yet to develop a unified representation that can encapsulate the rich and diverse information inherent in robotic systems, spanning language, action, and state. Such a unified representation could significantly improve the performance and generalization of robot learning algorithms, enabling seamless integration of language understanding and high-level task execution.

Multi-modal contrastive models such as CLIP [1] have demonstrated the ability to develop these desired intricate relationships between text and images, by learning shared representations that facilitate a wide range of downstream tasks including text-to-image generation [5], image captioning [6, 7], image classification [1] and segmentation [8, 9]. In this work, we explore what it takes to extend this idea to the robotics domain, where the alignment of language, states and actions can play a critical role in learning representations that can facilitate various downstream robot learning applications.

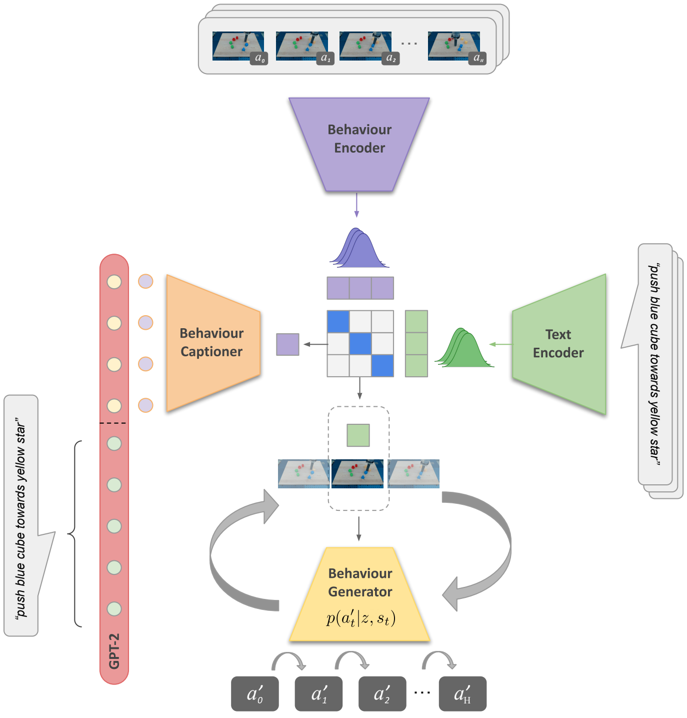

Fig. 1. We propose distributional encoders for both robot behaviour trajectories and textual descriptions and align the embedding space using a contrastive loss via the reparameterisation trick. We additionally regularise this space using two additional loss terms which encourage the embeddings to better align with the desired downstream tasks for behaviour captioning and generation.

The direct application of the CLIP architecture to connecting language and behaviours in robotics raises several challenges. Unlike the image-text pairing in CLIP, where static images are matched with their corresponding descriptions, the robotics domain deals with continuous, dynamic sequences of state-action pairs and their varying textual representations, making the one-to-one mapping between modalities more complex. Inherently these modalities exhibit one-to-many relationships where a single textual command can correspond to multiple valid robot trajectories, and conversely, a single robot trajectory could be accurately described by multiple textual commands. This relationship demands an alternative learning approach that can capture the variability and nuances of such

Krishan Rana and Niko Sünderhauf are with the QUT Centre for Robotics, and Andrew Melnik is with the University of Bielefeld. We acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. This work was supported by an Australian Research Council Discovery Project (project number DP220102398). *Correspondence email: `ranak@qut.edu.au`

International Conference on Robotics and Automation (ICRA), 2023 — Pretraining for Robotics Workshop

connections.

In this paper, we adapt the standard CLIP architecture to the domain of robotics with the aim of establishing a unified embedding space for language, actions, and states through contrastive pre-training. To tackle the inherent bidirectional one-to-many mapping between robot behaviours and textual descriptions, we propose to model the encoders as distributions from which we sample via reparameterisation [10]. The sampled embeddings are then aligned using the symmetric cross-entropy contrastive loss employed by CLIP. Furthermore, we encourage the model to learn generalizable and useful representations by regularizing the embedding space with two interconnected auxiliary downstream tasks: behaviour captioning and behaviour reconstruction.

Our proposed approach, known as CLASP (Contrastive Language, Actions, and State Pretraining), demonstrates superior retrieval performance compared to the traditional CLIP formulation when extended to the behaviour-text setting. Preliminary results additionally indicate the potential of the shared representation to facilitate downstream robot learning applications including robot behaviour captioning, text-conditioned behaviour generation and learning behaviour priors for reinforcement learning.

II. Related Work

In recent years, the pursuit of a shared representation for language, states, and actions has garnered significant interest in the robotics research community, aiming to develop more intuitive and versatile robotic systems. This endeavor involves the grounding of language in robotic actions and states, with various related works exploring different methodologies to achieve this goal.

1) Language-Conditioned Robot Learning: One prominent approach to grounding natural language in robot states and actions involves conditioning robot learning policies on language instructions [11, 12, 13, 14, 15]. This approach is based on the premise that an effective connection between language, actions, and states can be established via the task-centric loss function. However, applying these techniques in isolation frequently results in overfitting to initial object scenes, yielding suboptimal embeddings and limited task generalisation [16]. To mitigate these issues, prior research has explored the use of auxiliary multi-modal alignment losses [13, 16, 17, 18] in conjunction with standard imitation or reinforcement learning objectives. These studies have demonstrated that integrating appropriate regularisation with these objectives fosters the development of a more structured and coherent embedding space [16], substantially accelerating learning and improving the generalisation capabilities of the learned policy. In this work, we examine the multi-modal model alignment module in greater detail and investigate the requirements for learning an effective representation that can facilitate a variety of robot learning applications.

2) Pretraining for Robot Learning: In the domain of pretraining for robot learning, several studies have focused on the development of versatile shared embedding spaces using large-scale pre-training on video and language data. Nair *et al.*[19] use this to learn a general visual representation that could be used across a wide range of robot learning tasks. Fan *et al.*[20] build on this idea to train a similar shared embedding space and utilise the cosine similarity between text and video embeddings as a reward for downstream RL. Xiao *et al.*[21] finetune the CLIP model to align start-end images of a robot trajectory with textual descriptions in order to relabel new datasets. All these ideas demonstrate the versatility of a shared embedding space for different components of the robot learning pipeline. In our work, we extend these ideas to include robot actions within the multi-modal embedding space and explore the applicability of the shared representation to facilitate other downstream tasks including behaviour captioning as well as behaviour generation.

III. Motivation

The growing interest in developing shared representations across various modalities, such as text, images, and audio, has led to significant advancements in natural language processing and computer vision [1, 3, 22]. However, the robotics domain, which encompasses language, states, and actions, has yet to witness a dedicated effort to create a unified representation that can facilitate more natural and versatile robotic systems. The potential benefits of such a representation include the seamless integration of language understanding with high-level task execution within the robot learning pipeline, including text-guided exploration for reinforcement learning or hindsight instruction relabelling [23] for effective behaviour reuse. Although some works have indirectly addressed this challenge, a focused approach to establishing a shared representation for the complex robotics domain is still lacking. Unique challenges arise from the continuous and dynamic nature of state-action pairs, the diverse textual representations, and the inherent one-to-many relationships between language and robot behaviours. To tackle these complexities, innovative learning approaches capable of effectively capturing the intricacies of the relationships between language, actions, and states are needed, thus motivating the pursuit of novel methodologies in this research area.

IV. Problem Formulation

In this section, we formally define the problem of learning a shared embedding space for language, actions and states in the context of robotics.

Let $\mathcal{L}$ denote the language modality, where each element $l \in \mathcal{L}$ represents a natural language description or command. Similarly, let $\mathcal{B}$ denote the behavior modality, where each element $b \in \mathcal{B}$ represents a robot behavior, consisting of a sequence of state-action pairs $(s_1, a_1, s_2, \ldots, s_T, a_T)$, with T denoting the length of the sequence. The objective is to learn a shared embedding space $\mathcal{Z}$, where the elements $z_l \in \mathcal{Z}$ and $z_b \in \mathcal{Z}$ correspond to the embeddings of language and behaviours, respectively. This space should capture the complex relationships between these modalities and enable efficient transfer learning across various robot learning tasks. To achieve this, we propose a multi-modal contrastive learning framework, which consists of two encoder networks, $\phi_l : \mathcal{L} \rightarrow \mathcal{Z}$ and $\phi_b : \mathcal{B} \rightarrow \mathcal{Z}$. These encoders are designed to project the elements of each modality into the shared embedding space $\mathcal{Z}$, preserving the rich relationships between language and behaviours. The learning objective is to minimize a contrastive loss function that encourages the alignment of corresponding language and behaviour embeddings while pushing apart non-matching pairs. The proposed framework should also take into account the inherent bidirectional one-to-many mappings between robot behaviours and textual descriptions, as well as the temporal dependencies between state-action sequences.

V. Methodology

In this section, we present our methodology for learning a shared embedding space for language and behaviours in the context of robotics. We leverage contrastive learning, with distributional encoders to facilitate the learning of one-to-many mappings between text and behaviours and vice versa. Additionally, we incorporate two auxiliary tasks for regularising the model and improving generalisation. Full implementation details are provided in the Appendix section.

A. Distributional Encoders and Sampling

We utilise two distributional encoders, one for language (ϕ_l) and one for behaviours (ϕ_b), that output the parameters of a Gaussian distribution in the shared embedding space. For each language description l and behaviour b, we compute the mean (μ) and variance (σ^2) of the corresponding embeddings using the respective encoders:

$$\mu_l, \sigma_l^2 = \phi_l(l), \qquad \mu_b, \sigma_b^2 = \phi_b(b) \tag{1}$$

To obtain the required embeddings for alignment, we sample from these distributions using the reparameterisation trick [10], commonly used when training variational auto-encoder networks:

$$z_b = \mu_b + \epsilon_b \odot \sqrt{\sigma_b^2}, \qquad z_l = \mu_l + \epsilon_l \odot \sqrt{\sigma_l^2} \tag{2}$$

where ϵ_b and ϵ_l are random noise vectors drawn from a standard normal distribution ($\epsilon \sim \mathcal{N}(0, 1)$) and $\odot$ denotes element-wise multiplication.

B. Behaviour-Language Alignment Loss

To align the embeddings of behaviours and their corresponding textual descriptions, we utilise the same contrastive objective used for pairing images and captions in CLIP [1] which is based on the symmetric cross-entropy loss. Given a mini-batch of N samples, the loss function for our behaviour-text alignment is given by:

$$\mathcal{L}_{\text{align}} = -\frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{\exp \left\langle z_{b_i}, z_{l_i} \right\rangle / \tau}{\sum_{j=1}^{N} \exp \left\langle z_{b_i}, z_{l_j} \right\rangle / \tau} + \log \frac{\exp \left\langle z_{l_i}, z_{b_i} \right\rangle / \tau}{\sum_{j=1}^{N} \exp \left\langle z_{l_i}, z_{b_j} \right\rangle / \tau} \right] \tag{3}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the inner product between two vectors, z_{b_i} is the behaviour embedding for the i-th sample, z_{t_i} is the corresponding text embedding, and τ is a temperature hyperparameter. This loss encourages the model to align behaviour and text embeddings for each sample while pushing them away from the embeddings of other samples in the batch. Similar to the original CLIP model, the loss is computed for both behaviour-to-text and text-to-behaviour directions, and their average is used for optimization.

C. Auxiliary Tasks

To regularise the model and improve generalisation, we introduce two interconnected auxiliary tasks: behaviour captioning and behaviour generation.

1) Behaviour Captioning: The goal of this task is to predict the natural language description l of a given behaviour sequence b. Following from the CLIPCap model presented by Mokady *et al.*[6], we assume that all the necessary information for captioning a behaviour b is present in the sampled embedding z_b, given its alignment with text in the shared embedding space. This in essence should allow us to predict the corresponding caption directly from the behaviour embedding:

$$\max_{\theta} \sum_{i=1}^{N} \log p_{\theta} \left(l_1^i, \ldots, l_n^i \mid z_b^i \right), \tag{4}$$

where we refer to the captions as a sequence of tokens $l = l_1, \ldots, l_n$ padded to a maximum length n. Similar to [6], we focus on prefix fine-tuning [24] as a sample efficient strategy for training our captioning network. We utilise a pre-trained GPT-2 [25] model as the backbone of the captioning network and solely train a mapping network ψ which projects our behaviour embedding z_b to k embedding vectors suitable as input to the large language model:

$$p_1^i, \ldots, p_k^i = \psi \left(z_b \right) \tag{5}$$

The final objective for training the mapping component ψ is to predict the caption tokens conditioned on the prefix in an auto-regressive fashion using the cross-entropy loss:

$$\mathcal{L}_{\text{caption}} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \log p_{\theta} \left(l_j^i \mid p_1^i, \ldots, p_k^i, l_1^i, \ldots, l_{j-1}^i \right) \tag{6}$$

2) Behavior Generation: In the behaviour generation task, we aim to reconstruct the encoded action sequence $\boldsymbol{a}$, that is processed by the behaviour encoder, from the sampled language embedding z_l. Given that robot environments can be dynamic, we model the behaviour generator π as a closed-loop policy that conditions on both the text embedding and the current state s_t in order to generate the corresponding action a'_t [26, 27]. This network is trained using the mean squared error (MSE) loss, which measures the difference between the predicted action sequence and the ground truth action sequence:

$$L_\pi = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}||a_t - \pi(a'_t|z_l, s_t)||_2^2, \tag{7}$$

where a_t denotes the t-th ground truth action in the sequence, $\pi(a'_t|z_l, st)$ denotes the predicted action given the text embedding z_l and state s_t, and T is the total number of actions in the sequence.

D. Total Loss

The total loss for our model is a combination of the three loss terms, weighted by their respective hyperparameters β:

$$\mathcal{L}_{\text{CLASP}} = \beta_1\mathcal{L}_{\text{align}} + \beta_2\mathcal{L}_{\text{caption}} + \beta_3\mathcal{L}_\pi \tag{8}$$

Our model is trained by minimizing this loss, which encourages the learned shared embedding space to structure itself such that it can effectively align language, actions, and states. By striking a balance between these objectives, the model learns to capture the intricacies and nuances of the relationships between these modalities, leading to better generalisation and performance on downstream applications.

VI. Evaluation

A. Training Dataset

Our method is trained and evaluated using the Language-Table manipulation dataset [12]. This dataset focuses on a robotic arm that manipulates blocks on a tabletop, with tasks guided by natural language instructions. The dataset contains a mix of real-world and simulated demonstration data, featuring a variety of robot state-action trajectories, each paired with a natural language description of the corresponding behaviour. The test environment consists of a xArm5 robot, constrained to move in a 2D plane, with a cylindrical end-effector, in front of a smooth wooden board with a fixed set of 8 plastic blocks, comprising 4 colours and 6 shapes. Actions are 2D delta Cartesian setpoints, from the previous setpoint to the new one. State information consists of RGB third-person images of the robot and board as shown in Figure 2.

B. Alignment Evaluation

We evaluate the zero-shot retrieval accuracy of our model against a non-distributional variant on a held-out dataset from the `Language-Table` suite, comparing their top-1 and top-5 retrieval accuracies for both text and behaviour. We summarise the results in Table I.

TABLE I
Zero-shot retrieval on an unseen dataset

Text Retrieval				**Behaviour Retrieval**			
CLASP		CLASP (Distributional)		CLASP		CLASP (Distributional)	
R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
40.0	73.3	**73.3**	**93.3**	**86.7**	100	66.7	**100**

TABLE II
Behaviour-to-Description Translation Accuracy

CLASP (Distributional)	**CLASP**	**Seq2Seq**
46.6%	40.0%	26.6%

Our distributional encoders demonstrate improved generalization, with a 33.3% increase in top-1 text retrieval accuracy and a 20.0% increase in top-5 accuracy. Although top-5 behaviour retrieval accuracy remains at 100% for both variants, the distributional approach shows a drop in top-1 accuracy.

This discrepancy could be due to the distributional encoders capturing a broader range of behaviour representations, which enhances text retrieval performance but makes it harder to pinpoint the exact behaviour in top-1 results. Nevertheless, the distributional variant effectively identifies relevant behaviours within the top-5 results, suggesting that it is sensitive to subtle behaviour differences and could excel in fine-grained retrieval tasks. Further investigation is needed to confirm this hypothesis and understand the trade-offs between the distributional and non-distributional CLASP variants.

C. Behaviour Captioning

We further evaluate our model's ability to caption unseen behaviour trajectories by comparing its performance with a non-distributional variant (CLASP) and a Seq2Seq baseline that does not utilise the aligned representation space for captioning. The results are summarised in Table II.

Our model achieves a 46.6% translation accuracy, surpassing both alternatives. This improved performance can be attributed to the aligned embedding space, which enables effective information transfer between behaviour sequences and text captions, leading to increased captioning accuracy. The Seq2Seq baseline, without the shared space, only attains 26.6% accuracy, while the non-distributional CLASP variant reaches 40.0% accuracy. These findings suggest that incorporating distributional encoders allows our model to capture the nuanced behaviour-text relationships, resulting in better captioning performance.

We present qualitative captioning examples of our approach in Figure 2. Additionally, we provide the ground truth language descriptions from the dataset to emphasise the one-to-many mapping nature of behaviours-to-textual descriptions.

It is worth noting that a significant number of captioning failures stem from the visual model's inability to distinguish subtle object differences in our dataset, such as ”yellow pen-

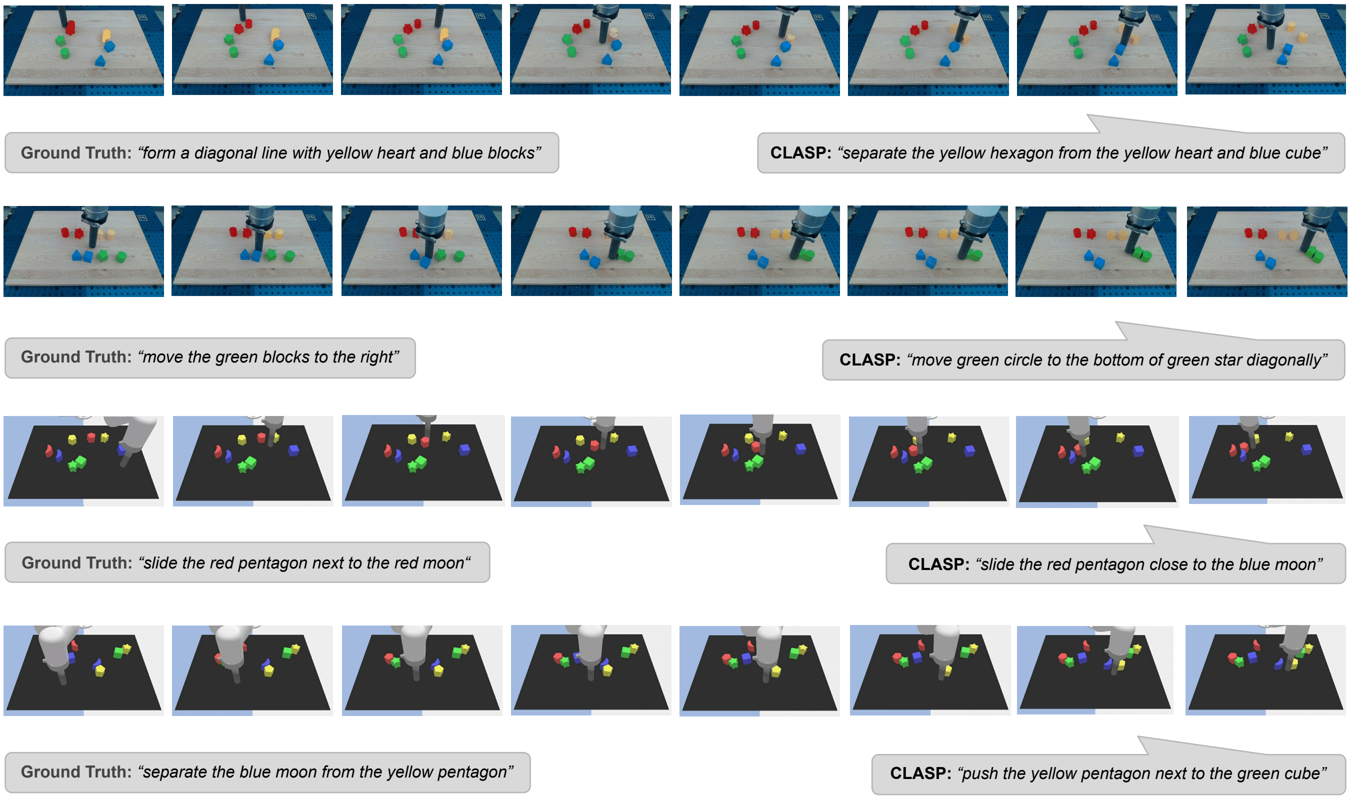
Ground Truth: "form a diagonal line with yellow heart and blue blocks"
CLASP: "separate the yellow hexagon from the yellow heart and blue cube"
Ground Truth: "move the green blocks to the right"
CLASP: "move green circle to the bottom of green star diagonally"
Ground Truth: "slide the red pentagon next to the red moon"
CLASP: "slide the red pentagon close to the blue moon"
Ground Truth: "separate the blue moon from the yellow pentagon"
CLASP: "push the yellow pentagon next to the green cube"

Fig. 2. Captioning unseen robot trajectories using our trained model on both real and simulation environments. Note the significant discrepancy between the labelled ground truth description and the generated description. This highlights the one-to-many relationship between the behaviours and textual descriptions.

tagon" from "yellow star" or "green cube" from "green star." In our experiments, we used a frozen CLIP visual encoder to process these images, which is not specifically designed for fine-grained object-level feature extraction. We believe that using a fine-tuned or alternative model would yield better results.

D. Behaviour Generation

In our final evaluation, we investigate the effectiveness of the shared embedding space for meaningful behaviour generation. The ability to generate useful behaviours from either a sampling space or textual descriptions is crucial for various robotic learning applications, such as facilitating exploration in reinforcement learning [26, 27, 28], model predictive control [29], or dataset generation for imitation learning. We assess the behaviour generator's capacity to produce meaningful behaviours in the context of the `Language-Table` environment. Here, useful behaviours are defined as trajectory sequences that result in block rearrangements constrained to the board region. We evaluate our skill generator based on this criterion and compare its performance against random exploration. We additionally learn a state-conditioned behaviour prior over this embedding space using the approach proposed in [27]. The results, presented in Table III, demonstrate that both methods leveraging the distributional embedding space induced by CLASP can produce a high proportion of useful behaviours in the environment, outperforming random exploration alone.

TABLE III
PERCENTAGE OF USEFUL TRAJECTORIES GENERATED DURING EXPLORATION

	CLASP		
Method	**Behaviour Prior**	**Text Encoding**	**Random Exploration**
Useful Trajectories	87.7%	60.0%	27.7%

VII. Conclusions

This body of work represents an initial step towards developing a shared embedding space for language, states, and actions in robotics. Preliminary results indicate that accounting for the bidirectional one-to-many nature of the text-behaviour relationship is essential when constructing this shared representation, especially for downstream tasks involving generative modeling. Our approach demonstrates improved retrieval performance compared to non-distributional methods and showcases the applicability of the shared embedding space across two distinct downstream tasks. It is important to note that the evaluation conducted in this study focused on a single robot domain, and further assessments across larger and more diverse datasets are necessary to establish the viability of the approach. We encourage continued research in this area as more extensive and varied datasets become available to the robot learning community.

Acknowledgments

The authors would like to thank Jad Abou-Chakra, Sourav Garg, Mingda Xu, Jesse Haviland and Robert Lee for their valuable and insightful discussions towards this contribution.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*, 2021.

[2] OpenAI, “Gpt-4 technical report,” *ArXiv*, vol. abs/2303.08774, 2023.

[3] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” *ArXiv*, vol. abs/2301.12597, 2023.

[4] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. H. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. R. Florence, “Palm-e: An embodied multimodal language model,” *ArXiv*, vol. abs/2303.03378, 2023.

[5] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” *ArXiv*, vol. abs/2204.06125, 2022.

[6] R. Mokady, “Clipcap: Clip prefix for image captioning,” *ArXiv*, vol. abs/2111.09734, 2021.

[7] M. Tang, Z. Wang, Z. Liu, F. Rao, D. Li, and X. Li, “Clip4caption: Clip for video caption,” *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[8] T. Lüddecke and A. Ecker, “Image segmentation using text and image prompts,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 7086–7096.

[9] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, “Language-driven semantic segmentation,” in *International Conference on Learning Representations*, 2022.

[10] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *CoRR*, vol. abs/1312.6114, 2013.

[11] M. Shridhar, L. Manuelli, and D. Fox, “Perceiver-actor: A multi-task transformer for robotic manipulation,” in *Conference on Robot Learning*, 2022.

[12] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, “Interactive language: Talking to robots in real time,” 2022.

[13] O. Mees, L. Hermann, and W. Burgard, “What matters in language conditioned robotic imitation learning over unstructured data,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 11 205–11 212, 2022.

[14] C. Lynch and P. Sermanet, “Language conditioned imitation learning over unstructured data,” *Robotics: Science and Systems*, 2021.

[15] T. Carta, C. Romac, T. Wolf, S. Lamprier, O. Sigaud, and P.-Y. Oudeyer, “Grounding large language models in interactive environments with online reinforcement learning,” 2023.

[16] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, “Bc-z: Zero-shot task generalization with robotic imitation learning,” in *Conference on Robot Learning*, 2022.

[17] D. I. A. T. J. Abramson, A. Ahuja, A. Brussee, F. Carnevale, M. Cassin, F. Fischer, P. Georgiev, A. Goldin, T. Harley, F. Hill, P. C. Humphreys, A. Hung, J. Landon, T. P. Lillicrap, H. Merzic, A. Muldal, A. Santoro, G. Scully, T. von Glehn, G. Wayne, N. Wong, C. Yan, and R. Zhu, “Creating multimodal interactive agents with imitation and self-supervised learning,” *ArXiv*, vol. abs/2112.03763, 2021.

[18] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, M. Wainwright, C. Apps, D. Hassabis, and P. Blunsom, “Grounded language learning in a simulated 3d world,” *ArXiv*, vol. abs/1706.06551, 2017.

[19] S. Nair, E. Mitchell, K. Chen, B. Ichter, S. Savarese, and C. Finn, “Learning language-conditioned robot behavior from offline data and crowd-sourced annotation,” in *Conference on Robot Learning*, 2021.

[20] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar, “Minedojo: Building open-ended embodied agents with internet-scale knowledge,” in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[21] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson, “Robotic skill acquisition via instruction augmentation with vision-language models,” *ArXiv*, vol. abs/2211.11736, 2022.

[22] A. Guzhov, F. Raue, J. Hees, and A. Dengel, “Audioclip: Extending clip to image, text and audio,” 2021.

[23] G. Cideron, M. Seurin, F. Strub, and O. Pietquin, “Higher: Improving instruction following with hindsight generation for experience replay,” *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 225–232, 2020.

[24] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. abs/2101.00190, 2021.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” 2019.

[26] K. Pertsch, Y. Lee, and J. J. Lim, “Accelerating reinforcement learning with learned skill priors,” in *Conference on Robot Learning (CoRL)*, 2020.

[27] K. Rana, M. Xu, B. Tidd, M. Milford, and N. Sunderhauf, “Residual skill policies: Learning an adaptable skill-based action space for reinforcement learning for robotics,” in *Conference on Robot Learning*, 2022.

[28] A. Singh, H. Liu, G. Zhou, A. Yu, N. Rhinehart, and S. Levine, “Parrot: Data-driven behavioral priors for reinforcement learning,” 2020.

[29] L. X. Shi, J. J. Lim, and Y. Lee, “Skill-based model-based reinforcement learning,” in *Conference on Robot Learning*, 2022.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[31] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” *International Conference on Learning Representations (ICLR)*, 2017.

Appendix

Implementation Details

A. Behaviour Encoder

Each training example consists of $(s, a, l) \sim D$, where $s \in \mathbb{R}^{T\times320\times180\times3}$ is the RGB observation history with varying sequence length T for the behaviour trajectory, $a \in \mathbb{R}^2$ is the delta Cartesian action of the robot arm, and l is the natural language instruction carried out by the robot. Each image frame is passed through a pre-trained CLIP image encoder to obtain a visual feature representation $f_s \in \mathbb{R}^{512}$ before concatenating it with the normalized robot action. We prepend a [CLS] token to this sequence, which will later serve as the final representation for the behaviour sequence. The sequence of state-action pairs, including the [CLS] token, is processed by an MLP to obtain the shape $[T+1, \mathtt{dmodel}]$, and 2D position-encoded before being passed through a standard transformer encoder model [30]. Our behaviour encoder transformer has 2 layers, with $\mathtt{dmodel} = 512$, 2 heads, a feed-forward width of 128, and a dropout rate of 0.1. The [CLS] token is then processed by a 3-layer MLP, which outputs the desired mean and sigma for the encoder distribution, both of dimension 512.

B. Text Encoder

We employ a pre-trained CLIP text encoder [1] for processing the input instruction text. The text is preprocessed by removing punctuation and extra spaces and is fed into a pre-trained CLIP text encoder, which produces a textual feature representation $f_l \in \mathbb{R}^{512}$. Subsequently, the textual representation is passed through a 3-layer MLP projector head, which generates the 512-dimensional mean and sigma for the desired distributional output of the textual encoder.

C. Behaviour Generator

The role of the behaviour generator is to take an embedding z from the shared latent space and map it to a sequence of meaningful actions captured by the behaviour dataset. Due to the dynamic nature of robot environments, we make this generation process closed-loop by conditioning it on the current state. As a result, the behaviour generator can be viewed as a closed-loop policy:

$$a_t = \pi(z, s_t) \tag{9}$$

We model the policy π as a 6-layer MLP, which takes the sampled embedding z and the current state s_t as inputs and outputs the corresponding 2-dimensional action a_t. The embedding space encapsulates a diverse range of behaviours, and selecting an appropriate z for sampling can be challenging. In this work, we describe two strategies used for evaluation:

1) Text-Conditioned Generation: In this approach, we leverage the shared embedding space and distributional nature of our encoders to sample from the embedding space during inference. Given the alignment between text and behaviours, we can map textual commands to a distribution over z, from which we can sample an appropriate z for decoding into a behaviour sequence. By doing so, we effectively utilize the learned connections between language and behaviour to generate meaningful action sequences based on the input textual commands.

2) State-Conditioned Behaviour Prior: We additionally explore the ability to utilise this shared embedding space to learn a state-conditioned behaviour prior. In this case the prior is task agnostic and captures the entire range of state-relevant behaviours in the embedding space. Such a prior has been shown to be useful for accelerating RL exploration [26, 27]. We follow the same strategy used in [27] to learn a state conditioned prior over an existing embedding space using normalising flows. The network parameterising the behaviour prior $f : \mathcal{Z} \times \mathcal{S} \to G$ is a conditional real NVP [31] which consists of four affine coupling layers, where each coupling layer takes as input the output of the previous coupling layer, and the robot state vector s_0 from the start of the behaviour sequence. We use a standard Gaussian $p_{\mathcal{G}}(g) \sim \mathcal{N}(0, I)$ as our base distribution for our generative model. We refer the reader to [27] for a more detailed treatment of this model. The loss function for a single example is given by:

$$\mathcal{L}_{prior} = \log p_{\mathcal{G}}(f(z, s_0)) + \log \left| \det \frac{\partial f}{\partial z^\top} \right| . \tag{10}$$

Once trained, the flow model allows us to sample an appropriate z from the embedding space via the bijective mapping function $f^{-1}(g, s) \sim p(z|s)$, where g is sampled from a simple Gaussian distrbution $\mathcal{N}(0, I)$.

D. Behaviour Captioner

As previously mentioned, the captioning network comprises a trainable mapping network and a frozen GPT-2 decoder network. The mapping network’s purpose is to project a sampled behaviour embedding z_b into K token embeddings, which can then be passed as input to the GPT model. Our mapping network consists of 8 multi-head self-attention layers, each with 8 heads. We set the prefix length K to 10. Once trained, the decoding process is performed through beam search to obtain the natural language description of a behaviour. Considering the one-to-many mapping from behaviours to text, we evaluate captioning performance manually via visual inspection, assessing the quality and relevance of the generated captions to the behaviour video stream.