Behavioral Cloning via Search in Video PreTraining Latent Space

Federico Malato*
University of Eastern Finland
fmalato@uef.fi

Florian Leopold*
University of Bielefeld
fleopold@techfak.uni-bielefeld.de

Amogh Raut
Indian Institute of Technology BHU

Ville Hautamäki
University of Eastern Finland

Andrew Melnik
University of Bielefeld

Abstract

Our aim is to build autonomous agents that can solve tasks in environments like Minecraft. To do so, we used an imitation learning-based approach. We formulate our control problem as a search problem over a dataset of experts' demonstrations, where the agent copies actions from a similar demonstration trajectory of image-action pairs. We perform a proximity search over the BASALT MineRL-dataset in the latent representation of a Video PreTraining model. The agent copies the actions from the expert trajectory as long as the distance between the state representations of the agent and the selected expert trajectory from the dataset do not diverge. Then the proximity search is repeated. Our approach can effectively recover meaningful demonstration trajectories and show human-like behavior of an agent in the Minecraft environment.

Introduction

This study was motivated by the MineRL BASALT 2022 challenge [1]. In the challenge, an agent must solve the following tasks: find a cave, catch a pet, build a village house, and make a waterfall [1]. The provided dataset of experts' demonstrations contains trajectories of image-action pairs. Additionally, both the MineRL BASALT dataset and environments do not contain reward information. Therefore, our primary focus was on Behavioural Cloning (BC) and Planning [2][3] methods to address the tasks, rather than deep reinforcement learning (DRL) [4][5].

Methods

A dataset of expert demonstrations solving the following tasks was provided [1]: find a cave, catch a pet, build a village house, and make a waterfall. Each episode is a trajectory of image-action pairs. No reward information is provided.

In our approach, we use experts' demonstrations to reshape the control problem as a search problem over a latent space of partial trajectories (called *situations*). Our work assumes that:

- Similar *situations* require similar solutions or actions.
- A *situation* can be represented in a latent space.
- The *situations* latent space is a metric space. Therefore, we can assess the numerical similarity between any two *situations*.

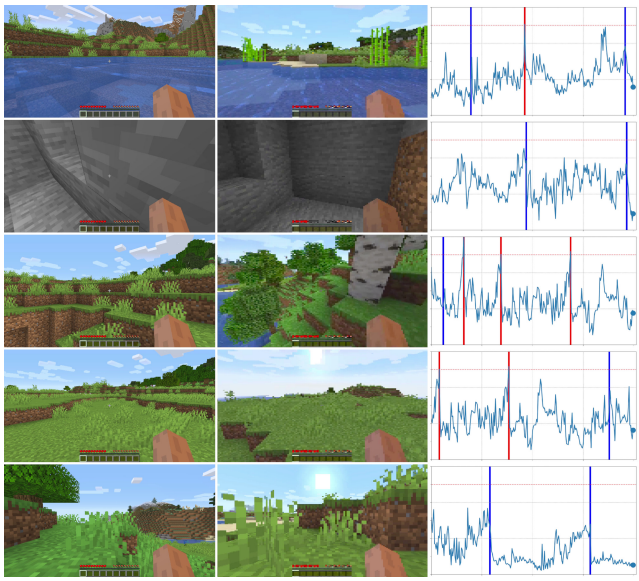

Figure 1: Similarity and divergence of five pairs of simulator and dataset *situation* trajectories. **Left column**: current frame from the MineRL environment. **Middle column**: current reference frame from the dataset trajectory that the agent follows. **Right column**: L1-distance plot between VPT embedding points of simulator and dataset *situations*. The current step shown as RGB images in the **Left** and **Middle** columns is highlighted by the rightmost bold dot marker in the **Right** column. X-axis indicates 256 time steps. Y-axis indicates L1 distance between two embedding points in the VPT latent space and ranges from 0.1 to 0.4. The deviation threshold is show by dashed red horizontal lines L1 = 0.35. Colored vertical lines mark new search events to find the most similar situation in the dataset. Blue lines indicate time-based initiated searches, red lines indicate deviation-threshold based initiated searches. Gray dotted lines: every 64 steps in the X-axis and every 0.1 steps in the Y-axis.

* Equal contribution.

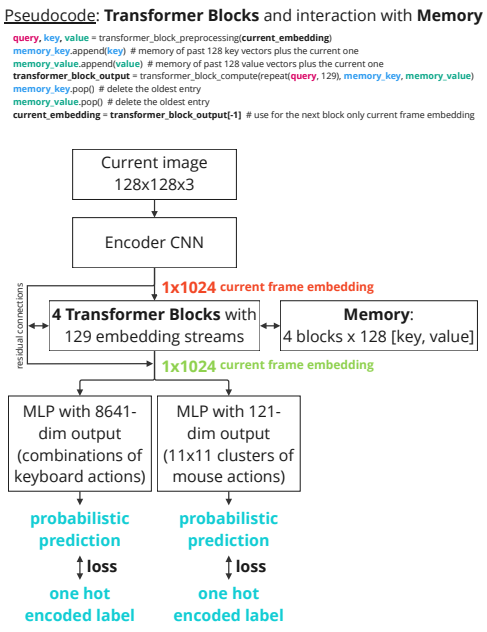

Figure 2: VPT architecture.

Video PreTraining (VPT) model Our approach uses a provided VPT model [6] for encoding a *situation* in a latent space (see Figure 2). The model uses the IMPALA [7] convolutional neural network (CNN) as backbone for the encoding of individual images. The CNN network encodes each image into a 1024-dimensional vector. The stack of 129 CNN outputs passes through four transformer blocks (see Figure 2). Additionally to the current frame, a memory stack stores the last 128 embeddings for each transformer block. The output of the last transformer block are 129 embedding vectors, each 1024-dimensional. The architecture discards 128 output embedding vectors of the last transformer block and processes further only the current's frame embedding vector. Two MLP output heads take as an input the current's frame embedding vector to predict actions. The first output head predicts a discrete action (one out of 8641 possible combinations of compound keyboard actions). The second output head predicts a computer mouse control as a discrete cluster of one of the possible 121=11x11 mouse displacement regions (±5 regions for X times ±5 regions for Y). The architecture is shown in Figure 2.

Search-based BC Search-based behavioral cloning (BC) aims to reproduce an expert's behavior with high fidelity by copying its solutions from past experience. We define a *situation* as a set $\{(o_\tau, a_\tau)\}_{\tau=t}^{t+\Delta t}$ of consecutive observation-action pairs coming from a set of provided expert's trajectories, where Δt is less or equal to the number of input slots of a transformer block that processes embedding vectors of input images.

We encode the expert's past *situations* through a provided VPT model [6]. Thus, we obtain a latent space populated by N-dimensional *situation* points. Due to the expert's optimality assumption, we can assume that each *situation* has been addressed and solved in an *optimal* way.

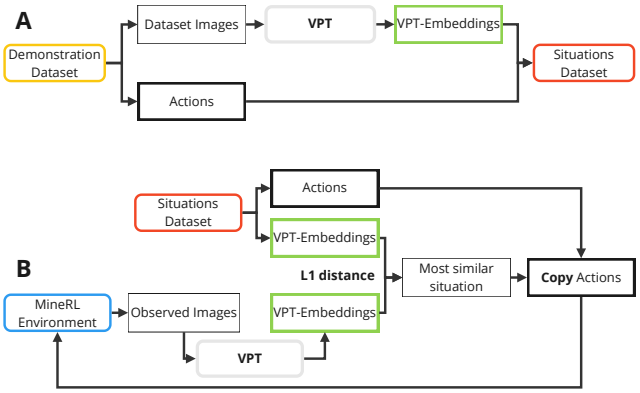

Figure 3: Our approach. (**A**) Training procedure. (**B**) Evaluation loop.

We encode each sampled *situation* with the same network. Then, we search the nearest embedding point in the dataset of *situation* points. Once the reference *situation* has been selected, we copy its corresponding actions. After each time-step we update the current and reference *situations*, by updating the queue of embedding vectors of images for the current *situation*, while shifting to the next time-step in the recorded trajectory from the dataset for the reference *situation*. To assess the similarity, we compute the L1 distance between the current *situation* and the reference *situation*. In most cases, the reference and the current *situations* will evolve differently over time, thus, their L1 distance will diverge. Therefore, at each timestep we recompute the similarity of the current and reference *situations*. A new search is performed whenever either one of two conditions is met:

- The L1 distance between current and reference *situations* overcomes a threshold (see red lines in Figure 1);
- The trajectory from the dataset has been followed for more than 128 time-steps (see blue lines in Figure 1).

Choosing feature divergence as a criterion for controlling search comes with a major advantage: whenever the copied actions can not be performed (e.g. there are physical constraints that limit the agent movement space), the features will diverge even faster. Thus, our agent will quickly perform a new search and address the faulty *situation*.

Our approach is illustrated in Figure 3. We refer to the generation of the latent space as the "training" procedure of our agent. Rather, it is a preprocessing step needed to ensure prior knowledge to our agent.

Experiments and Results

We applied our method to the MineRL BASALT Challenge 2022 [1], where it ranked top of the leaderboard at the end of Round 1. The agent had to demonstrate human-like behavior while completing the tasks. Our agent produces visually human-resembling behaviour in the tasks.

In Table 1 we report quantitative measurements of the L1 distance before and after a new search for the best matching trajectory from the dataset. In all four tasks, we found that the average L1 distance after a search is much lower than before it.

Environment	Spike Before	Spike After	Window Before	Window After
FindCave	$.37 \pm .02$	$.17 \pm .02$	$.23 \pm .05$	$.15 \pm .03$
VillageAnimalPen	$.37 \pm .01$	$.16 \pm .02$	$.22 \pm .04$	$.15 \pm .02$
MakeWaterfall	$.37 \pm .02$	$.16 \pm .02$	$.22 \pm .04$	$.15 \pm .02$
BuildVillageHouse	$.37 \pm .02$	$.18 \pm .03$	$.23 \pm .04$	$.16 \pm .02$

Table 1: Average L1 distance value and its standard deviation between current state and reference *situation*. Spike denotes deviation based new search, Window denotes time based new search. Values shown are right before and after a new search.

A *situation* encapsulates both current and past information. Therefore, at the very beginning of an episode, the *situation* embedding may be not informative. To mitigate this, we allow the agent to *warm up*, by keeping it still for the first second of a new episode. This way, the agent can gather some images and produce a more informative representation of the current *situation*. Using the *warm up* phase can be vital whenever the agent faces a dangerous *situation* at the beginning of an episode, e.g. when spawning close to a lava pit.

Discussion & Conclusion

Here we presented our approach that represents the control problem as a search problem over a latent space of partial trajectories (called *situations*) from a dataset of experts' demonstrations. Our approach can effectively recover meaningful demonstration trajectories and show human-like behavior of an agent in the Minecraft environment. Possible directions for improving the approach are methods of self-supervised segmentation of important objects in first-person views [8], multi-modal fusion of segmented representations [9], modularization of control [10][11] and involvement of working memory [12].

References

[1] R. Shah, C. Wild, S. H. Wang, N. Alex, B. Houghton, W. H. Guss, S. P. Mohanty, A. Kanervisto, S. Milani, N. Topin, P. Abbeel, S. Russell, and A. D. Dragan, "The minerl BASALT competition on learning from human feedback," *CoRR*, vol. abs/2107.01969, 2021.

[2] S. Beohar and A. Melnik, "Planning with rl and episodic-memory behavioral priors," *arXiv preprint arXiv:2207.01845*, 2022.

[3] S. Beohar, F. Heinrich, R. Kala, H. Ritter, and A. Melnik, "Solving learn-to-race autonomous racing challenge by planning in latent space," *arXiv preprint arXiv:2207.01275*, 2022.

[4] N. Bach, A. Melnik, M. Schilling, T. Korthals, and H. Ritter, "Learn to move through a combination of policy gradient algorithms: Ddpg, d4pg, and td3," in *International Conference on Machine Learning, Optimization, and Data Science*, pp. 631–644, Springer, 2020.

[5] M. Schilling, A. Melnik, F. W. Ohl, H. J. Ritter, and B. Hammer, "Decentralized control and local information for robust and adaptive decentralized deep reinforcement learning," *Neural Networks*, vol. 144, pp. 699–725, 2021.

[6] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune, "Video pretraining (vpt): Learning to act by watching unlabeled online videos," *arXiv preprint arXiv:2206.11795*, 2022.

[7] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu, "IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures," *CoRR*, vol. abs/1802.01561, 2018.

[8] A. Melnik, A. Harter, C. Limberg, K. Rana, N. Sünderhauf, and H. Ritter, "Critic guided segmentation of rewarding objects in first-person views," in *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pp. 338–348, Springer, 2021.

[9] T. Korthals, M. Hesse, J. Leitner, A. Melnik, and U. Rückert, "Jointly trained variational autoencoder for multi-modal sensor fusion," in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2019.

[10] A. Melnik, S. Fleer, M. Schilling, and H. Ritter, "Modularization of end-to-end learning: Case study in arcade games," *arXiv preprint arXiv:1901.09895*, 2019.

[11] K. Konen, T. Korthals, A. Melnik, and M. Schilling, "Biologically-inspired deep reinforcement learning of modular control for a six-legged robot," in *2019 IEEE International Conference on Robotics and Automation Workshop on Learning Legged Locomotion Workshop,(ICRA) 2019, Montreal, CA, May 20-25, 2019*, 2019.

[12] A. Melnik, F. Schüler, C. A. Rothkopf, and P. König, "The world as an external memory: the price of saccades in a sensorimotor task," *Frontiers in behavioral neuroscience*, vol. 12, p. 253, 2018.