Learning Visual Embeddings for Reinforcement Learning

Mihir Rana and Kenil Tanna
Center for Data Science, New York University

Objectives

To learn meaningful embedding representations for visual tasks entirely from a small number of unlabeled expert demonstration videos by constructing a self-supervised vision task, and use these representations to improve the training in reinforcement learning tasks.

Introduction

- Deep Reinforcement Learning (RL) methods struggle in tasks with sparse reward environments
- We use the approach proposed in [1]:

1. **Self-Supervised Computer Vision**: Classify time misalignment between a pair of frames sampled from a video to generate embeddings and get an understanding of the environment
2. **Imitation Learning**: Use these embeddings to extract dense rewards for improving the training of an agent

Figure 1: Dog Learns to Open Door by Watching Human

- We study the algorithm described in [1] and re-implement it for a maze environment constructed using [2] where the original rewards are extremely sparse, and demonstrate the effectiveness of this method, identifying potential problems, and proposing possible solutions.
- **Assumption**: Expert trajectory is *optimal*; hence, demonstrations encode distance between frames, i.e., closer frames have lesser distance in some abstract space (which we extract via embeddings)

Data

The dataset is constructed using [2] and comprises unlabelled demonstration videos of an expert navigating through (i) 1000 and (ii) 10000 mazes of grid sizes 8×8 and 16×16 each, with obstacles such as closed doors:

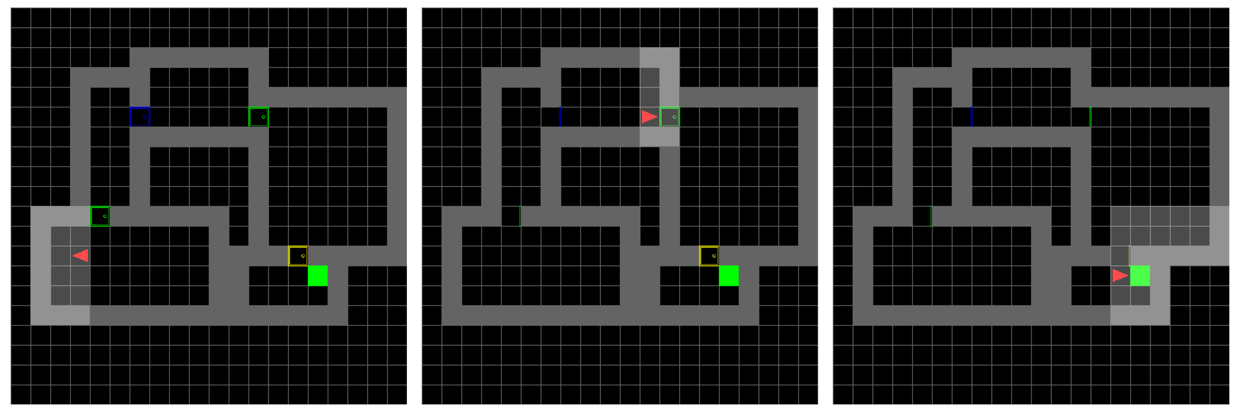


Figure 2: Different Agent Positions in 16×16 Maze

The paired frame data for the classification task is generated using the following procedure till 500000 (train), 100000 (val), and 100000 (test) frame pairs are obtained:

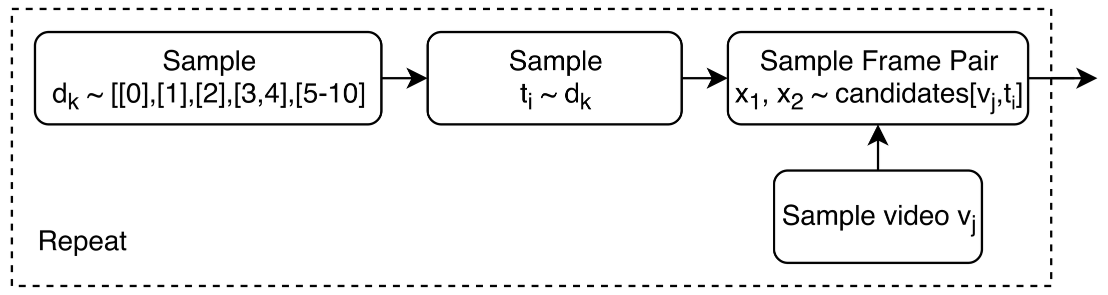


Figure 3: Data Generation Process
d_k = time bucket, t_i = time difference

Embedding Generation

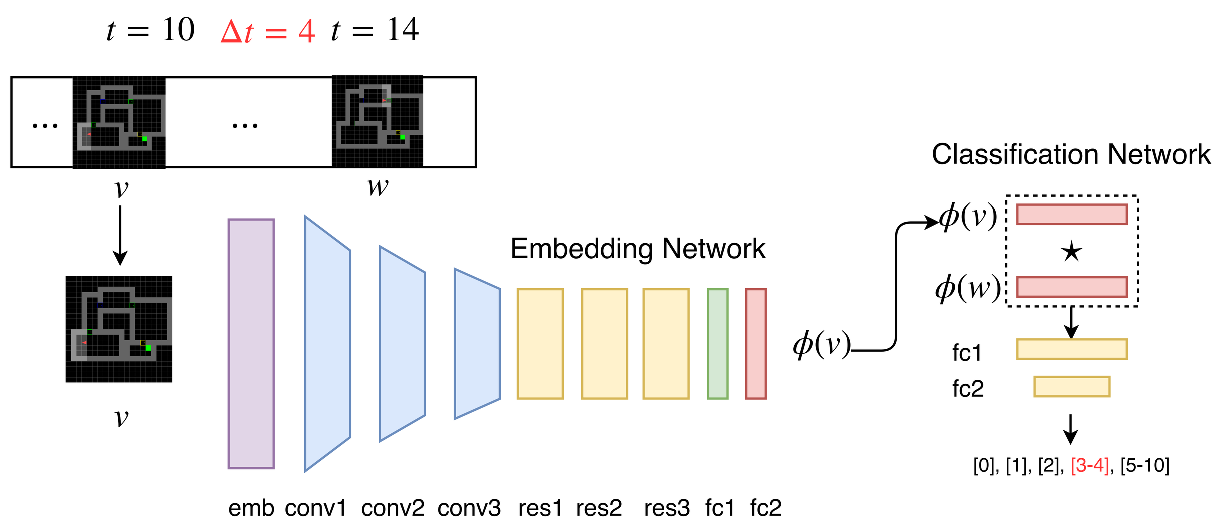


Figure 4: Embedding and Classification Networks.
Architecture from [1] with added initial Embedding (*emb*) layer

Classification Results

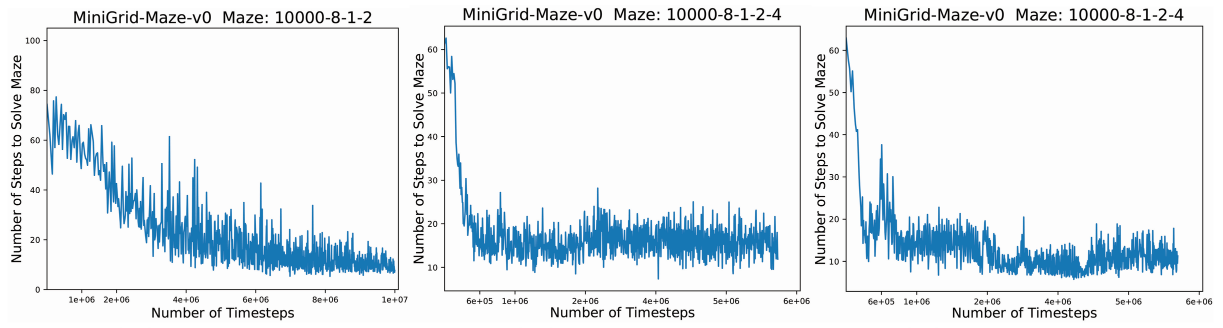
Grid Size	k	$\rho_{\text{emb,bfs}}$	$\rho_{\text{raw,bfs}}$
8×8	1.0	0.739	0.314
16×16	0.99	0.856	0.640

Table 1:
k: Fraction of Triples with Valid Triangle Inequality
$\rho_{emb,bfs}$: Pearson Correlation of (BFS, Embeddings+Cosine)
$\rho_{raw,bfs}$: Pearson Correlation of (BFS, Raw Obs+Cosine)

Grid Size	Mazes	Accuracy
8×8	1000	84%
8×8	10000	88%
16×16	1000	90%
16×16	10000	96%

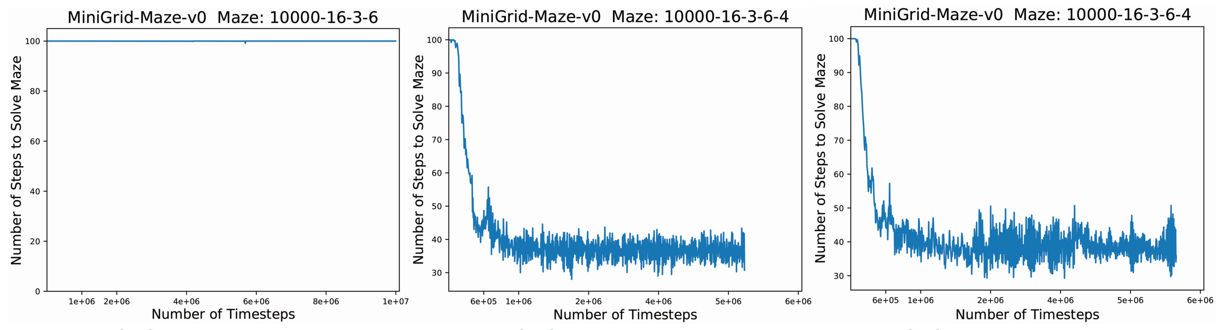
Table 2: Classification Test Accuracy for Different Configurations

Imitation Results



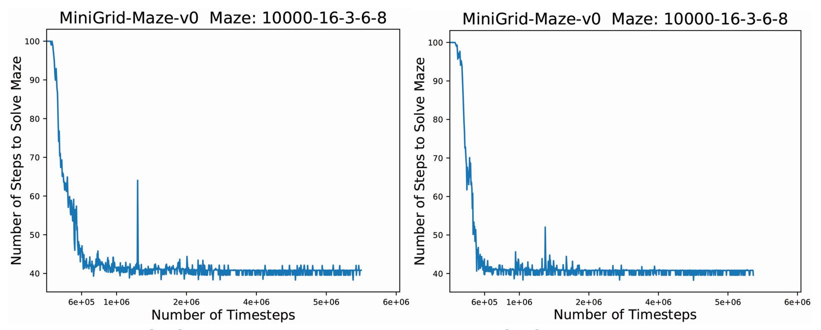
(a) Pure RL (b) IL+VAE (c) Our model

Figure 5: Avg. Steps to Solve 8×8 Maze vs. Timesteps
(Checkpoints every 4 frames)



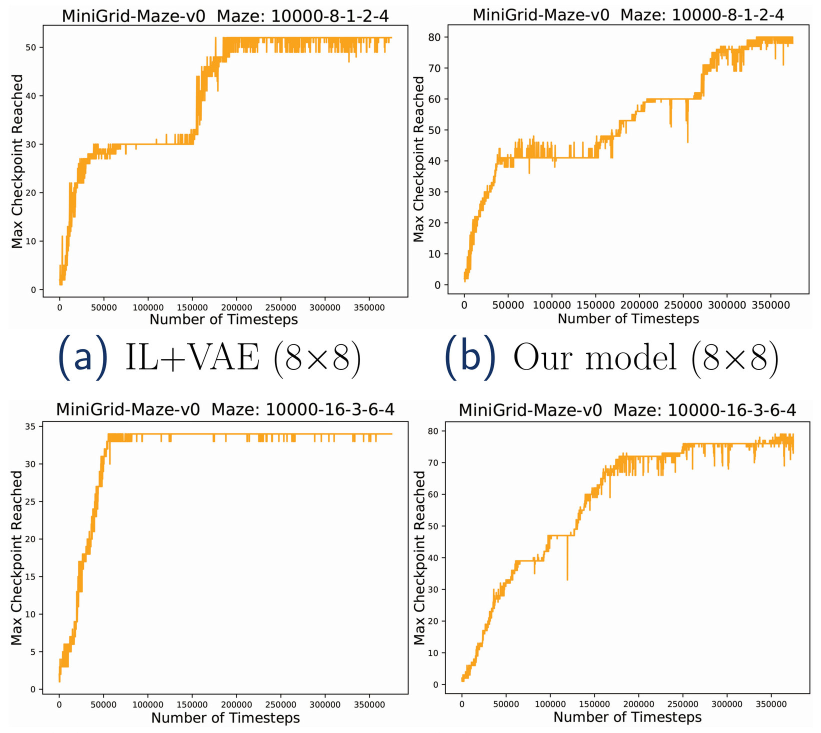
(a) Pure RL (b) IL+VAE (c) Our model

Figure 6: Avg. Steps to Solve 16×16 Maze vs. Timesteps
(Checkpoints every 4 frames)



(a) VAE (b) Our model

Figure 7: Avg. Steps to Solve 16×16 Maze vs. Timesteps
(Checkpoints every 8 frames)



(a) IL+VAE (8×8) (b) Our model (8×8)
(c) IL+VAE (16×16) (d) Our model (16×16)

Figure 8: Max. Checkpoint Crossed vs. Timesteps in Different Mazes
(Checkpoints every 4 frames)

Conclusions

- Embeddings lead to faster, better convergence especially in bigger mazes where pure RL fails
- Embedding space captures true Manhattan distance well, indicating it might be linear

Contributions

- Reversible environments necessary for [1], otherwise some checkpoints may be skipped
  - We set time limits for reaching checkpoints to avoid wasting whole episode
- [1] requires setting first checkpoint as initial state and same training and test environments to allow for checkpoints during testing
  - We propose to set closest starting state from other demonstrations to current one

Future Work

- Address domain gap with more variations in environment and extend to real-world settings
- Extend to partially observable environments

References

[1] Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas.
Playing hard exploration games by watching youtube.
In *Advances in Neural Information Processing Systems 31*, pages 2935–2945. 2018.

[2] Maxime Chevalier-Boisvert and Lucas Willems.
Minimalistic gridworld environment for openai gym.
https://github.com/maximecb/gym-minigrid, 2018.

Acknowledgements

We are immensely thankful to Dr. Rob Fergus (Associate Professor of Computer Science at Courant Institute, NYU and Director, Facebook AI Research, New York) and Ilya Kostrikov (PhD candidate, Computer Science at Courant Institute, NYU) for their guidance, valuable suggestions, and constructive feedback throughout this study. In addition, we are grateful to all CDS Capstone course instructors and staff for their support.