Toward Human Cognition-inspired High-Level Decision Making For Hierarchical Reinforcement Learning Agents

Rousslan Fernand Julien Dossa [1] **Takashi Matsubara** [2]

Abstract

The ability of humans to efficiently understand and learn to solve complex tasks with relatively limited data is attributed to our hierarchically organized decision-making process. Meanwhile, sample efficiency is a long-standing challenge for reinforcement learning (RL) agents, especially in long-horizon, sequential decision-making tasks with sparse and delayed rewards. Hierarchical reinforcement learning (HRL) augments RL agents with temporal abstraction to improve their efficiency in such complex tasks. However, the decision-making process of most HRL methods is often based directly on dense low-level information, while also using fixed temporal abstraction. We propose the hierarchical world model (HWM), which is geared toward capturing more flexible high-level, temporally abstract dynamics, as well as low-level dynamics of the task. Preliminary experiments on using the HWM with model-based RL resulted in improved sample efficiency and final performance. An investigation of the state representations learned by the HWM also shows their alignment with human intuition and understanding. Finally, we provide a theoretical foundation for integrating the proposed HWM with the HRL framework, thus building toward RL agents with hierarchically structured decision-making which aligns with the theorized principles of human cognition and decision process.

1. Introduction

Deep reinforcement learning (DRL) has proven to be a powerful set of automation methods, able to solve a gamut of tasks varying in complexity (Mnih et al., 2013; Schulman et al., 2017; Haarnoja et al., 2018b; Dabney et al., 2018; Hafner et al., 2020a). Still, conventional DRL methods can be very sample inefficient when applied to long-horizon, sequential decision-making tasks, which usually overlap with sparse and delayed rewards problems, further increasing their complexity.

A growing body of studies in human behavior, cognitive science, neuroscience, and computational biology suggests that human behavior is hierarchically organized (Botvinick et al., 2011). This is theorized to be a critical part that allows us to efficiently understand and learn to solve various challenging tasks with relatively limited amounts of data compared to RL methods (Tomov et al., 2020; Xia & Collins, 2020).

Inspired by this, the hierarchical reinforcement learning (HRL) framework aims to improve the efficiency of conventional (flat) RL by introducing temporal abstraction in the decision-making process of an agent (Dayan & Hinton, 1992; Sutton et al., 1999; Barto & Mahadevan, 2003; Botvinick et al., 2011). While existing HRL methods have empirically demonstrated a considerable improvement in efficiency over conventional DRL methods (Kulkarni et al., 2016; Vezhnevets et al., 2017; Florensa et al., 2017; Haarnoja et al., 2018a; Li et al., 2019), most HRL methods rely on *fixed length temporal abstraction*. Moreover, the decision-making occurring at higher levels in the agent's hierarchy is still often based on the *observations at the lowest level*.

In this work, we proposed the *hierarchical world model* (HWM), a world modeling method that aims to better fit the theorized hierarchical structure of the human decision-making process. Owing to its hierarchical structure, the proposed model inherently provides (1) a *temporally abstract state representation* summarizing an arbitrary number of lower-level states, and (2) an adaptive temporal abstraction mechanism to divide long-horizon, sequential decision tasks into smaller tasks of variable lengths.

Preliminary experiments in using the HWM as a representation learning mechanism for a Dreamer-based agent (Hafner et al., 2020a;b) demonstrated improvements in sample effi-

[1]Graduate School of System Informatics, Kobe University, Hyogo, Japan [2]Graduate School of Engineering Science, Osaka University, Osaka, Japan. Correspondence to: Rousslan F. J. Dossa <doss@ai.cs.kobe-u.ac.jp>.

Proceedings of the 39th International Conference on Machine Learning at the Decision Awareness in Reinforcement Learning Workshop, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

ciency and final performance of the agent. Further analysis of the representations learned by the proposed HWM also suggests that the HWM can learn a hierarchically structured internal state representation that aligns with human understanding and intuition (Botvinick & Weinstein, 2014; Tomov et al., 2020) on a given task.

Finally, we show the potential compatibility of the HWM with the HRL framework by expanding the underlying theory of the proposed HWM to incorporate more rigorous hierarchical decision-making. This would bring the practical HRL framework closer to the human decision-making it is originally inspired from, and serve as a tool for analyzing RL agents' decision-making. In practice, this could help broaden the gamut of tasks to which deep RL methods can be successfully applied for automation.

2. Preliminaries

2.1. Reinforcement learning and hierarchical reinforcement learning

We base ourselves on the formalism of a finite time horizon partially observable Markov decision processes (POMDP), which we denote by the tuple $(\mathbb{S}, \mathbb{A}, P_{s,a}, R, \mathbb{O}, P_o, H)$, where $\mathbb{S}$ is the state set, $\mathbb{A}$ is the action set, $P_{s,a} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \rightarrow [0, 1]$ is the state transition probability, $R : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$ the reward function, $\mathbb{O}$ the set of partial observations, $P_o : \mathbb{S} \rightarrow \mathbb{O}$ the emission probability distribution, and H the horizon (maximum episode length). The decision process of an RL agent can be represented by a stochastic policy $\pi : \mathbb{S} \times \mathbb{A}$, which defines the probability of action a being selected given a state s. From a probabilistic perspective (Levine, 2018; Sun & Bischl, 2019), the POMDP modeling the dynamics of the system and the decision making of the agent can be formally expressed as the following generative process:

$$p(O, S, A) = \prod_{t=1}^{H} p(o_t|s_t)\, p(s_t|s_{t-1}, a_{t-1})\, \pi(a_t|o_t) \quad (1)$$

In the hierarchical RL (HRL) framework, the policy of the agent decomposed into hierarchically structured component policies. The lowest level in the hierarchy operates at the finest time scale, the same as a flat RL policy. On the other hand, the higher levels in the hierarchy operate at a coarser time scale, thus resulting in temporal abstraction. This would allow HRL agents to explore the state space more efficiently by leveraging sequential combinations of low-level policies (also referred to as *options* or *skills*), allowing for a more efficient sequential decision making (Sutton et al., 1999; Barto & Mahadevan, 2003; Botvinick et al., 2011) in long-horizon tasks. Concurrently, temporal abstraction allows for a better credit assignment through time, which is especially helpful in long-horizon, sparse and delayed reward tasks (Sutton et al., 1999; Vezhnevets et al., 2017).

Consider the simplest case of a two-level hierarchical agent composed of π^{H} and π^{L} respectively representing the high and low-level policies. π^{H} selects a *skill* on which π^{L} will be conditioned. In practice, π^{H} is usually conditioned on the low-level observations, and its time scale is often set to a fixed length we denote as k hereafter (Vezhnevets et al., 2017; Kulkarni et al., 2016; Florensa et al., 2017; Haarnoja et al., 2018a). Denoting the *skill* space by $\mathbb{E}$, and augmenting the generative process of the POMDP in Eq. 1, we obtain:

$$p(O, S, A, E) = \prod_{\tau=0}^{H/k} \pi^{\mathrm{H}}(e_\tau|o_{k\tau}) \prod_{t=k\tau}^{k(\tau+1)} p(o_t|s_t) \; p(s_t|s_{t-1}, a_{t-1})\, \pi^{\mathrm{L}}(s_t|o_t, e_\tau). \quad (2)$$

The graphical models for Eq. 1 and Eq. 2 are illustrated in Fig. 1 (a) and (b), respectively.

2.2. Hierarchically organized behavior

A growing body of studies (Botvinick et al., 2011; Botvinick & Weinstein, 2014; Tomov et al., 2020; Xia & Collins, 2020) at the intersection of neuroscience, cognitive science, psychology, and computational biology suggests that the human decision-making process is hierarchically organized. For example, when faced with a task such as *making a trip abroad*, we divide it into a sequence of sub-tasks such as *booking the flight*, *packing the luggage*, *driving to the airport*, *boarding the plane*, and so on. Each sub-task can be further divided into sub-sub-tasks, down to the finest granularity of actions such as bodily movements. This concept is referred to as *temporal abstraction* and allows us to efficiently learn, plan, and act in a wide gamut of activities. It is a fundamental principle underlying the HRL framework introduced in 2.1, where each of the aforementioned sub-tasks would be realized by learning and executing the appropriate *skill*.

While existing HRL methods (Sutton et al., 1999; Vezhnevets et al., 2017; Kulkarni et al., 2016; Florensa et al., 2017; Haarnoja et al., 2018a; Li et al., 2019) do structure the decision-making process of the agent hierarchically, the *skill* selection at higher levels in the hierarchy is more often than not based on the observations at the finest level of the hierarchy. In the case of example task *making a trip abroad*, this is akin to having the high-level policy decide to *drive to the airport* while using a very exhaustive representation of the current state of the agent, instead of the more abstract state *luggage packed and ready to go to the airport*.

Following this line of thought, (Tomov et al., 2020; Xia & Collins, 2020) suggest that our hierarchical decision-making process is intertwined with a corresponding *temporally abstract state representation*. Intuitively, such abstracted state representation would align with intermediate sub-goals, milestones, or *bottleneck states* that contribute to

solving the overall task, thereby greatly simplifying planning, exploration, and execution.

The ability to create long-term plans and strategies without necessarily interacting with the world happens to also be tied to such abstract state representation. This is made possible by leveraging our internal model of the world, which is theorized to also be hierarchical. For example, we can both imagine the detailed process of *folding shirt* (low-level), as well as the more abstract process of *putting folded shirts into the suitcase* (high-level). Botvinick et al. empirically demonstrated the benefit of having a *temporally abstract model* (Botvinick & Weinstein, 2014). They proposed a model-based HRL variant of the *options framework* (Sutton et al., 1999), where the standard HRL agent is augmented with a temporally abstract model that allows the agent to directly plan at a coarser time scale. This resulted in a great improvement in performance, but also sample efficiency. However, the *skills*, and the temporally abstract model were manually engineered, thus limiting the generality of the proposed approach.

As suggested by (Barto & Mahadevan, 2003), it would be desirable to endow HRL agents with *an additional mechanism that allows autonomous extraction of temporally abstract state, and the corresponding temporal dynamics model*. Moving toward a human cognition-inspired decision-making process, we turn ourselves to recent methods for dynamics learning that fall under the umbrella of *world modeling*, to complement standard HRL agents with a general, end-to-end mechanism for discovery and learning of temporally abstract state representations and dynamics.

2.3. World models

The broad range of methods that have marked the recent resurgence of model-based RL (MbRL) are referred to as *world models* (Ha & Schmidhuber, 2018; Hafner et al., 2018; 2020a;b). Such methods have not only demonstrated either competitive or superior performance to the leading model-free RL methods but also improved the sample efficiency of the agents.

One of the key factors behind the success of world modeling methods is the introduction of self-supervised learning objectives (Kingma & Welling, 2014) to learn more compact and meaningful representations of the internal state belief s_t in Fig. 1 (a). This allowed separating the decision-making component (policy) from the raw and usually noisy pixel-based observations, greatly simplifying the learning of the decision-making process. Additionally, world modelling techniques leverage Recurrent Neural Networks (RNN) (Hochreiter & Schmidhuber, 1997; Cho et al., 2014; Goodfellow et al., 2016), to approximate the state transition dynamics $p(s_t|s_{t-1}, a_{t-1})$ from Eq. 1. Such approximation can then be used to collect samples in an *imaginary* environment in the place of the real environment, leading to a drastic improvement in sample efficiency.

Through the series of *Dreamer* agents (Hafner et al., 2020a;b), Hafner et al. take this concept one step further by seamlessly incorporating the prediction of the reward and episode termination with an actor-critic (Schulman et al., 2017; Haarnoja et al., 2018b) policy into the model. Leveraging the differentiable dynamics of the latter allows *Dreamer* agents to directly improve their policy in an end-to-end manner over simulated trajectories.

To further emphasize the gap between the desideratum of this study and the existing world modeling methods, let us consider how a world model enables RL agents to also create plans (sequences of actions) without rolling out the physical or simulated environment. This feature of models is indeed a close, albeit a drastically simplified version of our ability as humans to mentally simulate various scenarios and their development for a given task. Namely, we maintain a set of intricately organized internal beliefs about the surrounding environments and even the actors that populate them. A practical example would be a game of chess, where a professional player is planning multiple moves, anticipating various strategies of his opponent, and developing counter moves. For an example more grounded in daily life activities, let us again consider the task of planning a trip. For us, such planning is a process that ranges over low-level details such as "what documents to prepare", "which clothes to take ?", "what time to wake up at ?", to more and more abstract level such as "taking the train from city A to city B both in country C" then "taking the plane from city B to city D in country E", and so on (Tomov et al., 2020).

As humans, our decision-making process is indeed not limited to low-level and densely detailed state beliefs (representations) of the world. Most, if not all world model methods, however, only estimate the internal state belief and transition dynamics at the finest time scale. Inspired by the theorized hierarchically structured model of animals and humans presented in 2.2, we seek to augment conventional modeling methods with a hierarchical structured dynamics model.

2.4. Variational temporal abstraction

The variational temporal abstraction (VTA) (Kim et al., 2019) framework was introduced as a discovery method for temporally hierarchical structure and representation in sequential data. Formally, VTA assumes the existence of a sequence of observations $O = \{o_1, o_2, ...o_H\}$ of length H that can be decomposed into N non-overlapping sub-sequences $O = (O_1, O_2, ..., O_N)$, such that each sub-sequence $O_i = \{o^i_{1:l_i}\}$ has length l_i, and $\sum_{i=1}^{N} l_i = H$. Each observation o_t is generated from the corresponding *low-level state* w_t, such that each observation sub-sequence

O_i is associated with a low-level state sub-sequence W_i. Finally, each low-level state sub-sequence W_i is assumed to have been generated from a *temporally abstract state* z_i.

To efficiently separate sub-sequences W_i corresponding to different z_i, the VTA framework leverages a binary random variable M referred to as *binary boundary indicator* instead of modeling both the number of sub-sequences N and their lengths L. At an arbitrary time-step t, the binary indicator m_t specifies whether or not a new sub-sequence starts at time step $t+1$. The generative process for an observed sequence O is formally defined as follows:

$$p(O, W, Z, M) = \prod_{t=1}^{H} p(o_t|w_t)\, p(w_t|z_t, w_{t-1}, m_{t-1}) \\ p(z_t|z_{t-1}, m_{t-1})\, p(m_t|w_t). \tag{3}$$

The *temporally abstract* state z and the low-level state abstraction w are approximated using the proposed hierarchical recurrent state-space model, which is trained using sequential variational inference (Kingma & Welling, 2014).

While it does provide an adaptable mechanism to learn temporally abstracted dynamics and the corresponding abstract states, it cannot be directly incorporated into a hierarchically structured decision-making process. Namely, it does not model the influence of the actions of neither low nor high-level actions originating as causal components of the observed sequences.

3. Proposed method

3.1. Theoretical model

In this work, we propose the *hierarchical world model* (HWM), which combines conventional world modeling techniques and VTA to capture flexible, temporally abstract dynamics, and the corresponding state representations structured hierarchically.

First, we adopt the world model structure of the Dreamer agents (Hafner et al., 2020a;b) introduced in Section 2.3. For a given task formalized as a POMDP, such a world model approximates the internal belief state s_t as a single random variable using sequential variation inference. Following the VTA (Kim et al., 2019) framework, we assume that the internal belief state s_t itself can be decomposed into the hierarchically structured components w_t, and z_t, with the boundary indicator m_t determining when the temporally abstract transition takes place.

Since we are mainly interested in learning the dynamics of the environment, we consider the sequence of observations as generated by a fixed policy π, which influence is hereafter represented by the random variable A. Based on Eq. 1, and Eq. 3 we obtain the following generative process the proposed model is based upon:

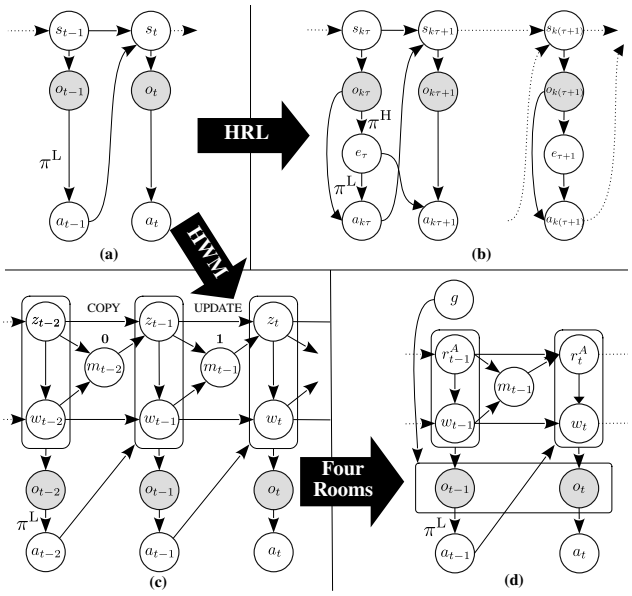


Figure 1. (a) Graphical model of a POMDP task dynamics and decision-making process of a flat RL agent. (b) Graphical model of the temporal abstraction introduced to the task following the HRL framework. (c) Dynamic Bayesian network of the POMDP task with a hierarchically structured state representation following the proposed HWM. (d) Dynamic Bayesian network of the proposed HWM applied to the *Four Rooms* task. The stochastic variable g denotes the goal of the agent, and r^A denotes the room of the agent. The variables g and r^A would correspond to the z variable in the proposed HWM formulation. Finally, w denotes more detailed information about the agent relatively to the room it is located in.

$$p(O, W, Z, M|A) = \prod_{t=1}^{H} p(o_t|w_t, z_t)\, p(m_t|w_t, z_t, a_t) \\ p(z_t|z_{t-1}, m_{t-1}, w_{t-1}, a_{t-1})\, p(w_t|z_t, w_{t-1}, a_{t-1}). \tag{4}$$

The corresponding graphical model is documented as Fig. 1 (c).

3.2. Learning and inference of task dynamics

The generative process proposed in Eq. 4 is modeled using a hierarchical recurrent state-space model (Kim et al., 2019; Saxena et al., 2021). More specifically, $p_\theta(o_t|w_t, z_t)$ is parameterized as the decoder component of a variational auto-encoder (Kingma & Welling, 2014). The prior over the temporally abstract transition is modeled by $p_\theta(z_t|z_{t-1}, m_{t-1}, w_{t-1}, a_{t-1})$ approximated using deep neural networks.

To improve the modeling of long-term dynamics, z_t is decomposed into a deterministic component c_t and and

stochastic component v_t (Hafner et al., 2018; 2020a;b; Kim et al., 2019). The deterministic transitions for c_t are modeled using the following rule:

$$c_t = \begin{cases} c_{t-1} & \text{if } m_{t-1} = 0 \text{ (COPY)} \\ f_{\text{z-rnn}}(z_{t-1}, h_{t-1}, c_{t-1}) & \text{otherwise (UPDATE)} \end{cases}$$

where $f_{\text{z-rnn}}$ is a GRU neural network (Cho et al., 2014; Kim et al., 2019), and h_{t-1} is represents the preceding sequence of (w, a) pairs. The stochastic component v_t is implemented as a Normal distribution: $v_t \sim \mathcal{N}(\mu_v(c_t), \sigma_v(c_t))$, where μ_v and σ_v are parameterized by their respective densely connected feed-forward neural networks. The temporally abstract state z_t is thus obtained by concatenating c_t and v_t.

Similarly, the low-level state w_t is also decomposed into deterministic and a stochastic components, respectively denoted by h_t and y_t. Unlike in VTA (Kim et al., 2019), h_t is seamlessly updated like in Clockwork VAEs (Saxena et al., 2021) using the rule:

$$h_t = f_{\text{w-rnn}}\big(w_{t-1}, a_{t-1}, h_{t-1}\big),$$

where $f_{\text{w-rnn}}$ is the GRU neural network associated with the low-level state transitions. The stochastic component y_t is implemented using a normal distribution $y_t \sim \mathcal{N}(\mu_y(h_t), \sigma_y(h_t))$, where μ_y and σ_y are parameterized by their respective densely connected feed-forward neural networks. The concatenation of h_t and y_t is then used to represent the low-level state.

Finally, the prior boundary detector $p_\theta(m_t|w_t, z_t, a_t)$ is parameterized using a densely connected neural network with a final sigmoid activation function.

The hierarchy of state representation components and the corresponding dynamics is inferred (Kingma & Welling, 2014) using the parameterized variational distribution $q_\phi(Z, W, M|O, A)$. The latter is decomposed as follows:

$$q_\phi(Z, W, M|O) = q_\phi(M|O) \prod_{t=1}^{H} q_\phi(w_t|z_t, M, O)\, q_\phi(z_t|M, O),$$

with $q_\phi(M|O) = \prod_t \text{Bernoulli}(m_t|\sigma(\varphi(O)))$, where σ is the sigmoid function, and φ is a temporal convolution operation over a sequence of observations. Both $q_\phi(w_t|z_t, M, O)$ and $q_\phi(z_t|M, O)$ are approximated using the mean field approximation-based method (Kim et al., 2019; Saxena et al., 2021).

The parameter vectors θ and ϕ are learned by maximizing the variational lower bound (VLB) derived from Eq. 4 as follows:

$$\log p(O|A) \geq \mathop{\mathbb{E}}_{q_\phi}\Big[\log p_\theta(O|Z, W)\Big] - \text{KL}\Big[q_\phi(Z, W, M|O, A) \,||\, p_\theta(Z, W, M|A)\Big]. \tag{5}$$

For a given sequence of observation-action pairs $\{(o_t, a_t)\}_{t=1}^{H}$, the VLB derived in Eq. 5 is approximated as:

$$J_{\text{HWM}}(\theta, \phi) = \sum_{t=1}^{T} \log p_\theta(o_t|w_t, z_t) - \text{KL}\big[q_\phi(z_t) \,||\, p_\theta(z_t)\big] - \text{KL}\big[q_\phi(w_t) \,||\, p_\theta(w_t)\big] - \text{KL}\big[q_\phi(m_t) \,||\, p_\theta(m_t)\big]. \tag{6}$$

The first term in Eq. 6 corresponds to the reconstruction objective across the observed sequence. The last three terms correspond to the Kullback-Leibler divergence between the generative and variational distributions used to approximate temporally abstract state z_t dynamics, the low-level state w_t's dynamics, and the sequence segmentation based on the boundary indicator m_t, respectively.

3.3. Incorporation with an actor-critic

To enable decision-making based on the proposed HWM, the model is further extended with a reward predictor $p_\theta(\hat{r}_t|w_t, z_t)$. This predictor is approximated as a Normal distribution $\hat{r}_t \sim \mathcal{N}(\mu_{\hat{r}}(w_t, z_t), 1)$ where the mean $\mu_{\hat{r}}(w_t, z_t)$ is parameterized using a feed-forward neural network. The reward predictor is then trained to maximize the likelihood of the reward r_t collected during trajectory sampling. Formally, this corresponds to minimizing the objective function $J_{\text{R}}(\theta, \phi) = \sum_{t=1}^{H} -\log p_\theta(r_t|w_t, z_t)$ jointly with $J_{\text{HWM}}(\theta, \phi)$ defined in Eq. 6.

Following (Hafner et al., 2020a;b), the actor-critic is composed of an action and a value model. In the scope of this paper, we defined the action model as the policy $\pi_\psi(a_h|w_h, z_h)$ to be a distribution over discrete actions. The value model is defined as a state value function that estimates the average reward-to-go from a given state (w_h, z_h), formally expressed as follows:

$$V_\psi^t = V_\psi(w_t, z_t) \approx \mathbb{E}_{q_\phi(\cdot|w_t, z_t)}\Big[\sum_{h=t}^{t+H} \gamma^{h-t} \hat{r}_h\Big].$$

Here, ψ denotes the vector of parameters that form the layers of the deep neural networks expressing both π_ψ and V_ψ.

The training objective of the value function is to regress the V_ψ^t estimate to the state value target denoted as V_λ^t. The latter is estimated using the λ-return estimation (Hafner et al., 2020a;b). The objective function for the value learning

components is thus formally expressed as:

$$J_{\mathrm{V}}(\psi) = \mathbb{E}_{p_\theta(w_t, z_t)}\Big[\sum_{h=t}^{t+H} \frac{1}{2}\big|\big|V_\psi(w_h, z_h) - V_\lambda^h\big|\big|^2\Big] \quad (7)$$

The role of the actor component is to select action that maximize leads to states with maximal value estimation. Exploration is implicitly incorporated by regularizing the policy entropy following (Haarnoja et al., 2018b; Hafner et al., 2020b). This is formally expressed as follows:

$$J_\pi(\psi) = \mathbb{E}_{p_\theta(w_t, z_t)}\Big[\sum_{h=t}^{t+H} -\log \pi_\psi(a_t|w_h, z_h)\, \mathrm{sg}(V_\psi^h) - \eta\, \mathcal{H}\big[a_h|w_h, z_h\big]\Big] \quad (8)$$

with *sg* denoting the *gradient stopping operation*, $\mathcal{H}$ the entropy of the policy π_ψ, and η the entropy regularization coefficient.

Finally, the actor-critic component objective function $J_{\mathrm{AC}}(\psi)$ is then constructed as follows:

$$J_{\mathrm{AC}}(\psi) = J_\pi(\phi) + J_{\mathrm{V}}(\psi). \quad (9)$$

The overall cost function of the proposed HWM as a representation learning component, and the actor-critic component to be minimized is thus expressed as follows:

$$J(\theta, \phi, \psi) = J_{\mathrm{HWM}}(\theta, \phi) + J_{\mathrm{R}}(\theta, \phi) + J_{\mathrm{AC}}(\psi). \quad (10)$$

While in practice the actor-critic component is jointly trained with the proposed HWM, let us emphasize that $J_{\mathrm{AC}}(\psi)$ is minimized over *simulated trajectories* of length H, instead of directing relying on data sampled during rollouts of the agent (Schulman et al., 2017; Haarnoja et al., 2018a; Dabney et al., 2018). Those simulated trajectories are produced by the approximated dynamics transitions $p_\theta(w_t)$, $p_\theta(z_t)$, and $p_\theta(m_t)$ of the proposed HWM that were described in Section 3.1 and 3.2.

4. Experimental setting

The experiments are grounded in one of the representative task of the HRL setting referred to as *Four Rooms* (Sutton et al., 1999), illustrated in Fig. 4 (a). We built on top of the publicly available implementation referred to as the *MiniGrid-FourRooms-v0* environment, provided by (Chevalier-Boisvert et al., 2018). For the purpose of our experiments, the environment is modified to provide RGB images of a bird's eye perspective of the maze as observations to the RL agent.

In *Four Rooms*, the agent (red triangle) illustrated in Fig. 4) (a) has to reach the exit (green tile) within a maximal episode length of $H = 100$ steps. The layout of the maze, which determines the position of the *doors* is fixed across all episodes. Both the starting position of the agent and the goal are set randomly at the beginning of each episode. The action space is simplified to three actions: *turn left*, *turn right*, and *move forward*.

For our purpose, we consider a *decomposed state representation* of s_t, illustrated in Fig. 1 (d). Namely, we can efficiently describe the state of the whole maze using 3 random variables. First, let G encode the room of the goal, as well as the relative x and y coordinates of the goal in said room. Next, the agent position in the maze can be decomposed into the room of the agent, denoted as the random variable R^A, and the variable W that encodes the relative x and y coordinates of the agent in R^A, as well as the direction it is facing. This is derived from the observation that the information encoded by G is fixed across an episode, while the room of the agent R^A changes less frequently when compared to the x and y coordinates, or the direction of the agent encoded in W. Notice that this compact representation can be matched with the two-level hierarchy of the proposed HWM as illustrated in Fig. 1 (c), by collapsing both G and R^A into the variable Z, because they both change at a slower pace than W. This is motivated by the experimental results of (Saxena et al., 2021), stipulating that in a hierarchically structured recurrent state-space model, slowly changing information in the observations is encoded into higher levels of the latent variable hierarchy. Meanwhile, fast-changing components are encoded at the lower levels.

We design the first experimental phase to demonstrate how the proposed HWM captures an adaptable temporally abstracted state dynamics. To this end, we train an RL agent instantiated as Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) using the reference implementation provided in the *Stable Baselines 3* RL algorithm library (Raffin et al., 2021). The pre-trained PPO agent is used to generate a dataset of $25,000$ observation-action pairs (o_t, a_t), corresponding to $1,736$ distinct episode trajectories. This dataset is then used to train an instance of the proposed HWM without an actor-critic component, to maximize the objective function $J_{\mathrm{HWM}(\theta,\phi)}$ derived in Eq. 6. The objective of this experimental phase is solely to analyze the learned hierarchical state representation. It is thus separate from the decision-making aspect, which is focused upon in the next experimental phase.

The second experimental phase aims to investigate whether or not the proposed HWM is a valid world modeling method, and if it provides any performance or sample efficiency improvement over classical world modeling methods. To this end, we jointly train the proposed HWM and the actor-critic component to minimize the overall objective function $J(\theta, \phi, \psi)$ described Eq. 10 on the same

MiniGrid-FourRooms-v0 used in the first experimental phase. The same procedure is also conducted on the *Pong* task of the Atari game suite (Bellemare et al., 2013) to verify against the baseline Dreamer agent (Hafner et al., 2020a;b). Due to computational limitations, the agents are trained for $2,000,000$ environment steps, corresponding to $400,000$ model and actor-critic update steps for *MiniGrid-FourRoooms-v0*. For *Atari's Pong*, the agents are trained for $20,000,000$, corresponding to $1,250,000$ world model and actor-critic update steps.

5. Results

5.1. Hierarchically structured state representation

Recall that the purpose of the HWM is to provide (1) a *temporally abstract state representation* summarizing an arbitrary number of lower-level states while at the same time filling the role of a world model (Ha & Schmidhuber, 2018; Hafner et al., 2020a;b). Additionally, the model should also provide (2) an adaptive temporal abstraction mechanism to divide trajectories into coherent sub-sequences (Kim et al., 2019).

To evaluate the proposed method, a trained instance of the proposed HWM following the setting described in Section 4 is fed trajectories of observation-action pairs (o_t, a_t). For each trajectory, the first temporally abstract state z_0 is inferred using the learned posterior $q_\phi(z_t)$. For $t > 0$, the temporal transition is modeled using the learned $p_\theta(z_t)$. The lower-level state w_t at each step is inferred using the posterior $q_\phi(w_t)$, similarly to Dreamer agents (Hafner et al., 2020a;b). The boundary indicator m_t is estimated using the learned prior distribution $p_\theta(m_t)$. Finally, each observation o_t is reconstructed using the learned decoder $p_\theta(o_t|w_t, z_t)$.

From lines (a) and (e) in Fig. 2, we observe that the HWM manages to accurately reconstructs the provided observations, while modeling the high and low-level state transitions, thus satisfying the requirement (1). Requirement (2) is also satisfied, as the prior boundary indicator $p_\theta(m_t)$ can accurately predict the change in the room of the agent. This is indicated in Fig. 2 by a change of color at the next step every time the prior boundary indicator detects a new segment. In this specific case, the proposed trajectory segmentation also coincides with the intuited abstracted dynamics we derived for the *Four Rooms* task, as illustrated in Fig. 1 (d).

Through the lines (b), (c), and (d) in Fig. 2, we aim to illustrate what part of the reconstructed observation each of the latent variables z_t and w_t is responsible for. From (b), we observe that only the layout of the maze is reconstructed when we pass zeros as input to the decoder $p_\theta(o_t|w_t, z_t)$. Moreover, there are smeared traces of red color over the mazes, reminiscent of the agent depiction.

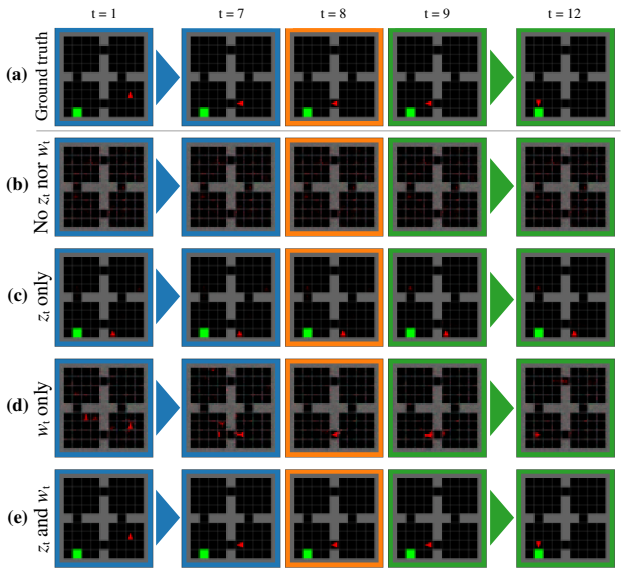
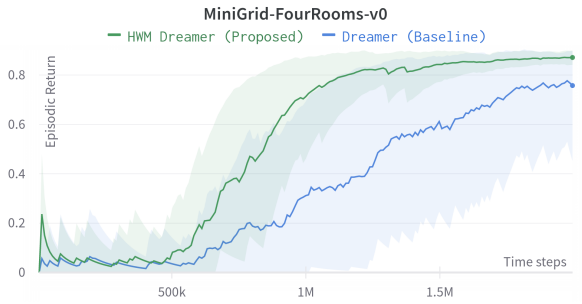


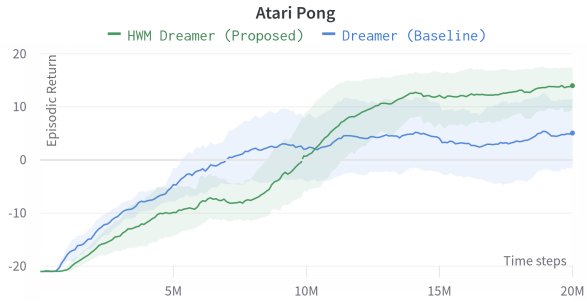
Figure 2. Illustrative example of the reconstruction and segmentation of an arbitrary trajectory performed by the proposed HWM. Each color represents a segment that corresponds to a temporally abstract state s_t. The segmentation is performed using the learned prior boundary distribution $m_t \sim p_\theta(m_t)$. Intermediate steps of long segments are omitted. Row (a) shows the ground truth observation of the trajectory. From row (b) to (e), the hierarchically structured components z_t and w_t of the HWM's state belief are progressively turned on just before reconstructing the observed frame.

When conditioning the decoder on the high-level state z_t only, the goal tile, as well as a blurry depiction of the agent, appear in the reconstructions, illustrated by the line (c). On the other hand, when conditioning the decoder solely on w_t, not only is the goal absent from the frame but the position of the agent in the maze becomes ambiguous, as illustrated in line (d). These results suggest that the information about the goal and the room of the agent tend to be encoded at the higher level by z_t, while other, more *refined* information is encoded at the lower level by w_t. Notice also how the traces of red are present in rows (b) and (d) become absent once z_t has been incorporated in row (c). We attribute the red traces to the ambiguity regarding the information about the agent's position in the state representation used for the reconstruction. This further strengthens the observation that z_t also encodes information that contributes to precisely situating the agent in the maze, which is also corroborated by (Saxena et al., 2021).

One caveat would be that the role of encoding information about the agent seems to be shared by the combination of w_t and z_t. Namely, on line (c) at time step $t = 8$, the blurry agent is depicted in the room corresponding to the previous segment of $t \in \{1, ..., 7\}$. Concurrently, on line (d) at time



(a) Episodic return on *MiniGrid-FourRooms-v0* task



(b) Episodic return on *Atari Pong* task

Figure 3. Averaged episodic return of each agent across 4 seeds. The vertical axis holds the episodic return values, while the horizontal one shows the number of time steps (environment interactions).

steps $t \in \{7, 8, 9\}$, the agent is localized in the correct room, albeit with a less defined depiction. This could be solved by further refining the HWM's structure and using methods such as (Higgins et al., 2017; Zhao et al., 2017) to better disentangle the information captured toward different levels of the hierarchy.

5.2. Sample efficiency and performance improvement

Fig. 3a and Fig. 3b documents the averaged episodic return across four seeds for *MiniGrid-FourRooms-v0* and *Atari's Pong* task respectively.

In both cases, the *HWM Dreamer* (green line) that uses the proposed HWM instead of the reference world model (Hafner et al., 2020a;b) demonstrates a non-negligible gain in sample efficiency, when compared to the baseline *Dreamer* agent (blue line). Moreover, it also achieves the highest final performance with less variance across runs.

This would suggest that owing to its hierarchically structured internal state representation, the proposed HWM is further incentivized to learn a state representation that accelerates and stabilizes the learning of the actor-critic component.

6. Future work

In this section, we motivate how the proposed HWM can be further leveraged in the context of the HRL framework. This ties into bringing the decision-making process of HRL agents toward one that is more aligned with the hierarchically organized behavior observed in animals and humans, as described in Section 2.2.

To this end, let us first extend the proposed generative process in Eq. 1 to account for the decision-making of an HRL agent. The resulting generative process is illustrated in Fig. 5 of Appendix A, while being formally expressed as:

$$\begin{aligned} p(O, W, Z, M, A, E) = \prod_{t=1}^{H} & p(o_t|w_t, z_t)\, p(m_t|w_t, z_t, a_t) \\ & p(z_t|z_{t-1}, m_{t-1}, w_{t-1}, a_{t-1}) \\ & p(w_t|z_t, w_{t-1}, a_{t-1}) \\ & \pi^{\mathrm{L}}(a_t|w_t, z_t, e_t)\, \pi^{\mathrm{H}}(e_t|z_t). \end{aligned} \tag{11}$$

Marginalizing out the observation O, low-level state representation component W, boundary indicator M, and the low-level action A variables from Eq. 11 allows us to recover a *temporally abstract* MDP.

$$p(Z, E) = \prod_{\tau} p(z_\tau|z_{\tau-1}, e_{\tau-1})\, \pi^{\mathrm{H}}(e_\tau|z_\tau). \tag{12}$$

Such temporally abstract MDP can be considered as a simplified version of the task to solve. The motivation that lead to such abstraction mechanism is illustrated and intuited in Fig. 4.

Given the example of the *Four Rooms* task introduced in Section 4, this abstract MDP would correspond to a simpler problem of navigating between the eight rooms, as illustrated at the layer (c) in Fig. 4. Namely, the high-level policy π^{H} would thus be set to solve the task of making the overall agent reach a given room by guiding the low-level policy. At the low-level, the corresponding *sub-policy* expressed by π^{L} executes the necessary sequence of actions that concretizes the high-level instruction, as proposed in the HRL framework (Dayan & Hinton, 1992; Sutton et al., 1999; Barto & Mahadevan, 2003; Florensa et al., 2017; Li et al., 2019).

By doing so, we expect the resulting *HWM HRL Dreamer* agent to expedite the exploration process in virtue of said exploration being conducted over a temporally abstracted and principled state space. We believe this also happens to align with how a human would decompose such a task, and learn how to solve it efficiently, as described in Section 2.2.

Appendix A supplements the theory presented in this section with the complete derivation of the objective functions for each components of the *HWM HRL Dreamer*.

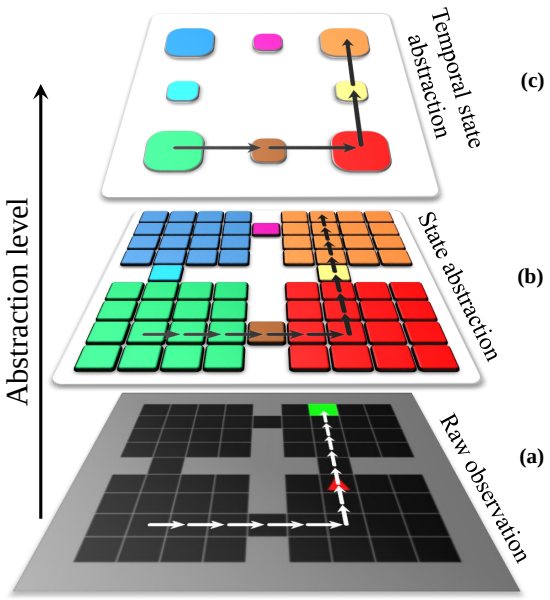


Figure 4. Conceptual diagram based on the *Four Rooms* task, illustrating the different levels of abstractions that the proposed HWM is designed to model. At the level of raw pixel-based observations (a), the optimal trajectory of the agent from its starting room to the goal tile is depicted using white arrows. From that layer (a), we can conceptualize a simpler state abstraction (b) that focuses on each cell of the maze that the agent can traverse. This abstraction layer would be equivalent to the low-dimensional latent space that a world model such as the one in Dreamer learns. Finally, layer (c) represents the temporally abstract state space that the proposed HWM is specifically designed to model. At this layer, all the cells that form a *room* of the maze are considered as an aggregate state. This simplifies the original task to the problem of *navigating from the starting room to the room that contains the goal (exit)*.

7. Conclusion

In this work, we leveraged the recent progress in world modeling methods and the framework of variation temporal abstraction to propose the hierarchical world model (HWM). The proposed model captures both temporally abstract and granular dynamics, as well as the corresponding hierarchically structured state representations.

Owing to its design, the HWM maintains the ability of world model methods (Ha & Schmidhuber, 2018; Hafner et al., 2018; 2020a;b) to provide a compact state representation that is also hierarchically structured for standard MbRL agents. This was verified in our preliminary experiments, which demonstrated an increase in sample efficiency and final performance for a Dreamer agent that used the proposed HWM instead of its original world model component. A complementary set of experiments also show that the proposed HWM can learn semantically meaningful, temporally abstract state representations that happen align with human understanding of the task to solve.

Additionally, we posit that the adaptive temporal abstraction mechanism provided by the HWM can be extended onto an HRL agent, thus building toward a human cognition-based, hierarchically structured decision-making process. We also provide the theoretical arguments underlying the extension of the proposed HWM with the HRL framework

Ongoing and future endeavors will consist of further refining the theoretical components of such a framework, and empirically verifying its properties. We expect the resulting *HWM HRL Dreamer* agent to yield higher sample efficiency and final performance compared to traditional HRL and MbRL methods, thus broadening the range of tasks that can be reliably solved and automated by RL-based agents.

Acknowledgements

This work was partially supported by the Japan Science and Technology Agency (JST) PRESTO Program (JPMJPR21C7), JST-Mirai Program (JPMJMI20B8), the Japan Society for the Promotion of Science (JSPS) KAKENHI 19H04172, and the JSPS KAKENHI 19K20344, Japan.

Rousslan F.J. Dossa thanks the Japanese Ministry of Education, Culture, Sports, Science, and Technology (MEXT) for its scholarship grant from the year 2017 to 2023.

References

Barto, A. G. and Mahadevan, S. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 2003.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *"Journal of Artificial Intelligence Research"*, 2013.

Botvinick, M. and Weinstein, A. Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2014.

Botvinick, M. M., Niv, Y., and Barto, A. G. *Hierarchically organised behaviour and its neural foundations: a reinforcement-learning perspective*, pp. 264–299. "Cambridge University", 2011.

Chevalier-Boisvert, M., Willems, L., and Pal, S. Minimalistic gridworld environment for openai gym, 2018.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2018.

Dayan, P. and Hinton, G. E. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*. Morgan-Kaufmann, 1992.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. *CoRR*, abs/1704.03012, 2017.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Ha, D. and Schmidhuber, J. World models. *CoRR*, abs/1803.10122, 2018.

Haarnoja, T., Hartikainen, K., Abbeel, P., and Levine, S. Latent space policies for hierarchical reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018a.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018b.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *ArXiv*, abs/1912.01603, 2020a.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models, 2020b.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations, 2017*, 2017.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 1997.

Kim, T., Ahn, S., and Bengio, Y. Variational Temporal Abstraction. *arXiv e-prints*, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., and Tenenbaum, J. B. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909, 2018.

Li, A. C., Florensa, C., Clavera, I., and Abbeel, P. Sub-policy adaptation for hierarchical reinforcement learning. *CoRR*, abs/1906.05862, 2019.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning, 2013.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 2021.

Saxena, V., Ba, J., and Hafner, D. Clockwork variational autoencoders. In *Advances in Neural Information Processing Systems*, 2021.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*. Journal of Machine Learning Research, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Sun, X. and Bischl, B. Tutorial and survey on probabilistic graphical model and variational inference in deep reinforcement learning. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 1999.

Tomov, M. S., Yagati, S., Kumar, A., Yang, W., and Gershman, S. J. Discovery of hierarchical representations for efficient planning. *PLOS Computational Biology*, 2020.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. Journal of Machine Learning Research, 2017.

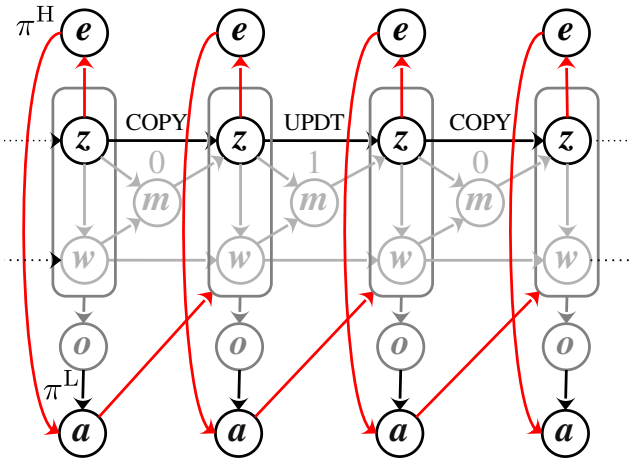


Figure 5. Dynamic Bayesian network representing the generative process combining the HWM and HRL frameworks. The red line emphasizes the influence of the high-level policy's action over the high-level, temporally abstract state z's transition modeled by the proposed HWM.

Xia, L. and Collins, A. G. E. Temporal and state abstractions for efficient learning, transfer and composition in humans. *bioRxiv*, 2020.

Zhao, S., Song, J., and Ermon, S. Infovae: Information maximizing variational autoencoders. *ArXiv*, abs/1706.02262, 2017.

A. Appendix A: Training objective of an *HWM HRL Dreamer* agent

In this appendix section, we supplement Section 6 with the derivation of the complete training objective for the desribed *HWM HRL Dreamer* agent. From the theoretical perspective, the temporally abstract MDP derived in Eq. 12 happens to exactly match the standard MDP formulation upon which RL algorithms are based upon (Sutton & Barto, 2018). Consequently, any RL method that is proven to work with a task that follows the standard MDP formulation can also be applied at the temporally abstract layer. The proposed *HWM Dreamer* agent described in Section 3.3 thus appears a natural fit for such extension. First, the state value estimator corresponding to the low-level policy is overloaded with the high-level action (Schaul et al., 2015). By doing so, the state value estimator at the low level approximates the value of being in a given state when following an arbitrary instruction from the high-level policy. Formally, this is equivalent to replacing all occurrences of $V_\psi(w_h, z_h)$ in Eq. 7 with $V^{\mathrm{L}}_\psi(w_h, z_h, e_h)$. We refer to this overloaded state value objective function as $J^{\mathrm{L}}_{\mathrm{V}}(\psi)$ to indicate that it corresponds to the low-level policy π^{L}_ψ. Namely,

$$J^{\mathrm{L}}_{\mathrm{V}}(\psi) = \mathbb{E}_{p_\theta(w_t, z_t)}\Big[\sum_{h=t}^{t+H} \frac{1}{2}\big|\big|V^{\mathrm{L}}_\psi(w_h, z_h, e_h) - V^{\mathrm{L},h}_\lambda\big|\big|^2\Big] \tag{13}$$

Similarly, the actor's objective function $J_\pi(\psi)$ defined in Eq. 8 is also overloaded to account for the guidance of the low-level policy π^{L}_ψ by the high-level policy π^{H}_ψ as follows:

$$J^{\mathrm{L}}_{\pi}(\psi) = \mathbb{E}_{p_\theta(w_t, z_t)}\Big[\sum_{h=t}^{t+H} -\log \pi^{\mathrm{L}}_\psi(a_t|w_h, z_h, e_h)\, \mathrm{sg}(V^{\mathrm{L},h}_\psi) - \eta\, \mathcal{H}\big[a_h|w_h, z_h, e_h\big]\Big]. \tag{14}$$

At the high-level, the policy π^{H}_ψ outputs a high-level action based on the high-level component Z of the proposed hierarchical state representation described in Section 3.2. The learning objective for the high-level state value estimator V^{H}_ψ is thus derived following the same principles as the for the traditional state value case described in Section 3.3. This is formally expressed as follows:

$$J^{\mathrm{H}}_{\mathrm{V}}(\psi) = \mathbb{E}_{p_\theta(z_t)}\Big[\sum_{\tau=t}^{(t+H)/k} \frac{1}{2}\big|\big|V^{\mathrm{H}}_\psi(z_\tau) - V^{\mathrm{H},\tau}_\lambda\big|\big|^2\Big] \tag{15}$$

where τ designate a time step at the coarser, temporally abstract level. To simplify the notation, it is assumed to temporally abstract transition occur at a fixed interval k, hence the values of τ ranging from the arbitrary time step t to $(t + H)/k$, with H being the maximal length of a trajectory simulated using the learned dynamics. In practice, such transitions are of variable length and dictated by the prior distribution over the boundary indicator described in Section 3.2.

Subsequently, the learning objective for π^{H}_ψ is similarly deduced from Eq. 8 as follows:

$$J^{\mathrm{H}}_{\pi}(\psi) = \mathbb{E}_{p_\theta(z_t)}\Big[\sum_{\tau=t}^{(t+H)/k} -\log \pi^{\mathrm{H}}_\psi(e_\tau|z_\tau)\, \mathrm{sg}(V^{\mathrm{H},\tau}_\psi) - \eta\, \mathcal{H}\big[e_\tau|z_\tau\big]\Big]. \tag{16}$$

Aggregating the actor-critic losses for the high and low-level policies as

$$J^{\mathrm{L}}_{\mathrm{AC}}(\psi) = J^{\mathrm{L}}_{\pi}(\psi) + J^{\mathrm{L}}_{\mathrm{V}}(\psi)$$

and

$$J^{\mathrm{H}}_{\mathrm{AC}}(\psi) = J^{\mathrm{H}}_{\pi}(\psi) + J^{\mathrm{H}}_{\mathrm{V}}(\psi)$$

respectively, the final training objective for an *HWM HRL Dreamer* agent can be expressed as:

$$J_{\text{HWM-HRL}(\theta,\phi,\psi)} = J_{\mathrm{HWM}}(\theta, \phi) + J_{\mathrm{R}}(\theta, \phi) + J^{\mathrm{L}}_{\mathrm{AC}}(\psi) + J^{\mathrm{H}}_{\mathrm{AC}}(\psi) \tag{17}$$