Semantic HELM: A Human-Readable Memory for Reinforcement Learning

Fabian Paischer [1], **Thomas Adler** [1], **Markus Hofmarcher** [2], **Sepp Hochreiter** [1]

[1] ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning,
[2] JKU LIT SAL eSPML Lab, Institute for Machine Learning,
Johannes Kepler University, Linz, Austria
`paischer@ml.jku.at`

Abstract

Reinforcement learning agents deployed in the real world often have to cope with partially observable environments. Therefore, most agents employ memory mechanisms to approximate the state of the environment. Recently, there have been impressive success stories in mastering partially observable environments, mostly in the realm of computer games like Dota 2, StarCraft II, or MineCraft. However, existing methods lack interpretability in the sense that it is not comprehensible for humans what the agent stores in its memory. In this regard, we propose a novel memory mechanism that represents past events in human language. Our method uses CLIP to associate visual inputs with language tokens. Then we feed these tokens to a pretrained language model that serves the agent as memory and provides it with a coherent and human-readable representation of the past. We train our memory mechanism on a set of partially observable environments and find that it excels on tasks that require a memory component, while mostly attaining performance on-par with strong baselines on tasks that do not. On a challenging continuous recognition task, where memorizing the past is crucial, our memory mechanism converges two orders of magnitude faster than prior methods. Since our memory mechanism is human-readable, we can peek at an agent's memory and check whether crucial pieces of information have been stored. This significantly enhances troubleshooting and paves the way toward more interpretable agents.

1 Introduction

In reinforcement learning (RL) an agent interacts with an environment and learns from feedback provided in the form of a reward function. In many applications, especially in real-world scenarios, the true state of the environment is not directly accessible to the agent, but rather approximated via observations that reveal mere parts of it. In such environments, the capability to approximate the true state by virtue of an agent's perception is crucial (Åström, 1964; Kaelbling et al., 1998). To this end, many approaches track events that occurred in the past. The brute-force strategy is to simply store all past observations. However, this is often infeasible and it is much more efficient to store more abstract representations of the history. Thus, many RL algorithms use memory mechanisms such as LSTM (Hochreiter & Schmidhuber, 1997) or Transformer (Vaswani et al., 2017) to compress sequences of high-dimensional observations. This has led to impressive successes mostly in the realm of mastering computer games on a human or even super-human level. Some examples are Dota 2 (Berner et al., 2019), StarCraft II (Vinyals et al., 2019), or MineCraft (Baker et al., 2022; Patil et al., 2022).

Most state-of-the-art methods dealing with partial observability in RL employ a memory mechanism that is not intelligible for humans. In this regard, we draw inspiration from the semantic memory
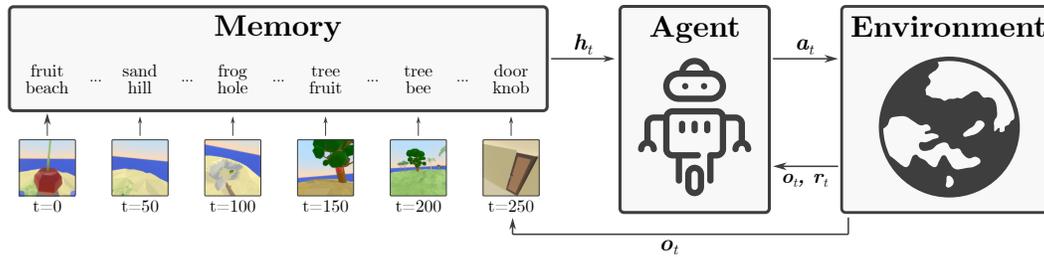
Figure 1: We add a semantic and human-readable memory to an agent to tackle partially observable RL tasks. We map visual observations o_t to the language domain via CLIP retrieval. The memory component, a pretrained language encoder, operates on text only and compresses a history of tokens into a vector h_t. The agent takes an action a_t based on the current observation o_t and the compressed history h_t.

present in humans (Yee et al., 2017) and propose to represent past events in human language. Humans memorize abstract concepts rather than every detail of information they encountered in the past (Richards & Frankland, 2017; Bowman & Zeithamova, 2018). Their ability to abstract is heavily influenced by the exposure to language in early childhood (Waxman & Markow, 1995). Further, humans use language on a daily basis to abstract and pass on information. Therefore, language is a natural choice as a representation for compounding information and has the key advantage of being human-readable. This enables analyzing if critical pieces of information have entered the memory or not. Based on this data, it becomes clear which parts of the system require refinement. Moreover, natural language has been shown to be effective as a compressed representation of past events in RL (Paischer et al., 2022a).

Our proposed method, Semantic HELM (SHELM), leverages pre-trained foundation models to construct a memory mechanism that does not require any training. We use CLIP (Radford et al., 2021) to associate visual inputs with language tokens. Thereby, the vocabulary of the CLIP language encoder serves as a semantic database of concepts from which we retrieve the closest tokens to a given observation. These tokens are passed to a pretrained language model that serves as memory and provides the agent with a coherent representation of the past.

We illustrate the benefits of a human-readable and semantic memory in partially observable RL problems. First, we conduct a qualitative analysis on whether a CLIP vision encoder is capable of extracting semantics out of synthetic environments. Then, we test SHELM on a set of partially observable 2D MiniGrid (Chevalier-Boisvert et al., 2018), and 3D MiniWorld (Chevalier-Boisvert, 2018) environments. We find that even though these environments are only partially observable, they often do not necessitate a memory mechanism as they are solvable by a memory-less policy. The MiniGrid-Memory task, however, clearly requires memory and SHELM reaches state-of-the-art performance. On more realistic 3D environments such as Avalon (Albrecht et al., 2022) and Psychlab (Leibo et al., 2018), SHELM successfully assesses the semantics of visual observations. In turn, it requires approximately 100 times fewer interaction steps than prior methods on Psychlab's continuous recognition task that explicitly evaluates for memory capacity. On Avalon, we find that the addition of our semantic memory performs on-par with the current state-of-the-art for a limited interaction budget, while adding interpretability to the memory.

2 Methods

Our goal is to express visual observations in language space such that the resulting representations become comprehensible for humans. To this end, we instantiate a mapping from images to text in the form of pretrained components which are entirely frozen during training. This way the available computational resources are invested into performance rather than interpretability. Before we describe SHELM, we briefly review the HELM framework, which serves as a starting point for our work.

2.1 Background

HELM (Paischer et al., 2022a) was proposed as a framework for RL in partially observable environments. It utilizes a pretrained language model (LM) as memory mechanism that compresses past
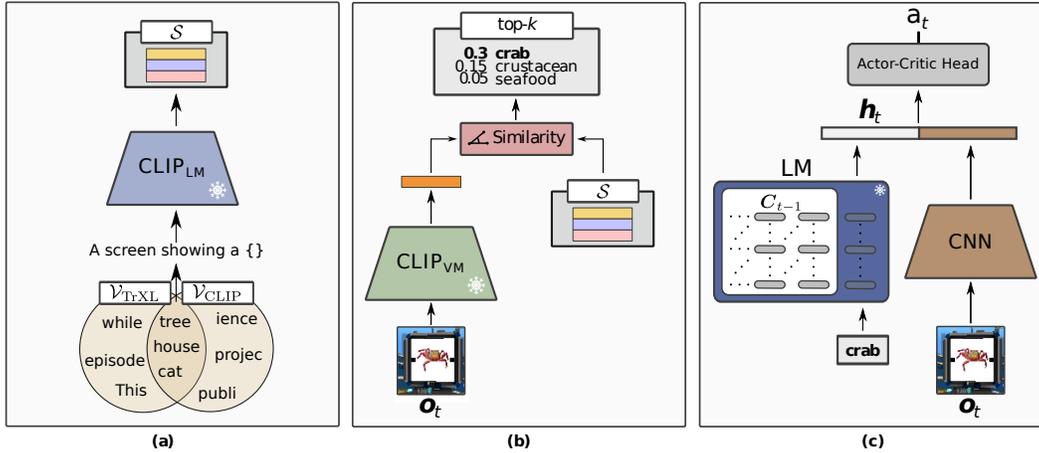
Figure 2: Architecture of SHELM. We compile a semantic database $\mathcal{S}$ by encoding prompt-augmented tokens from the overlapping vocabularies of CLIP and the LM **(a)**. Given an observation $\boldsymbol{o}_t$ we retrieve the top-k embeddings present in $\mathcal{S}$ and select their corresponding text tokens **(b)**. These tokens are passed to the LM which represents the memory module of SHELM **(c)**. C_{t-1} represents the memory cache of the LM which tracks past tokens.

observations. To map environment observations $\boldsymbol{o}_t \in \mathbb{R}^n$ to the LM, it introduces the FrozenHopfield (FH) mechanism, which performs a randomized attention over pretrained token embeddings $\boldsymbol{E} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k)^\top \in \mathbb{R}^{k \times m}$ of the LM, where k is the vocabulary size and m is the embedding dimension. Let $\boldsymbol{P} \in \mathbb{R}^{m \times n}$ be a random matrix with entries sampled independently from $\mathcal{N}(0, n/m)$. The FH mechanism performs

$$\boldsymbol{x}_t = \boldsymbol{E}^\top \operatorname{softmax}(\beta \boldsymbol{E} \boldsymbol{P} \boldsymbol{o}_t), \tag{1}$$

where β is a scaling factor that controls the dispersion of $\boldsymbol{x}_t$ within the convex hull of the token embeddings. This corresponds to a spatial compression of observations to a mixture of tokens in the LM embedding space. Since $\boldsymbol{P}$ is random, the association of observations $\boldsymbol{o}_t$ with token embeddings $\boldsymbol{e}_i$ is arbitrary, i.e., not meaningful. That is, the FH mechanism does not preserve semantics. For temporal compression, HELM leverages a pretrained LM. At time t, HELM obtains a compressed history representation by

$$\boldsymbol{h}_t = \operatorname{LM}(\boldsymbol{c}_{t-1}, \boldsymbol{x}_t) \tag{2}$$

where $\boldsymbol{c}_t$ represents the context cached in the memory register of the LM up to timestep t.

More recent work has shown that the FH mechanism is prone to representation collapse if observations are visually similar to each other (Paischer et al., 2022b,c). They propose a new method, namely HELMv2, which substitutes the random mapping with a pretrained CLIP encoder. Subsequently, they adopt a batch normalization layer (Ioffe & Szegedy, 2015) with fixed shifting and scaling parameters to transfer the image embedding to the language space. Consequently, HELMv2 computes the inputs to the LM as

$$\boldsymbol{x}_t = \operatorname{BN}_{\boldsymbol{\beta}=\boldsymbol{\mu}_E, \boldsymbol{\gamma}=\boldsymbol{\sigma}_E}(\operatorname{CLIP}_{\text{VM}}(\boldsymbol{o}_t)), \tag{3}$$

where $\operatorname{CLIP}_{\text{VM}}$ denotes the CLIP vision model and $\boldsymbol{\mu}_E$ and $\boldsymbol{\sigma}_E$ denote mean and standard deviation of the embedded vocabulary $\boldsymbol{E}$. This effectively fits the statistics of the image embeddings to those of the LM embeddings. Since the embedding spaces of CLIP and the LM were trained independently they are not semantically aligned. Therefore, also HELMv2 fails to preserve semantics of observations and, consequently, the memory mechanism of HELMv2 is not human-readable.

2.2 Semantic HELM

Semantic HELM (SHELM) inherits the high-level architecture from HELM but introduces some changes to the memory module. Similarly to HELMv2, we also use CLIP to embed environment observations. However, we replace the batch normalization layer of HELMv2 with a token-retrieval mechanism. We retrieve tokens that are similar to an observation in CLIP space and pass them to the LM in the form of text so they can be regarded as textual descriptions of environment observations.

In a first step, we determine the overlap of the token vocabularies of CLIP and the LM. This is necessary to (i) to control for the number of tokens the LM receives per observation, and (ii) to avoid loss of information due to different tokenizers used by CLIP and the LM. Thereby, we obtain a set of tokens $\mathcal{V}$. Since CLIP was pretrained on image-caption pairs, we augment each token $v \in \mathcal{V}$ with a set of pre-defined prompts $\mathcal{P} = \{p_1, p_2, \dots\}$[1]. The contents of $\mathcal{P}$ are hyperparameters of our method and can be designed to specifically target certain aspects of the different environments. We embed a token v in the CLIP output space by computing the average embedding of its prompt augmentations. That is, we define the function

$$\text{embed}(v) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \text{CLIP}_{\text{LM}}(\text{concat}(p, v)). \tag{4}$$

We do this for every $v \in \mathcal{V}$, which results in a set $\mathcal{S}$ of CLIP-embedded tokens

$$\mathcal{S} = \{\text{embed}(v)\}_{v \in \mathcal{V}}. \tag{5}$$

We denote by $\arg\max^k$ an extension of the $\arg\max$ operator that returns the subset of the k largest elements in a set. For each observation we retrieve the k most similar tokens in terms of cosine similarity by

$$\mathcal{S}^* = \arg\max_{\boldsymbol{s} \in \mathcal{S}}^{k} \text{cossim}(\boldsymbol{s}, \text{CLIP}_{\text{VM}}(\boldsymbol{o}_t)), \tag{6}$$

where

$$\text{cossim}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}. \tag{7}$$

Note that $k = |\mathcal{S}^*|$ is another hyperparameter of our method, namely the number of tokens that represent an observation. Effectively, k controls the degree of compression in the memory.

Finally, we embed single tokens $\boldsymbol{v}$ corresponding to the set of tokens in $\mathcal{S}^*$ in the LM embedding space and pass them to the LM. In this manner, we provide the LM with a textual representation of the current observation. While HELM and HELMv2 also leverage the latent structure of a pre-trained LM, they do not explicitly represent the observations in the form of text, thus, do not provide a human-readable form of past observations. Another improvement over HELMv2 is that SHELM removes the restriction of HELMv2 that the embedding spaces of CLIP and LM must have the same dimensionality. In turn, any CLIP-like encoder can be used as semantic extractor for SHELM. Figure 2 shows an illustration of the methodology of SHELM.

3 Experiments

First, we investigate in Section 3.1 whether CLIP can extract semantics of artificial scenes. Next, we train SHELM on four different environments, namely MiniGrid (Chevalier-Boisvert et al., 2018), MiniWorld (Chevalier-Boisvert, 2018), Avalon (Albrecht et al., 2022), and Psychlab (Leibo et al., 2018). We compare SHELM to HELMv2, LSTM (a recurrent baseline based on the LSTM architecture), and the popular Dreamerv2 (Hafner et al., 2021) and Dreamerv3 (Hafner et al., 2023). We show that a semantic memory boosts performance in environments that are both heavily dependent on memory and photorealistic. Finally, in Section 3.6 we perform ablation studies on the benefit of semantics and the trade-off between interpretability and performance.

We train all HELM variants and LSTM with Proximal Policy Optimization (PPO, Schulman et al., 2017) on RGB observations. Following Paischer et al. (2022a), we instantiate the LM with a pretrained TransformerXL (TrXL, Dai et al., 2019) model. For training Dreamerv2 and Dreamerv3, we use the respective codebases[2][3] and train on RGB observations. We report results via IQM (Agarwal et al., 2021) and 95% bootstrapped confidence intervals (CIs) unless mentioned otherwise. We follow (Colas et al., 2019) and perform a Welch's t-test with a reduced significance level of $\alpha = 0.025$ at the end of training to test for statistical significance. We elaborate on the architectural design and hyperparameter sweeps in Appendix F.

---

[1]Following the original implementation at https://github.com/openai/CLIP/blob/main/notebooks/Prompt_Engineering_for_ImageNet.ipynb

[2]Dreamerv2: https://github.com/danijar/dreamerv2

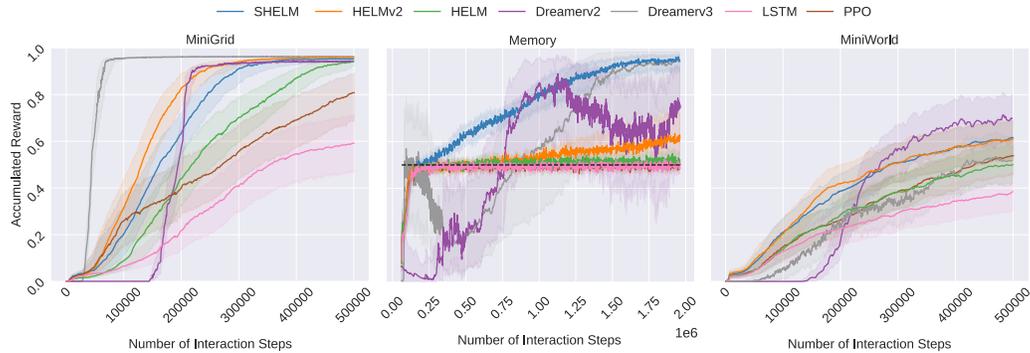[3]Dreamerv3: https://github.com/danijar/dreamerv3

Figure 3: **Left:** Accumulated reward for different methods on six MiniGrid environments, **Middle:** on the MiniGrid-Memory task, **Right:** on eight MiniWorld tasks. We report IQM and 95% CIs across 30 seeds for each method.

3.1 Extracting semantics of virtual scenes

First, we analyse whether CLIP vision encoders are able to extract semantics from artificial scenes that are typically encountered in RL environments. We compare the two smallest ResNet (He et al., 2016) and ViT (Dosovitskiy et al., 2021) architectures, namely RN50 and ViT-B/16, since those add the least amount of computational overhead to SHELM. In this regard, we look at observations sampled via a random policy, provide them to the CLIP vision encoder and retrieve the closest tokens in the CLIP embedding space as explained in Section 2.2. We find that the retrieval of tokens strongly varies between vision encoder architectures for MiniWorld and Avalon (see Figure 9 and Figure 8 in the appendix). The differences are especially prominent for MiniWorld environments, where the ViT-B/16 encoder recognizes shapes and colors, whereas RN50 ranks entirely unrelated concepts highest. More photorealistic observations from the Avalon benchmark show a similar, but less pronounced trend. We also observe a strong bias toward abstract tokens such as *screenshot*, *biome*, or *render*. We alleviate the retrieval of these tokens by including them in our prompts for retrieval. Therefore, instead of using the prompt "An image of a <tok>", we consider prompts such as "A screenshot of a <tok>", or "A biome containing a <tok>", where <tok> stands for a token in $\mathcal{V}$. We only use prompts for Avalon and Psychlab, since the effect of this prompting scheme was negligible on the other environments. Table 1 in the appendix features the full list of prompts for the two environments. We also add an analysis on using off-the-shelf captioning engines, such as BLIP-2 (Li et al., 2023) in Appendix B. We find that, while BLIP-2 correctly recognizes shapes, colors, and objects, it lacks accuracy in their compositionality. Based on this analysis we use the ViT-B/16 encoder in combination with our designed prompts as semantics extractor for SHELM.

3.2 MiniGrid

We compare all methods on a set of six partially observable grid-world environments as in as (Paischer et al., 2022a). Additionally, we train on the MiniGrid-MemoryS11-v0 environment, which we refer to as Memory. The Memory task requires the agent to match the object in the starting room to one of the two objects at the end of a corridor. The agent's view is restricted so it cannot observe both objects at the same time. Therefore, it needs to memorize the object in the starting room. If the agent chooses the wrong object at the end of the corridor, it receives no reward and the episode ends.

Figure 3 (left and middle, respectively) shows the results across the six MiniGrid environments and the Memory environment. On the MiniGrid environments, Dreamerv3 excels and converges much faster than any other method. However, final performance is approximately equal for Dreamerv3, HELMv2, and SHELM. Interestingly, although all MiniGrid environments are partially observable, they are solvable with a memory-less policy. In the Memory environment, SHELM performs on par with Dreamerv3 but exhibits faster convergence and more stable training. Figure 11 visualizes the tokens passed to the memory of SHELM after sampling episodes from a trained policy. SHELM primarily maps the ball to the token *miner*, and the key to the token *narrow*. Although the retrieved tokens do not represent the semantically correct objects in the observations, they are still stored as two different language tokens in the memory. This enables SHELM to discriminate between them and enables faster learning as mirrored in the learning curves. The LSTM baseline suffers from poor

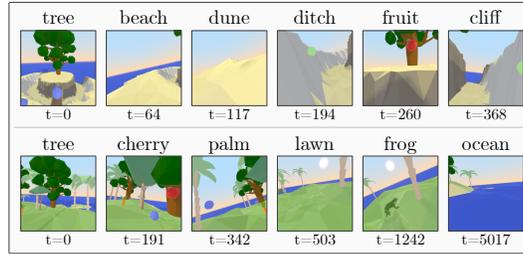| Method	Mean Human Normalized Score
HELM	0.143 ± 0.010
HELMv2	0.145 ± 0.010
PPO	0.146 ± 0.010
SHELM	0.155 ± 0.011
Dreamerv2	0.168 ± 0.012

Figure 4: **Left:** Mean and standard deviation of human normalized score across all tasks on the Avalon test worlds after 10 M timesteps. **Right:** Two sample episodes and their corresponding token mappings for episodes sampled from Avalon with a random policy.

sample efficiency and does not learn to utilize its memory within the budget of 2 M interaction steps. LSTM requires approximately 5 M steps to solve the task.

3.3 MiniWorld

The MiniWorld benchmark suite provides a set of minimalistic 3D environments. In line with Paischer et al. (2022b), we select eight different tasks from MiniWorld and train our method and all baselines on those. The tasks comprise collecting and placing objects, finding objects, or navigating through mazes. A detailed description of each task can be found in Appendix A. Prior work on the MiniWorld benchmark mostly used single environments to evaluate specialized approaches (Liu et al., 2021; Venuto et al., 2019; Zha et al., 2021; Khetarpal et al., 2020), or handcraft novel environments (Sahni et al., 2019; Hutsebaut-Buysse et al., 2020). To the best of our knowledge, HELMv2 is the only method that was evaluated on all eight tasks and has been state of the art so far (Paischer et al., 2022b).

Figure 3 (right) shows the results for all methods. Interestingly, SHELM reaches performance on par with HELMv2, even though semantics can be extracted from the 3D observations to a certain degree (as explained in Section 3.1). Further, Dreamerv2 outperforms both and reaches state-of-the-art performance. Surprisingly, Dreamerv3 attains a significantly lower score and performs on par with HELM and the memory-less PPO. This might be due to suboptimal hyperparameters, even though Hafner et al. (2023) claim that the choice of hyperparameters transfers well across environments. LSTM again suffers from poor sample efficiency reaching the lowest scores out of all compared methods. Finally, we again observe that PPO can in principle solve all tasks, which yields further evidence that partial observability does not automatically imply necessity for memory. Our results suggest that a memory mechanism can result in enhanced sample efficiency (SHELM vs PPO). However, memory is not imperative to solve these tasks.

3.4 Avalon

Avalon is an open-ended 3D survival environment consisting of 16 different tasks. The aim for the agent is to survive as long as possible by defending against predators, hunting animals, and eating food in order to restore energy. An episode ends if the agent has no energy left. This can happen if the agent receives environmental damage (e.g. from falling), is being killed by a predator, or does not eat frequently. The agent receives a dense reward as the difference in energy between consecutive timesteps. Additionally, it receives a delayed reward upon successful completion of a task, e.g., eating food. The observation space are RGBD images as well as proprioceptive input that comprise, e.g., the agent's energy. We adopt the same architecture and training strategy as Albrecht et al. (2022) for all HELM variants. Specifically, we add the history compression branch to the baseline PPO agent and train on all 16 tasks including their difficulty curriculum. The history compression branch is only based on RGB images and does not receive the proprioceptive state. We also train Dreamerv2, and a memory-less baseline (PPO), which is identical to the PPO baseline in Albrecht et al. (2022). Due to computational constraints and since we observed superior performance of Dreamerv2 on 3D environments, we neglect Dreamerv3 and LSTM and train on a limited budget of 10 M interaction steps. We elaborate on training details and hyperparameter selection in Appendix F. The final performance of an agent is measured in terms of mean human normalized scores on a curated set of 1000 test worlds.
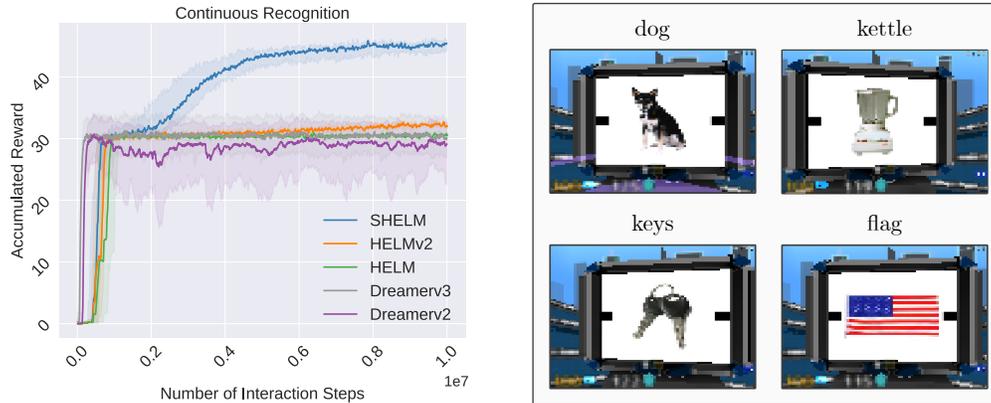
Figure 5: Performance for all methods on CR task from Psychlab. We report IQM and 95% CIs across 5 seeds (**left**). Observation and corresponding tokens for SHELM on CR environment (**right**).

The results are shown in Figure 4, left. For detailed results per task see Table 2 in the appendix. SHELM and Dreamerv2 yield the highest performance on average after 10 M interaction steps. However, the difference to the memory-less PPO is not statistically significant. To further investigate this finding we train PPO for 50 M interaction steps and compare it to the baseline results reported in Albrecht et al. (2022) in Appendix D. Indeed, we find that our memory-less baseline attains scores on-par with memory-based approaches trained for 50 M interaction steps. This yields further evidence that Avalon does not necessitate the addition of a memory mechanism. Finally, we can glance at SHELM's memory to identify failure cases. We show observations of two episodes and their token correspondences for SHELM in Figure 4, right. The observations are mostly mapped to semantically meaningful tokens that represent parts in the image observations. However, we also observe some cases where CLIP retrieves tokens that are semantically related, but not correct, which we show in Figure 12 in the appendix.

3.5 PsychLab

The Psychlab environment suite (Leibo et al., 2018) consists of 8 complex tasks and was designed to isolate various cognitive faculties of RL agents including vision and memory. The continuous recognition (CR) task of Psychlab was specifically designed to target the memory capacity of an agent. In this task the agent is placed in front of a monitor which displays a stream of objects. The agent then needs to identify whether it has already seen a particular object by swiping either left or right. If the agent correctly identifies an object it receives a reward of +1. A policy that always swipes in one direction achieves a reward of around 30, which we refer to as the random baseline. Every reward higher than that indicates that an agent actually utilizes its memory. This task is ideal to evaluate the memory capacity of an agent, since episodes last for about 2000 timesteps and usually require remembering up to 50 different objects.

We train all methods for 10 M interaction steps on the CR task (Figure 5). We neglect the memory-less baseline because this task is unsolvable without memory. SHELM indeed learns to effectively utilize its memory within 10 M interaction steps and significantly outperforms all competitors. Other approaches require interaction steps in the range of billions until convergence (Parisotto et al., 2020; Fortunato et al., 2019). These works report human normalized scores which are not publicly available, therefore we cannot compare SHELM to those. HELMv2 is the only competitor that attains a performance better than random, but reaches significantly lower scores than SHELM. Surprisingly, both variants of Dreamer do not exceed performance of the random baseline. Finally, we inspect the memory of SHELM and show the token mappings that are passed on to the memory module for some sample observations in Figure 5. In most cases SHELM assigns semantically correct tokens to the displayed objects. However, we also show some cases where the token retrieval of SHELM conflates different objects in Figure 13. We find that this can mostly be attributed to the low resolution of observations. SHELM can recover from these failure cases when using higher resolutions (see Figure 14).

3.6 Ablation studies

Are semantics important? We slightly alter the HELMv2 implementation from (Paischer et al., 2022b,c) by retrieving the closest tokens in the language space after their frozen batch normalization layer, and finally passing those to the LM. This setting is similar to SHELM in that the LM receives tokens in the form of text. However, semantics are not preserved since visual features are merely shifted to the language space. We call this setting HELMv2-Ret and train it on the Memory environment (see Figure 6). We find that if the mapping from observation to language is arbitrary, the performance decreases drastically.

Is it important to learn task-specific features? SHELM would be even more interpretable if not only the memory module but also the branch processing the current observation (CNN in Figure 2) could utilize language tokens. Therefore, we substitute the CNN with a CLIP vision encoder (SHELM-CLIP). An important consequence of this methodological change is that even features from the current observation can be interpreted by our retrieval mechanism. However, Figure 6 suggests that it is vital to learn task-specific features.

Should the history branch operate on task-specific features? To answer this question we substitute the CLIP encoder in the history branch with the CNN encoder learned at the current timestep (SHELM-CNN). An immediate drawback of that is the loss of readability, since we do not represent the observations in text form anymore. Further, we again observe a drastic drop in performance. The reason for this is that the task-specific features must be learned before they actually provide any benefit to the history branch. Thus, we prefer to keep abstract semantic features in text form since that does not require any training and has the additional benefit of being human-readable.
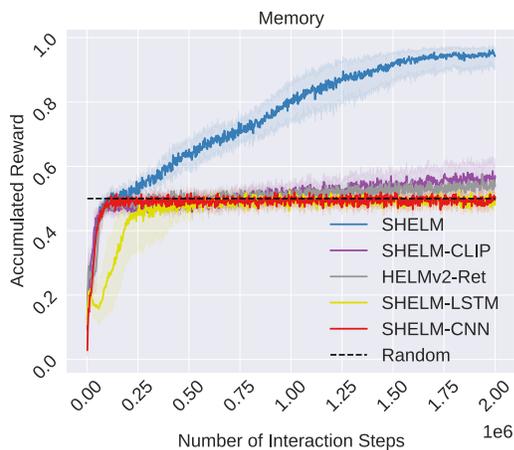

Figure 6: Ablation study on the effect of semantics, the influence of task-specific features in the history and the current timestep, and the importance of the pretrained LM. We report IQM and 95% CIs across 30 seeds on the MiniGrid-Memory task.

Is a pretrained language encoder required? We replace the LM with an LSTM operating on the text tokens (SHELM-LSTM). This setting resulted in performance equivalent to randomly choosing an object at the end of the corridor. However, we believe that after longer training SHELM-LSTM can eventually learn to solve the task, since the LSTM baseline also solved the task after longer trainng. Thus, the simpler and more sample efficient method is to maintain the frozen pretrained encoder instead of learning a compressed history representation.

We show additional results for an ablation study where we exchange the ViT-B/16 vision encoder with a RN50 in Appendix E. SHELM appears to be quite sensitive to the selection of vision encoder, which corroborates our qualitative findings in Section 3.1.

4 Related work

RL and partial observability RL with incomplete state information can necessitate a memory for storing the history of observations an agent encountered. However, storing the entire history is often infeasible. History Compression tries to answer the question of what information to store in a stream of observations (Schmidhuber, 1992; Schmidhuber et al., 1993; Zenke et al., 2017; Kirkpatrick et al., 2016; Schwarz et al., 2018; Ruvolo & Eaton, 2013). A plethora of prior works have used history compression to tackle credit assignment (Arjona-Medina et al., 2019; Patil et al., 2022; Widrich et al., 2021; Holzleitner et al., 2021), and partial observability (Hausknecht & Stone, 2015; Vinyals et al., 2019; Berner et al., 2019; Pleines et al., 2022). The memory maintained by an agent can either

be external (Hill et al., 2021; Wayne et al., 2018), or directly integrated into the feature extraction pipeline via the network architecture. An orthogonal approach for history compression is training recurrent dynamics models (Ha & Schmidhuber, 2018; Pasukonis et al., 2022; Hafner et al., 2020, 2021). We believe language is very well suited as medium for compression to summarize past events, as suggested by Paischer et al. (2022a).

Language in RL   Language provides useful abstractions for RL. These abstractions can be leveraged to represent high-level skills (Sharma et al., 2022; Jiang et al., 2019; Jacob et al., 2021). LMs have been used to improve exploration in text-based environments (Yao et al., 2020), or in visual environments via a language oracle (Mu et al., 2022). Pretrained vision-language models (VLMs) provide abstract embedding spaces that can be used for exploration in visual environments (Tam et al., 2022). We leverage VLMs to spatially compress visual observations to language tokens. Furthermore, pretrained LMs were leveraged for (i) initializing policies in text-based environments (Li et al., 2022), (ii) grounding to various environments (Andreas et al., 2018; Huang et al., 2023; Carta et al., 2023; Hill et al., 2021), (iii) sequence modeling in the offline RL setup (Reid et al., 2022), and (iv) generating high-level plans (Huang et al., 2022a,b; Wang et al., 2023; Singh et al., 2022; Ichter et al., 2022; Liang et al., 2022a; Dasgupta et al., 2023; Du et al., 2023; Shah et al., 2022; Ahn et al., 2022; Zeng et al., 2022). Other works train agents to generate descriptions of virtual scenes (Yan et al., 2022), or thought processes of humans (Hu & Clune, 2023). Additionally, language has been used for augmenting the reward function (Wang et al., 2019; Bahdanau et al., 2019; Goyal et al., 2019; Carta et al., 2022; Kwon et al., 2023), or learning a dynamics model (Zhong et al., 2022, 2021, 2020; Wu et al., 2023). To manage a language-based memory, Park et al. (2023) stores a record of an agent's memory which comprises different levels of abstraction. Importantly, all agent-environment interactions in their work are scripted, while our agent enables a memory based on language for visual inputs.

Language for interpretability   In the realm of supervised learning, a plethora of prior works had used language as human-interpretable medium to explain classification decisions in computer vision (Hendricks et al., 2016, 2018; Park et al., 2018; Zellers et al., 2019; Hernandez et al., 2022), or in natural language processing (Andreas & Klein, 2017; Zaidan & Eisner, 2008; Camburu et al., 2018; Rajani et al., 2019; Narang et al., 2020). Interpretability methods in RL are scarce and usually follow a post-hoc approach (Puiutta & Veith, 2020). Intrinsically interpretable models are designed to be inherently interpretable even during training time and are preferable over post-hoc approaches (Rudin et al., 2021). They often restrict the complexity of the model class, which in turn results in reduced performance of the agent. Therefore, (Glanois et al., 2021) propose to adopt a modular approach to interpretability. To this end, our work focuses on the memory module. This enables us to provide some extent of intrinsic interpretability while exceeding performance of existing (non-interpretable) methods on tasks that necessitate memory.

Foundation models   The advent of the Transformer architecture (Vaswani et al., 2017) gave rise to foundation models (FMs, Bommasani et al., 2021), such as GPT-3 (Brown et al., 2020). As shown by Petroni et al. (2019); Talmor et al. (2020); Kassner et al. (2020); Mahowald et al. (2023), pretrained LMs can learn abstract symbolic rules and show sparks of reasoning. We leverage their abstraction capabilities for history compression in RL. Further, vision FMs have been demonstrated to be well adaptable to foreign domains (Adler et al., 2020; Evci et al., 2022; Ostapenko et al., 2022; Parisi et al., 2022). Our approach combines language-based FMs with vision-langauge models, such as CLIP (Radford et al., 2021) or ALIGN (Jia et al., 2021). We use CLIP to obtain language tokens that semantically correspond to concepts present in synthetic environments.

5   Limitations

Token-level abstraction   One potential shortcoming of our method is that our retrieval is based on single tokens. However, we have shown that for environments where one or a few objects are important, this is sufficient. Further, our approach is very flexible and the semantic database can easily be augmented with more detailed descriptions of objects and their relationships. We aim to investigate more in this direction in future work.

Wall clock time   A current limitation of SHELM is wall clock time. The rollout phase is particularly expensive since each timestep needs to be propagated through the LM. Despite that, it is still more efficient than, e.g., Dreamerv2. This is due to the fact that the memory mechanism is kept frozen and need not be updated durint the training phase. A potential solution for decreasing the complexity of the rollout phase would be distillation of the LM into smaller actor networks as in (Parisotto & Salakhutdinov, 2021).

Modality gap   Our semantic mapping is limited by the inherent ability of CLIP vision encoders to extract semantically meaningful features in synthetic environments. However, CLIP suffers from the modality gap (Liang et al., 2022b), i.e., a mis-alignment between image and text embedding spaces. In the future, we aim at incorporating methods that mitigate the modality gap (Fürst et al., 2022; Ouali et al., 2023).

Distribution shift   We use a pretrained CLIP model to retrieve tokens that are similar to visual observations. However, CLIP was pretrained on large-scale data crawled from the web. We have shown that it can still extract semantics of synthetic environments when those are sufficiently photorealistic, i.e., MiniWorld, Avalon, or Psychlab. Further, prompting can enhance the quality of the retrieval. For environments such as MiniGrid, the retrieval yields tokens that do not correspond to aspects in the image. However, Gupta et al. (2022) has shown that CLIP can handle task-specific texts for MiniGrid, thus, we would like to investigate the effect of augmenting our semantic database with such texts in the future.

6   Conclusion

In many real-world scenarios an agent requires a memory mechanism to deal with partial observability. Current memory mechanisms in RL act as a black box where it is not comprehensible for humans what pieces of information were stored. To solve this problem, we proposed a new method called Semantic HELM that represents past events in form of human-readable language by leveraging pretrained vision-language models. We showed compelling evidence that even for synthetic environments our memory mechanism can extract semantics from visual observations. Further, SHELM outperforms strong baselines on photorealistic memory-dependent environments through its human-readable memory module. In cases where SHELM fails to extract semantics from observations, we can investigate the cause by inspection of the memory module. Even in such cases, SHELM mostly performs on-par with other memory-based approaches.

We believe that we can further enhance our method by generating full captions from history observations instead of only a small number of words. This could enable (i) a long-term memory that textually summarizes all captions of corresponding observations, (ii) a potential for planning in form of text from visual observations similar to Patel et al. (2023), and (iii) modeling dynamics of an environment in language space.

References

Adler, T., Brandstetter, J., Widrich, M., Mayr, A., Kreil, D. P., Kopp, M., Klambauer, G., and Hochreiter, S. Cross-Domain Few-Shot Learning by Representation Fusion. *CoRR*, abs/2010.06498, 2020. arXiv: 2010.06498.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. G. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29304–29320, 2021.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., and Zeng, A. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

Albrecht, J., Fetterman, A. J., Fogelman, B., Kitanidis, E., Wróblewski, B., Seo, N., Rosenthal, M., Knutins, M., Polizzi, Z., Simon, J. B., and Qiu, K. Avalon: A benchmark for RL generalization using procedurally generated worlds. *CoRR*, abs/2210.13417, 2022. doi: 10.48550/arXiv.2210. 13417.

Andreas, J. and Klein, D. Analogs of linguistic structure in deep representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2893–2897, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10. 18653/v1/D17-1311.

Andreas, J., Klein, D., and Levine, S. Learning with Latent Language. In Walker, M. A., Ji, H., and Stent, A. (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 2166–2179. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1197.

Arjona-Medina, J. A., Gillhofer, M., Widrich, M., Unterthiner, T., Brandstetter, J., and Hochreiter, S. RUDDER: Return Decomposition for Delayed Rewards. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13544–13555, 2019.

Bahdanau, D., Hill, F., Leike, J., Hughes, E., Hosseini, S. A., Kohli, P., and Grefenstette, E. Learning to Understand Goal Specifications by Modelling Reward. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Baker, B., Akkaya, I., Zhokhov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video pretraining (VPT): learning to act by watching unlabeled online videos. *CoRR*, abs/2206.11795, 2022. doi: 10.48550/arXiv.2206.11795.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., Pinto, H. P. d. O., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. Dota 2 with Large Scale Deep Reinforcement Learning. *CoRR*, abs/1912.06680, 2019.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., Arx, S. v., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., and al, e. On the Opportunities and Risks of Foundation Models. *CoRR*, abs/2108.07258, 2021. arXiv: 2108.07258.

Bowman, C. R. and Zeithamova, D. Abstract Memory Representations in the Ventromedial Prefrontal Cortex and Hippocampus Support Concept Generalization. *Journal of Neuroscience*, 38(10): 2605–2614, 2018. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2811-17.2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Camburu, O., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. e-snli: Natural language inference with natural language explanations. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9560–9572, 2018.

Carta, T., Lamprier, S., Oudeyer, P., and Sigaud, O. EAGER: asking and answering questions for automatic reward shaping in language-guided RL. *CoRR*, abs/2206.09674, 2022. doi: 10.48550/arXiv.2206.09674.

Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., and Oudeyer, P. Grounding large language models in interactive environments with online reinforcement learning. *CoRR*, abs/2302.02662, 2023. doi: 10.48550/arXiv.2302.02662.

Chevalier-Boisvert, M. MiniWorld: Minimalistic 3D Environment for RL & Robotics Research, 2018. Publication Title: GitHub repository.

Chevalier-Boisvert, M., Willems, L., and Pal, S. Minimalistic Gridworld Environment for OpenAI Gym, 2018. Publication Title: GitHub repository.

Colas, C., Sigaud, O., and Oudeyer, P. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. In *Reproducibility in Machine Learning, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1285.

Dasgupta, I., Kaeser-Chen, C., Marino, K., Ahuja, A., Babayan, S., Hill, F., and Fergus, R. Collaborating with language models for embodied reasoning. *CoRR*, abs/2302.00763, 2023. doi: 10.48550/arXiv.2302.00763.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Du, Y., Watkins, O., Wang, Z., Colas, C., Darrell, T., Abbeel, P., Gupta, A., and Andreas, J. Guiding pretraining in reinforcement learning with large language models. *CoRR*, abs/2302.06692, 2023. doi: 10.48550/arXiv.2302.06692.

Evci, U., Dumoulin, V., Larochelle, H., and Mozer, M. C. Head2Toe: Utilizing Intermediate Representations for Better Transfer Learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6009–6033. PMLR, 2022.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. *CoRR*, abs/2206.08853, 2022. doi: 10.48550/arXiv.2206.08853. arXiv: 2206.08853.

Fortunato, M., Tan, M., Faulkner, R., Hansen, S., Badia, A. P., Buttimore, G., Deck, C., Leibo, J. Z., and Blundell, C. Generalization of reinforcement learners with working and episodic memory. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 12448–12457, 2019.

Fürst, A., Rumetshofer, E., Lehner, J., Tran, V. T., Tang, F., Ramsauer, H., Kreil, D. P., Kopp, M. K., Klambauer, G., Bitto-Nemling, A., and Hochreiter, S. CLOOB: Modern hopfield networks with infoLOOB outperform CLIP. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Glanois, C., Weng, P., Zimmer, M., Li, D., Yang, T., Hao, J., and Liu, W. A survey on interpretable reinforcement learning. *CoRR*, abs/2112.13112, 2021.

Goyal, P., Niekum, S., and Mooney, R. J. Using Natural Language for Reward Shaping in Reinforcement Learning. In Kraus, S. (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 2385–2391. ijcai.org, 2019. doi: 10.24963/ijcai.2019/331.

Gupta, T., Karkus, P., Che, T., Xu, D., and Pavone, M. Foundation models for semantic novelty in reinforcement learning. *CoRR*, abs/2211.04878, 2022. doi: 10.48550/arXiv.2211.04878.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *NeurIPS*, pp. 2455–2467, 2018.

Hafner, D., Lillicrap, T. P., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *ICLR*. OpenReview.net, 2020.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Hausknecht, M. J. and Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. In *2015 AAAI Fall Symposia, Arlington, Virginia, USA, November 12-14, 2015*, pp. 29–37. AAAI Press, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. Generating visual explanations. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 3–19. Springer, 2016. doi: 10.1007/978-3-319-46493-0_1.

Hendricks, L. A., Hu, R., Darrell, T., and Akata, Z. Grounding visual explanations. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pp. 269–286. Springer, 2018. doi: 10.1007/978-3-030-01216-8_17.

Hernandez, E., Schwettmann, S., Bau, D., Bagashvili, T., Torralba, A., and Andreas, J. Natural language descriptions of deep visual features. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Hill, F., Tieleman, O., Glehn, T. v., Wong, N., Merzic, H., and Clark, S. Grounded Language Learning Fast and Slow. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Holzleitner, M., Gruber, L., Arjona-Medina, J. A., Brandstetter, J., and Hochreiter, S. Convergence Proof for Actor-Critic Methods Applied to PPO and RUDDER. *Trans. Large Scale Data Knowl. Centered Syst.*, 48:105–130, 2021. doi: 10.1007/978-3-662-63519-3_5.

Hu, S. and Clune, J. Thought cloning: Learning to think while acting by imitating human thinking. *CoRR*, abs/2306.00323, 2023. doi: 10.48550/arXiv.2306.00323.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9118–9147. PMLR, 2022a.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Jackson, T., Brown, N., Luu, L., Levine, S., Hausman, K., and Ichter, B. Inner monologue: Embodied reasoning through planning with language models. In Liu, K., Kulic, D., and Ichnowski, J. (eds.), *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2022b.

Huang, W., Xia, F., Shah, D., Driess, D., Zeng, A., Lu, Y., Florence, P., Mordatch, I., Levine, S., Hausman, K., and Ichter, B. Grounded decoding: Guiding text generation with grounded models for robot control. *CoRR*, abs/2303.00855, 2023. doi: 10.48550/arXiv.2303.00855.

Hutsebaut-Buysse, M., Mets, K., and Latré, S. Pre-trained word embeddings for goal-conditional transfer learning in reinforcement learning. *CoRR*, abs/2007.05196, 2020.

Ichter, B., Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., Kalashnikov, D., Levine, S., Lu, Y., Parada, C., Rao, K., Sermanet, P., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Yan, M., Brown, N., Ahn, M., Cortes, O., Sievers, N., Tan, C., Xu, S., Reyes, D., Rettinghouse, J., Quiambao, J., Pastor, P., Luu, L., Lee, K., Kuang, Y., Jesmonth, S., Joshi, N. J., Jeffrey, K., Ruano, R. J., Hsu, J., Gopalakrishnan, K., David, B., Zeng, A., and Fu, C. K. Do as I can, not as I say: Grounding language in robotic affordances. In Liu, K., Kulic, D., and Ichnowski, J. (eds.), *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2022.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456. JMLR.org, 2015.

Jacob, A. P., Lewis, M., and Andreas, J. Multitasking inhibits semantic drift. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 5351–5366. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.421.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021.

Jiang, Y., Gu, S. S., Murphy, K. P., and Finn, C. Language as an Abstraction for Hierarchical Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, 101:99–134, 1998.

Kassner, N., Krojer, B., and Schütze, H. Are Pretrained Language Models Symbolic Reasoners over Knowledge? In Fernández, R. and Linzen, T. (eds.), *Proceedings of the 24th Conference on Computational Natural Language Learning, CoNLL 2020, Online, November 19-20, 2020*, pp. 552–564. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.conll-1.45.

Khetarpal, K., Klissarov, M., Chevalier-Boisvert, M., Bacon, P., and Precup, D. Options of interest: Temporal abstraction with interest functions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4444–4451. AAAI Press, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

Kwon, M., Xie, S. M., Bullard, K., and Sadigh, D. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Leibo, J. Z., de Masson d'Autume, C., Zoran, D., Amos, D., Beattie, C., Anderson, K., Castañeda, A. G., Sanchez, M., Green, S., Gruslys, A., Legg, S., Hassabis, D., and Botvinick, M. M. Psychlab: A psychology laboratory for deep reinforcement learning agents. *CoRR*, abs/1801.08116, 2018.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

Li, S., Puig, X., Paxton, C., Du, Y., Wang, C., Fan, L., Chen, T., Huang, D.-A., Akyürek, E., Anandkumar, A., Andreas, J., Mordatch, I., Torralba, A., and Zhu, Y. Pre-Trained Language Models for Interactive Decision-Making. *CoRR*, abs/2202.01771, 2022. arXiv: 2202.01771.

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. Code as policies: Language model programs for embodied control. *CoRR*, abs/2209.07753, 2022a. doi: 10.48550/arXiv.2209.07753.

Liang, W., Zhang, Y., Kwon, Y., Yeung, S., and Zou, J. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b.

Liu, E. Z., Raghunathan, A., Liang, P., and Finn, C. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6925–6935. PMLR, 2021.

Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., and Fedorenko, E. Dissociating language and thought in large language models: a cognitive perspective, 2023.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Mu, J., Zhong, V., Raileanu, R., Jiang, M., Goodman, N. D., Rocktäschel, T., and Grefenstette, E. Improving Intrinsic Exploration with Language Abstractions. *CoRR*, abs/2202.08938, 2022. arXiv: 2202.08938.

Narang, S., Raffel, C., Lee, K., Roberts, A., Fiedel, N., and Malkan, K. Wt5?! training text-to-text models to explain their predictions. *CoRR*, abs/2004.14546, 2020.

Ostapenko, O., Lesort, T., Rodríguez, P., Arefin, M. R., Douillard, A., Rish, I., and Charlin, L. Foundational Models for Continual Learning: An Empirical Study of Latent Replay. *arXiv preprint arXiv:2205.00329*, 2022.

Ouali, Y., Bulat, A., Martinez, B., and Tzimiropoulos, G. Black box few-shot adaptation for vision-language models, 2023.

Paischer, F., Adler, T., Patil, V. P., Bitto-Nemling, A., Holzleitner, M., Lehner, S., Eghbal-Zadeh, H., and Hochreiter, S. History compression via language models in reinforcement learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17156–17185. PMLR, 2022a.

Paischer, F., Adler, T., Radler, A., Hofmarcher, M., and Hochreiter, S. Foundation models for history compression in reinforcement learning. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022b.

Paischer, F., Adler, T., Radler, A., Hofmarcher, M., and Hochreiter, S. Toward semantic history compression for reinforcement learning. In *Second Workshop on Language and Reinforcement Learning*, 2022c.

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. The Unsurprising Effectiveness of Pre-Trained Vision Models for Control. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17359–17371. PMLR, 2022.

Parisotto, E. and Salakhutdinov, R. R. Efficient Transformers in Reinforcement Learning using Actor-Learner Distillation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., Botvinick, M., Heess, N., and Hadsell, R. Stabilizing Transformers for Reinforcement Learning. In *International Conference on Machine Learning*, pp. 7487–7498. PMLR, November 2020.

Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., and Rohrbach, M. Multimodal explanations: Justifying decisions and pointing to the evidence. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8779–8788. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00915.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior, 2023.

Pasukonis, J., Lillicrap, T., and Hafner, D. Evaluating long-term memory in 3d mazes, 2022.

Patel, D., Eghbalzadeh, H., Kamra, N., Iuzzolino, M. L., Jain, U., and Desai, R. Pretrained language models as visual planners for human assistance, 2023.

Patil, V. P., Hofmarcher, M., Dinu, M.-C., Dorfer, M., Blies, P. M., Brandstetter, J., Arjona-Medina, J. A., and Hochreiter, S. Align-RUDDER: Learning From Few Demonstrations by Reward Redistribution. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17531–17572. PMLR, 2022.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P. S. H., Bakhtin, A., Wu, Y., and Miller, A. H. Language Models as Knowledge Bases? In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2463–2473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1250.

Pleines, M., Pallasch, M., Zimmer, F., and Preuss, M. Generalization, Mayhems and Limits in Recurrent Proximal Policy Optimization. *CoRR*, abs/2205.11104, 2022. doi: 10.48550/arXiv.2205.11104. arXiv: 2205.11104.

Puiutta, E. and Veith, E. M. S. P. Explainable Reinforcement Learning: A Survey. In Holzinger, A., Kieseberg, P., Tjoa, A. M., and Weippl, E. (eds.), *Machine Learning and Knowledge Extraction*, Lecture Notes in Computer Science, pp. 77–95, Cham, 2020. Springer International Publishing. ISBN 978-3-030-57321-8. doi: 10.1007/978-3-030-57321-8_5.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Rajani, N. F., McCann, B., Xiong, C., and Socher, R. Explain yourself! leveraging language models for commonsense reasoning. In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4932–4942. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1487.

Reid, M., Yamada, Y., and Gu, S. S. Can Wikipedia Help Offline Reinforcement Learning? *CoRR*, abs/2201.12122, 2022. arXiv: 2201.12122.

Richards, B. A. and Frankland, P. W. The Persistence and Transience of Memory. *Neuron*, 94(6): 1071–1084, 2017. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2017.04.037.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges, July 2021.

Ruvolo, P. and Eaton, E. ELLA: An Efficient Lifelong Learning Algorithm. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 507–515. JMLR.org, 2013.

Sahni, H., Buckley, T., Abbeel, P., and Kuzovkin, I. Addressing sample complexity in visual tasks using HER and hallucinatory gans. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5823–5833, 2019.

Schmidhuber, J. Learning Complex, Extended Sequences Using the Principle of History Compression. *Neural Comput.*, 4(2):234–242, 1992. doi: 10.1162/neco.1992.4.2.234.

Schmidhuber, J., Mozer, M. C., and Prelinger, D. Continuous history compression. In Hüning, H., Neuhauser, S., Raus, M., and Ritschel, W. (eds.), *Proc. of Intl. Workshop on Neural Networks, RWTH Aachen*, pp. 87–95. Augustinus, 1993.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347, 2017. arXiv: 1707.06347.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & Compress: A scalable framework for continual learning. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4535–4544. PMLR, 2018.

Shah, D., Osiński, B., brian ichter, and Levine, S. LM-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *6th Annual Conference on Robot Learning*, 2022.

Sharma, P., Torralba, A., and Andreas, J. Skill induction and planning with latent language. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 1713–1726. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.120.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. Progprompt: Generating situated robot task plans using large language models. *CoRR*, abs/2209.11302, 2022. doi: 10.48550/arXiv.2209.11302.

Talmor, A., Elazar, Y., Goldberg, Y., and Berant, J. oLMpics - On what Language Model Pre-training Captures. *Trans. Assoc. Comput. Linguistics*, 8:743–758, 2020.

Tam, A. C., Rabinowitz, N. C., Lampinen, A. K., Roy, N. A., Chan, S. C. Y., Strouse, D. J., Wang, J. X., Banino, A., and Hill, F. Semantic Exploration from Language Abstractions and Pretrained Representations. *CoRR*, abs/2204.05080, 2022. doi: 10.48550/arXiv.2204.05080. arXiv: 2204.05080.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Venuto, D., Boussioux, L., Wang, J., Dali, R., Chakravorty, J., Bengio, Y., and Precup, D. Avoidance learning using observational reinforcement learning. *CoRR*, abs/1909.11228, 2019.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nat.*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z.

Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.-F., Wang, W. Y., and Zhang, L. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6629–6638. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00679.

Wang, Z., Cai, S., Liu, A., Ma, X., and Liang, Y. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents, 2023.

Waxman, S. R. and Markow, D. Words as Invitations to Form Categories: Evidence from 12- to 13-Month-Old Infants. *Cognitive Psychology*, 29:257–302, 1995.

Wayne, G., Hung, C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J. W., Mirowski, P., Leibo, J. Z., Santoro, A., Gemici, M., Reynolds, M., Harley, T., Abramson, J., Mohamed, S., Rezende, D. J., Saxton, D., Cain, A., Hillier, C., Silver, D., Kavukcuoglu, K., Botvinick, M. M., Hassabis, D., and Lillicrap, T. P. Unsupervised predictive memory in a goal-directed agent. *CoRR*, abs/1803.10760, 2018.

Widrich, M., Hofmarcher, M., Patil, V. P., Bitto-Nemling, A., and Hochreiter, S. Modern Hopfield Networks for Return Decomposition for Delayed Rewards. In *Deep RL Workshop NeurIPS 2021*, 2021.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6.

Wu, Y., Fan, Y., Liang, P. P., Azaria, A., Li, Y., and Mitchell, T. M. Read and reap the rewards: Learning to play atari with the help of instruction manuals, 2023.

Yan, C., Carnevale, F., Georgiev, P., Santoro, A., Guy, A., Muldal, A., Hung, C.-C., Abramson, J., Lillicrap, T., and Wayne, G. Intra-agent speech permits zero-shot task acquisition, 2022.

Yao, S., Rao, R., Hausknecht, M. J., and Narasimhan, K. Keep CALM and Explore: Language Models for Action Generation in Text-based Games. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 8736–8754. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.704.

Yee, E., Jones, M., and McRae, K. *Semantic Memory*. 10 2017. doi: 10.1002/9781119170174. epcn309.

Zaidan, O. and Eisner, J. Modeling annotators: A generative approach to learning from annotator rationales. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 31–40. ACL, 2008.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. From recognition to cognition: Visual commonsense reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6720–6731. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00688.

Zeng, A., Wong, A., Welker, S., Choromanski, K., Tombari, F., Purohit, A., Ryoo, M. S., Sindhwani, V., Lee, J., Vanhoucke, V., and Florence, P. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language. *CoRR*, abs/2204.00598, 2022. doi: 10.48550/arXiv.2204.00598. arXiv: 2204.00598.

Zenke, F., Poole, B., and Ganguli, S. Continual Learning Through Synaptic Intelligence. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017.

Zha, D., Ma, W., Yuan, L., Hu, X., and Liu, J. Rank the episodes: A simple approach for exploration in procedurally-generated environments. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Zhong, V., Rocktäschel, T., and Grefenstette, E. RTFM: generalising to new environment dynamics via reading. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Zhong, V., Hanjie, A. W., Wang, S. I., Narasimhan, K., and Zettlemoyer, L. SILG: the multi-domain symbolic interactive language grounding benchmark. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 21505–21519, 2021.

Zhong, V., Mu, J., Zettlemoyer, L., Grefenstette, E., and Rocktäschel, T. Improving policy learning via language dynamics distillation. *CoRR*, abs/2210.00066, 2022. doi: 10.48550/arXiv.2210.00066.

Åström, K. J. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1964.

A   Environments

We choose 8 diverse 3D environments of the MiniWorld benchmark suite:

- **CollectHealth:** The agent spawns in a room filled with acid and must collect medikits in order to survive as long as possible.
- **FourRooms:** The agent must reach a red box that is located in one of four interconnected rooms.
- **MazeS3Fast:** A procedurally generated maze in which the agent needs to find a goal object.
- **PickupObjs:** Several objects are placed in a large room and must be collected by the agent. Since the agent receives a reward of 1 for each collected object, the reward is unbounded.
- **PutNext:** Several boxes of various colors and sizes are placed in a big room. The agent must put a red box next to a yellow one.
- **Sign:** The agent spawns in a U-shaped maze containing various objects of different colors. One side of the maze contains a sign which displays a color in written form. The aim is to collect all objects in the corresponding color.
- **TMaze:** The agent must navigate towards an object that is randomly placed at either end of a T-junction.
- **YMaze:** Same as TMaze, but with a Y-junction.

We neglect the OneRoom and the Hallway environments, since those are easily solved by all compared methods. Further, we neglect the Sidewalk environment since it is essentially the same task as Hallway with a different background. Since the rewards of PickupObjs and CollectHealth are unbounded, we normalize them to be in the range of $[0, 1]$, which is the reward received in all other environments. For a more detailed description of the MiniGrid environments we refer the reader to Paischer et al. (2022a).

B   Token retrieval for synthetic environments

Since we perform retrieval on a token level we investigate the effect of augmenting tokens with different prompts, and the effect of different vision encoders on retrieval performance. We analyse the former at the example of the Avalon environment. Figure 7 shows some examples. We observed that simply encoding single tokens in CLIP space results in a bias toward abstract tokens such as *biome*, or *screenshot*. The same occurs for using the prompts originally used for zero-shot classification on the ImageNet dataset[4] (Deng et al., 2009). However, one can alleviate this bias by including these frequently retrieved tokens in the prompt itself (see Figure 7, bottom). However, we found this strategy to be effective only for Avalon and Psychlab environments. The sets of prompts for both environments can be found in Table 1. For MiniGrid and MiniWorld we retrieve single tokens without any prompt.

Next, we investigate the influence of the choice of vision encoder architecture, e.g., RN50 vs ViT-B/16. We only consider those encoders since they induce the least complexity on our history compression pipeline. We show the closest tokens in CLIP space for observations of MiniWorld (see Figure 9) and Avalon environments (see Figure 8). For MiniWorld, CLIP can extract shapes and colors. However, the retrievals are very noisy, which is mirrored in the attained score of SHELM. For Avalon, however, the token retrievals are more convincing. Generally, we find that retrievals using the RN50 encoder of CLIP tend to be more noisy than retrieval utilizing a ViT-B/16 encoder.

Instead of using CLIP and our token retrieval we may be able to use pretrained image captioning systems, such as BLIP-2 (Li et al., 2023) directly. To investigate this, we use BLIP-2 to generate captions for MiniGrid-Memory, Avalon and Psychlab (Figure 10). The captions generated for the MiniGrid-Memory in the left column contain correctly detected colors (green and red), and shapes (square), but incorrectly combines them, e.g. there is neither a green square, nor a green arrow pointing to the left. A similar trend can be observed for Avalon in the middle comlumn, where BLIP-2 recognizes a frog, but incorrectly assumes that it stands on top of a rock, while the rock is

---

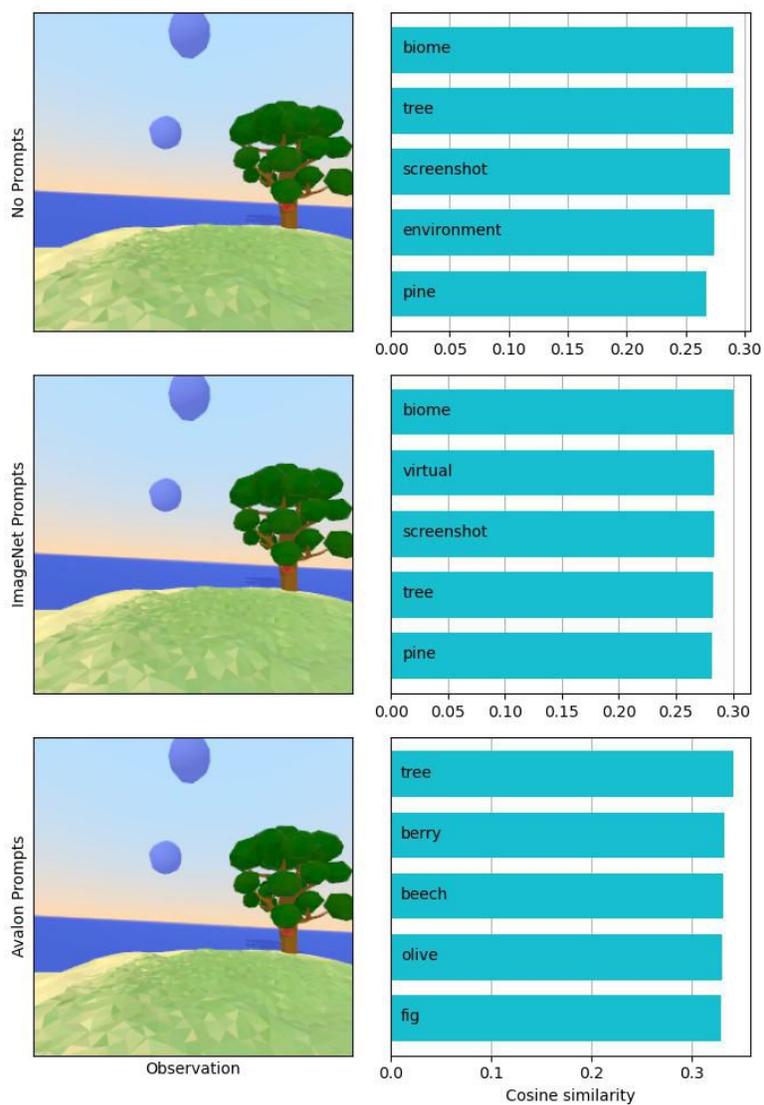[4]available at https://github.com/openai/CLIP/tree/main

Figure 7: Token rankings for ViT-B/16 encoder of CLIP on Avalon observations. Tokens are encoded with the CLIP language encoder without prompt (**top**), with ImageNet specific prompts (**middle**), or prompts designed for Avalon (**bottom**).
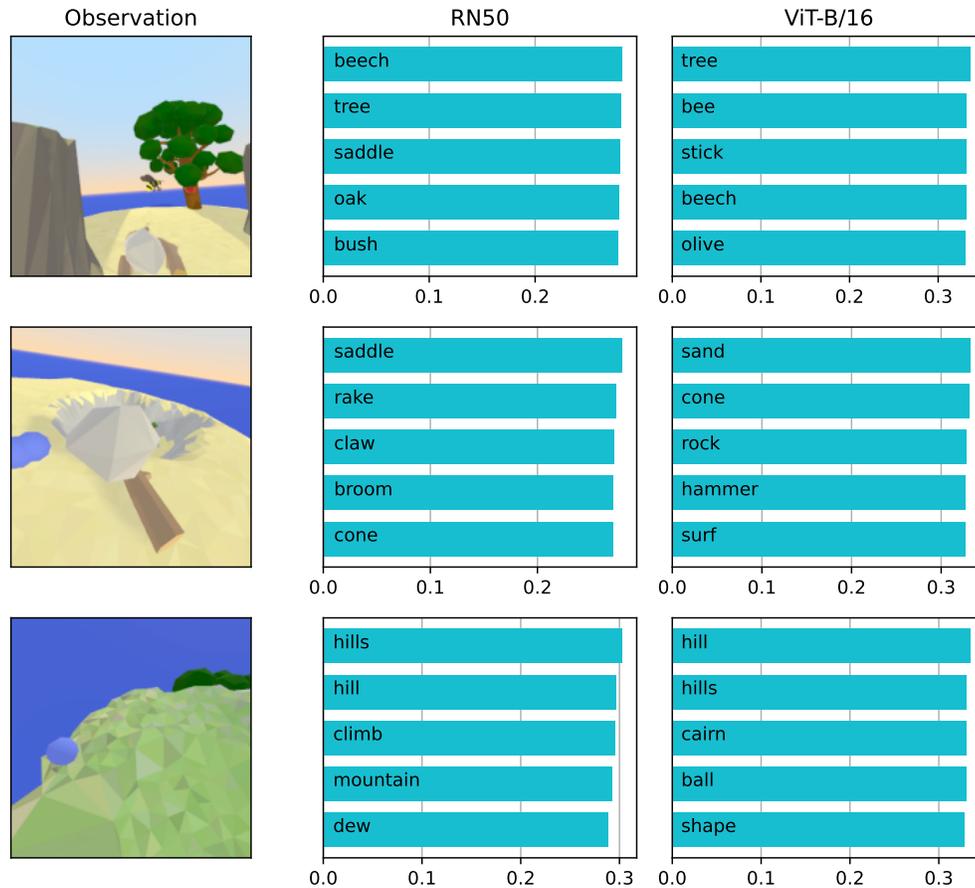
| | Observation | RN50 | ViT-B/16 |

Figure 8: Token rankings for RN50 and ViT-B/16 encoders of CLIP on Avalon observations.

clearly beside the frog in the foreground. The second observation for Avalon is very accurate though. For the two Psychlab observations in the right column, BLIP-2 correctly describes the objects on the screen, but also mentions redundant information, such "the objects are on a screen". Overall, BLIP-2 correctly recognizes shapes, colors, and different objects, but is often incorrect on their compositionality. Further, a drawback of using off-the-shelf captioning engines is that we cannot control for the context length for each observation. This can quickly result in excessive context lengths, which in turn, results in a drastic increase of computational complexity.

C Qualitative analyses

We show token mappings for the memory mechanism of SHELM to identify potential failures and successes. Figure 11 shows a few sample episodes from a trained policy on the Memory environment. Clearly, CLIP is not capable of extracting semantics of the abstract 2D gridworld environments. Thereby, it maps the ball to the token *miner* and the key to the token *narrow*. For minimalistic 3D environments, CLIP is capable of extracting colors and shapes of objects as can be seen in Figure 9. However, these results are still uninspiring since the token retrievals are very noisy.

We also visualize episodes collected by a random policy from the Avalon environment in Figure 12. As opposed to the minimalistic MiniGrid and MiniWorld environments, CLIP successfully extracts semantically meaningful concepts which are subsequently passed to the memory. There are still some failure cases though. For example, in the second episode the agent is attacked by a bee. The feelers of the bee are mistaken for a *sword* as the bee moves into close proximity of the agent. Further after the agent successfully defends itself against the bee, it mistakes the dead bee for a *crab*. Alleviating
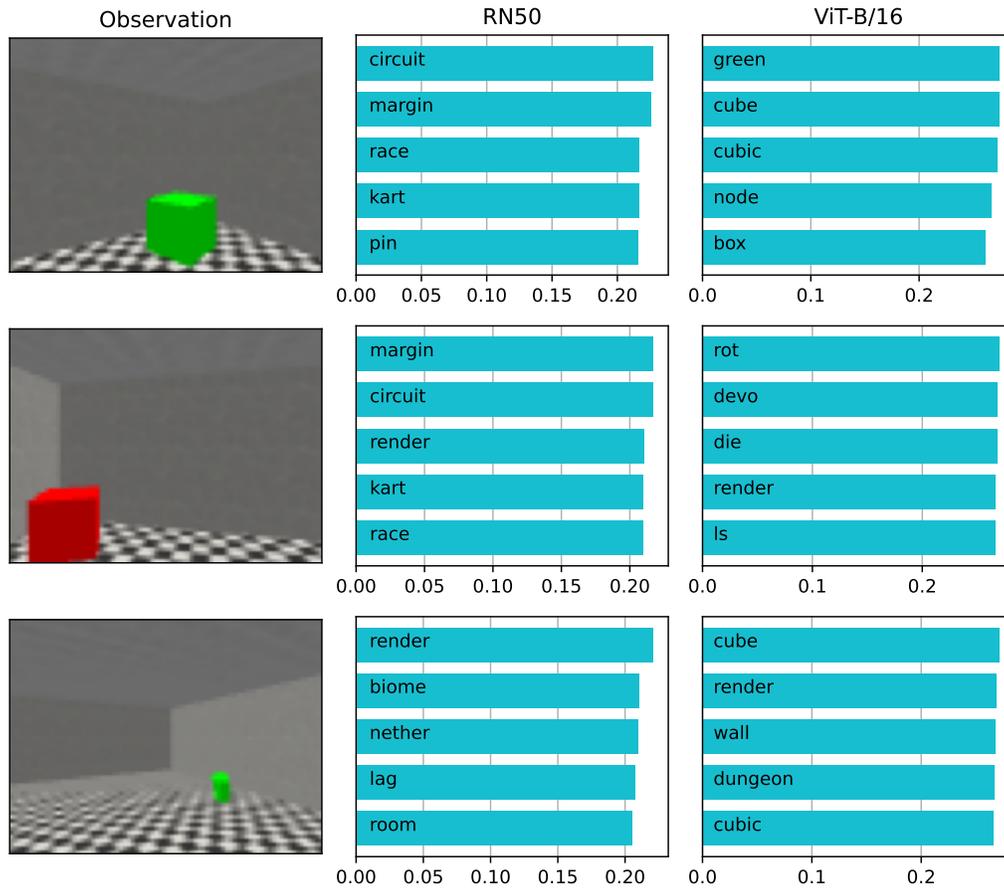
Figure 9: Token rankings for RN50 and ViT-B/16 encoders of CLIP on MiniWorld observations.

Figure 10: Captions generated via BLIP-2 for observations of the MiniGrid-Memory task, Avalon and Psychlab.

Table 1: Prompts used for token retrieval for the Avalon and Psychlab environments.

Environment	Prompts
Avalon	a screenshot of a screenshot of a a screenshot of many a biome containing a biome containing a a biome containing many a biome full of
Psychlab	a render of a render of a a screenshot of a screenshot of a a screen showing a screen showing a

such failure cases would require grounding CLIP to the respective environment as was done in (Fan et al., 2022).

Further, we visualize token mappings for objects the agent encounters in the continuous recognition task of Psychlab in Figure 13. The majority of token retrievals are semantically correct. However, sometimes the agent confuses different objects or conflates to the same token for different objects. An example for that are two middle objects in row 4 which are both mapped to the token *icon*, or the first two objects in row 6 that are mapped to the token *tuber*. We suspect that this occurs due to downscaling to a lower resolution which is conducted within the DMLab engine. Indeed, when taking a closer look at token retrievals at a higher resolution, they are mapped to different tokens (see Figure 14). Therefore, we consider two different aspects on how to alleviate this issue, (i) increasing the resolution of the observations and (ii) retrieving more than one token for an observation. The former results in increased computational complexity, which we aim to avoid, since the task is computationally very expensive already. The latter assumes that the retrieval differs at least in their top-k most likely tokens and is a viable solution.

D Additional results

We show results for PPO, HELM, HELMv2, and SHELM on all Avalon tasks after 10 M interaction steps in Table 2. Dreamerv2 attains the highest score on average after 10 M timesteps, but performs on-par with SHELM. Further, there is no significant difference between the HELM variants though. Since SHELM is demonstrably capable of extracting semantics from Avalon environments (see Figure 12), we believe that the main reason for these results is that the Avalon tasks do not require the agent to use its memory.

To further investigate this finding we run the memory-less PPO baseline for 50 M interaction steps and compare it to results reported in Albrecht et al. (2022) in Table 3. Surprisingly, our PPO baseline significantly outperforms the PPO baseline of Albrecht et al. (2022). Further, we find that our PPO baseline performs on-par with all other memory-based baselines, yielding further evidence that a memory mechanism is not imperative to solve the Avalon tasks.

E Additional ablation studies

We provide an additional ablation study in the choice of vision encoder for SHELM. Our qualitative results in Section 3.1 indicated that vision transformer based encoders are better in extracting semantics of synthetic environments. To corroborate this finding, we run an experiment on the MiniGrid-Memory environment where we exchange the ViT-B/16 with a ResNet-50. The results can be observed in Figure 15. We observe that the ViT-B/16 encoder is much better in discriminating between the different objects which results in a substantial improvement over the RN50 encoder.
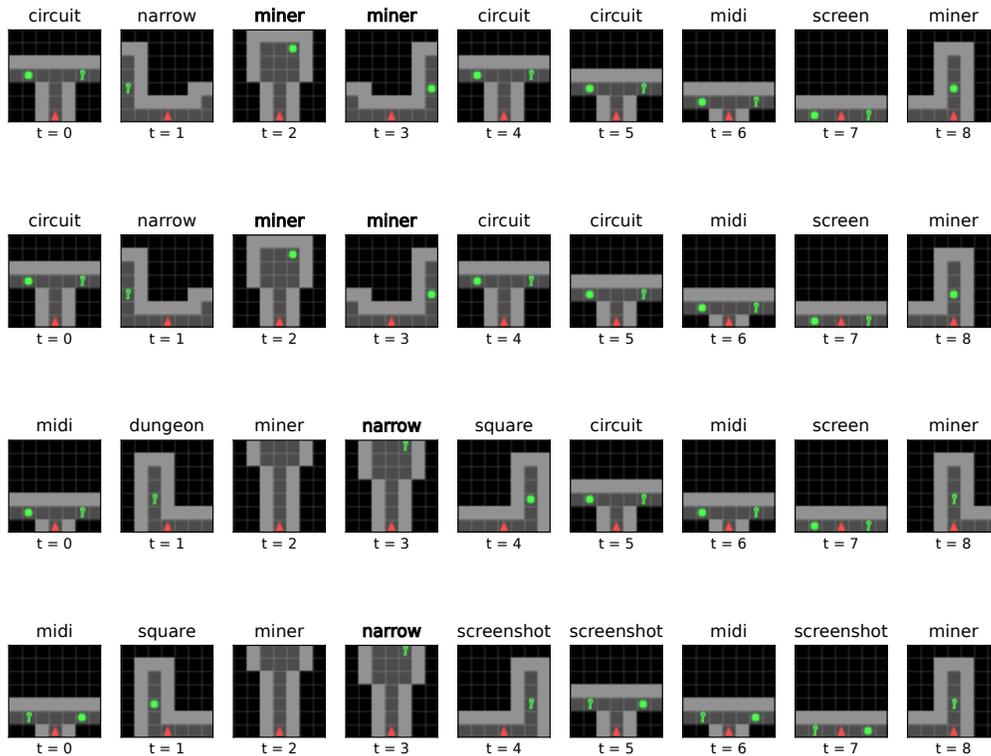
Figure 11: Episodes sampled for a trained SHELM policy on the MiniGrid-MemoryS11-v0 environment. The object *ball* is consistently mapped to the token *miner*, while the object *key* maps to the token *narrow*.

F Hyperparameters and training details

Since the memory component of SHELM does not need not be trained, our memory requirements are comparably low. All our experiments were run on either a single GTX1080Ti or a single A100 GPU. The time requirements of our experiments vary for each environment. For MiniGrid and MiniWorld one run takes approximately two hours, while for Psychlab one run requires two days. These experiments were run on a single GTX1080Ti. For Avalon, we used a single A100 for training where one run to train for 10 M interaction steps takes approximately 15 hours.

MiniGrid and MiniWorld We adapt the hyperparameter search from Paischer et al. (2022a). Particularly, we search for learning rate in $\{5e\text{-}4, 3e\text{-}4, 1e\text{-}5, 5e\text{-}5\}$, entropy coefficient in $\{0.05, 0.01, 0.005, 0.001\}$, rollout length in $\{32, 64, 128\}$ for SHELM. To decrease wall-clock time of HELM variants, we vary the size of the memory register of TrXL such that it can fit the maximum episode length. We lower the number of interaction steps for the gridsearch if we observe convergence before the 500k interaction steps. If no convergence is observed within the 500K interaction steps, we tune for the entire duration. We apply the same scheme for tuning the LSTM baseline and tune the same hyperparameters as in Paischer et al. (2022a).

Avalon After visual inspection of token retrievals for observations we found that there is no substantial difference in retrieved tokens for observations in close proximity to each other. Therefore, we introduce an additional hyperparemeter, namely *history-frameskip*. Note that the history-frameskip is not functionally equivalent to the commonly used frameskip in, e.g., (Mnih et al., 2015). Rather, we actually discard frames within the skip. For example, for a history-frameskip of 10 the agent only
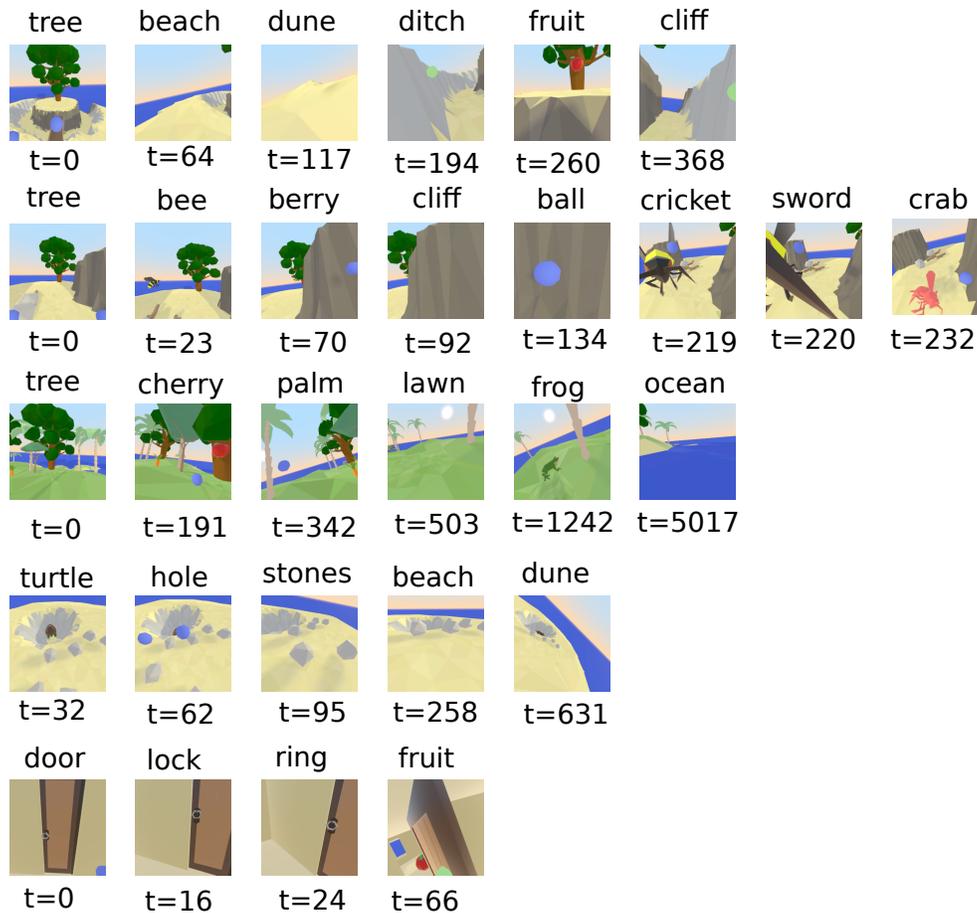
Figure 12: Episodes sampled from a random policy on various Avalon tasks.

observes the first and the eleventh frame in the history branch. The current branch observes every timestep as usual. We search over history-frameskip in $\{3, 5, 10\}$ and adapt the memory of the agent to $\{256, 128, 64\}$ timesteps respectively. Further we search over learning rate in $\{2.5\text{e-}4, 1\text{e-}4, 7\text{e-}5\}$, and the number of retrieved tokens in $\{1, 2, 4\}$. If an observation is represented as more than one token, this effectively reduces the number of observations that fit into the memory register of TrXL, and thereby introduces a recency bias. Hyperparameters for the Dreamerv2 baseline are taken from (Albrecht et al., 2022). For the memory-less baseline we search over learning rates $\{5\text{e-}4, 2.5\text{e-}4, 1\text{e-}4, 5\text{e-}5\}$. We used their respective codebase to run our experiments.[5]

PsychLab Due to the computational complexity of the Psychlab environments we only run a gridsearch over the learning rate in $\{5\text{e-}4, 3\text{e-}4, 1\text{e-}4, 5\text{e-}5\}$. Further we use 64 actors and set the rollout size to 256. For SHELM on continuous-recognition we only retrieve the closest token for an observation.

G   Potential negative societal impact

Our method relies on the use of FMs, which are trained on non-curated datasets which were crawled from the web. Therefore, these models readily reflect the biases and prejudices found on the web and, consequently, so might the resulting RL agent trained by our method. Generally, deployment of RL

---

[5]https://github.com/Avalon-Benchmark/avalon

Figure 13: Sample observations for continuous recognition task of Psychlab. The agent must swipe in a certain direction depending on whether it has encountered an object before in the same episode.
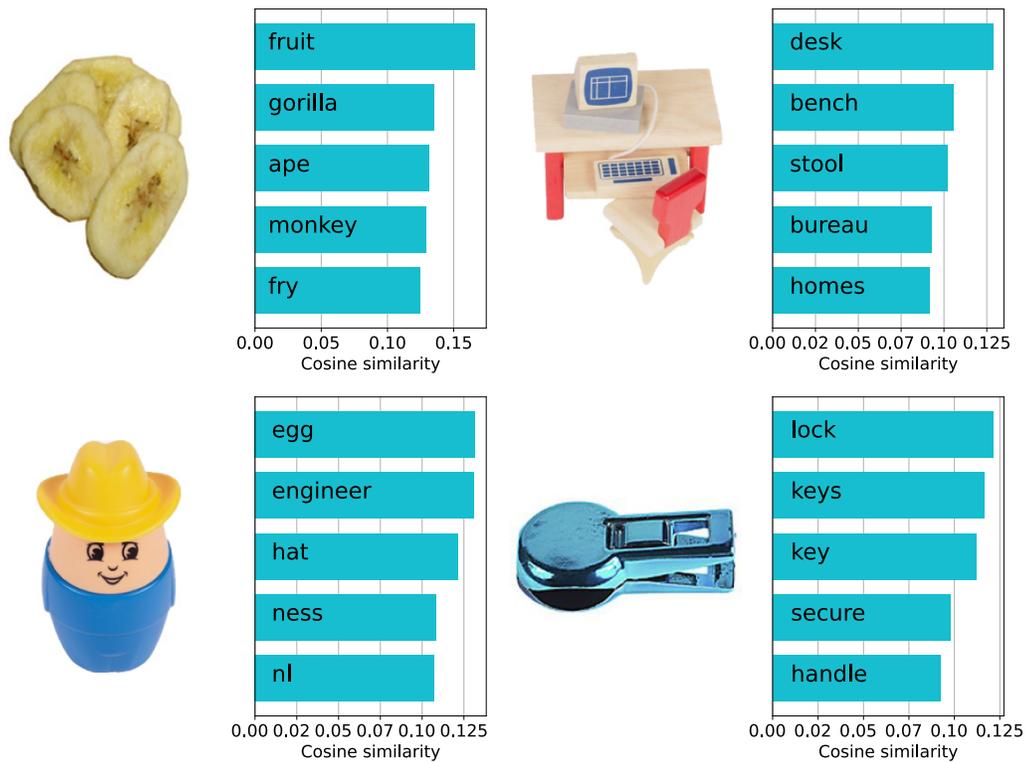
Figure 14: Top-5 token retrievals for objects that are conflated by SHELM at a lower resolution. For higher resolution CLIP successfully maps the objects to different tokens.
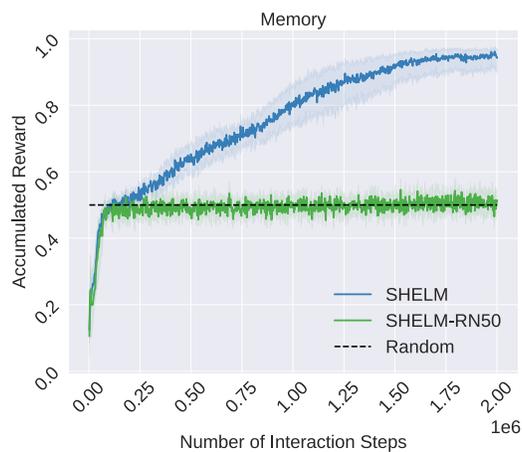

Figure 15: Ablation study on the choice of vision encoder used for the retrieval from our semantic database. We report IQM and 95% CIs across 30 seeds on the MiniGrid-Memory task.

Table 2: Mean human normalized score for for all Avalon tasks for PPO, HELMv2, SHELM, and Dreamerv2 after 10M interaction steps. We show mean and standard deviations.

Task	PPO	HELM	HELMv2	SHELM	Dreamerv2
eat	0.700 ± 0.070	0.671 ± 0.071	0.714 ± 0.063	0.693 ± 0.071	0.656 ± 0.077
move	0.312 ± 0.066	0.277 ± 0.068	0.294 ± 0.061	0.291 ± 0.063	0.362 ± 0.073
jump	0.232 ± 0.053	0.232 ± 0.058	0.219 ± 0.051	0.217 ± 0.050	0.233 ± 0.055
climb	0.120 ± 0.042	0.125 ± 0.039	0.211 ± 0.052	0.118 ± 0.038	0.140 ± 0.048
descend	0.203 ± 0.048	0.184 ± 0.044	0.108 ± 0.035	0.202 ± 0.048	0.283 ± 0.089
scramble	0.241 ± 0.056	0.213 ± 0.052	0.301 ± 0.056	0.271 ± 0.058	0.306 ± 0.089
stack	0.094 ± 0.035	0.058 ± 0.029	0.075 ± 0.031	0.115 ± 0.041	0.106 ± 0.041
bridge	0.020 ± 0.017	0.046 ± 0.026	0.040 ± 0.024	0.028 ± 0.018	0.046 ± 0.031
push	0.109 ± 0.039	0.069 ± 0.030	0.069 ± 0.029	0.110 ± 0.044	0.102 ± 0.044
throw	0.000 ± 0.000	0.017 ± 0.020	0.000 ± 0.000	0.000 ± 0.000	0.007 ± 0.011
hunt	0.059 ± 0.029	0.077 ± 0.033	0.091 ± 0.034	0.091 ± 0.039	0.067 ± 0.034
fight	0.179 ± 0.046	0.189 ± 0.048	0.207 ± 0.062	0.185 ± 0.051	0.299 ± 0.070
avoid	0.234 ± 0.048	0.192 ± 0.047	0.211 ± 0.045	0.227 ± 0.050	0.377 ± 0.062
explore	0.172 ± 0.045	0.194 ± 0.051	0.191 ± 0.050	0.204 ± 0.052	0.196 ± 0.051
open	0.086 ± 0.035	0.134 ± 0.042	0.040 ± 0.024	0.102 ± 0.038	0.055 ± 0.031
carry	0.080 ± 0.031	0.101 ± 0.041	0.073 ± 0.030	0.054 ± 0.027	0.087 ± 0.035
navigate	0.006 ± 0.007	0.000 ± 0.000	0.000 ± 0.003	0.006 ± 0.008	0.006 ± 0.008
find	0.005 ± 0.007	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.003 ± 0.004
survive	0.049 ± 0.015	0.066 ± 0.005	0.041 ± 0.013	0.051 ± 0.016	0.033 ± 0.012
gather	0.007 ± 0.006	0.008 ± 0.019	0.009 ± 0.007	0.006 ± 0.003	0.006 ± 0.004
overall	0.146 ± 0.010	0.143 ± 0.010	0.145 ± 0.010	0.155 ± 0.011	0.168 ± 0.012

Table 3: Mean human normalized score over all Avalon tasks after 50M interaction steps. We show mean and standard deviations. Results for PPO, Dreamerv2, and Impala were taken from Albrecht et al. (2022). Asterisk indicates our own results.

Method	Mean Human Normalized Score
PPO	0.165 ± 0.014
Dreamerv2	0.199 ± 0.012
Impala	0.203 ± 0.013
PPO*	0.200 ± 0.012

agents in the real world requires a carefully designed interface to guarantee safe execution of selected actions. That said, we do not expect any negative societal impact from our memory mechanism. Our semantic memory enhances memory-based methods, thus, can yield more insights into why agents make certain decisions.

H Reproducibility Statement

We make all our code and random seeds used in our experiments, as well as obtained results publicly available at https://github.com/ml-jku/helm. The pretrained language encoder, as well as the pretrained CLIP encoder are publicly available on the huggingface hub (Wolf et al., 2020).