Contribution Title

First Author[1][0000−1111−2222−3333], Second Author[2,3][1111−2222−3333−4444], and Third Author[3][2222−−3333−4444−5555]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Reinforcement learning (RL) has achieved remarkable success in tasks with fully observable environments, yet many real-world scenarios present significant partial observability challenges. In a partially observable Markov decision process (POMDP) [4], the agent receives observations that provide only incomplete information about the underlying state, forcing it to rely on internal memory or belief states to make decisions. Addressing partial observability is critical, as agents without a proper internal model of the environment can suffer from unstable or suboptimal policies. Traditional solutions incorporate recurrent neural networks into RL agents to integrate information over time [3], or maintain explicit belief state estimations via Bayesian updates. However, these approaches often struggle to generalize or to efficiently capture complex temporal dependencies when observations are high-dimensional and noisy.

An emerging approach to tackling partial observability is to learn a *world model* of the environment that serves as the agent's internal representation of state [1]. World models aim to capture the dynamics and structure of the environment in a latent space, enabling the agent to predict future observations and rewards from past interactions [1, **?**]. By learning a compact latent state that summarizes the history, an agent can effectively plan and act despite missing immediate information, as the latent state acts as a memory. Model-based RL methods leveraging learned world models have shown improved sample efficiency and generalization in visual control tasks [2, **?**]. Nonetheless, learning accurate and robust world models remains challenging, especially in the presence of irrelevant or distracting features in observations, as the agent must learn to attend to latent factors that are predictive of future outcomes [5].

Meanwhile, the principle of *predictive coding* has gained traction as a powerful framework for representation learning in both neuroscience and machine learning [7, **?**]. Predictive coding theories posit that neural systems continually predict their incoming sensory input and encode only the deviations (prediction errors), thereby efficiently utilizing representational capacity [7]. This idea has been translated into neural network models that learn by predicting future inputs and minimizing prediction error at each time step [**?**,**?**]. By training models to anticipate their next observation, predictive coding naturally yields internal representations that capture temporal structure and causality in the data. Indeed, predictive models have been leveraged as unsupervised objectives for learning useful features and as a basis for memory architectures [10, **?**].

Recently, *Temporal Predictive Coding* (TPC) has been proposed as an extension of predictive coding for sequential processing and memory [9]. Tang et al. [9] introduced TPC as a biologically plausible model of sequential memory, demonstrating that a TPC network can memorize and accurately recall sequences of inputs by continually predicting each next element in the sequence. The TPC model integrates recurrent dynamics with a predictive coding objective, allowing it to form stable latent representations of temporal contexts. Notably, TPC was shown to be theoretically equivalent to a classical Asymmetric Hopfield Network with an implicit whitening of inputs, which endows it with greater capacity and stability for sequence storage [9]. These properties suggest that TPC can maintain an internal model of a temporal process – effectively a learned world model of sequence dynamics – using a local prediction-based learning rule.

Given these advances, an intriguing question is whether TPC can serve as a useful world model for an RL agent operating in a POMDP. In this paper, we investigate the integration of Temporal Predictive Coding networks into the representation learning backbone of an RL agent. The motivation is that a TPC-based world model could enable the agent to predict and infer hidden state information, filtering out unpredictable distractions and focusing on task-relevant dynamics [5]. We hypothesize that by training an agent to internally predict its next observations (or latent features thereof), the agent will learn a latent state representation that captures the underlying environment state needed for decision-making. Our approach combines a TPC objective with standard RL optimization, aiming to imbue the agent with a predictive memory that can better tackle partial observability.

In summary, our contributions are as follows:

- We propose a novel agent architecture that incorporates a Temporal Predictive Coding network as a world model for reinforcement learning in partially observable environments.
- We provide a comprehensive analysis of the learned representations, showing that the TPC-based world model captures latent state information and improves memory over standard recurrent baselines.
- Through experiments on challenging POMDP benchmarks, we demonstrate that agents equipped with TPC world models achieve superior performance

and sample efficiency compared to alternative approaches, particularly in the presence of high-dimensional observations and distracting irrelevant features.

2 Background

2.1 Reinforcement Learning in Partially Observable Environments

In a POMDP, the true state s_t of the environment is not directly observable; instead, the agent receives an observation o_t that is a stochastic function of the underlying state [4]. The agent must therefore rely on its observation history $o_{1:t}$ (or a belief distribution over states) to infer s_t for making optimal decisions. This challenge has led to various strategies for integrating memory into RL agents. One common approach is to equip agents with recurrent neural networks (e.g., LSTM or GRU) that process sequences of observations, thereby learning an implicit state representation [3, **?**]. Another approach is to explicitly maintain a probabilistic belief state and update it through Bayesian filtering or neural inference networks [**?**]. In either case, the objective is to approximate the information state – a sufficient statistic of the history – that can inform the policy [4, **?**]. However, learning such state representations purely from sparse reward signals can be difficult. This is where auxiliary objectives and inductive biases are often introduced to guide representation learning.

2.2 World Models and Representation Learning in RL

Model-based RL seeks to learn a model of the environment's dynamics (and sometimes the reward function) to facilitate planning and more efficient policy learning. In high-dimensional observation spaces (e.g. raw pixels), it is intractable to model the dynamics directly in pixel space; instead, methods learn a latent dynamics model, often termed a *world model* [1]. A seminal example is the World Models framework by Ha and Schmidhuber [1], which trained a variational autoencoder and a recurrent network to encode observations into a latent state and predict future observations, allowing a controller to plan in the learned latent space. Follow-up works have significantly advanced latent world models for RL: for instance, PlaNet and Dreamer [2, **?**] learn recurrent state-space models (RSSMs) that predict forward in a compact latent space and enable planning or policy learning through imagined rollouts. These models have achieved state-of-the-art results on various continuous control benchmarks from pixels, highlighting the value of learned representations that capture the environment's dynamics. A key element of such approaches is the use of auxiliary representation learning losses: the agent is trained not only to maximize reward but also to reconstruct observations or predict future outcomes, which shapes the latent state to be informative about the environment's hidden state and dynamics [2, **?**,**?**]. Indeed, auxiliary tasks like forward prediction of observations, reward prediction, or contrastive estimation of future state similarity [6, **?**] have been found to improve the quality of learned representations in partially observable settings.

An important insight from these studies is that not all aspects of an observation are relevant for prediction and control; an effective world model should filter out noise or irrelevant features and focus on predictive information. For example, Nguyen et al. [5] propose a temporal predictive coding method that maximizes mutual information between latent states across time steps, effectively encouraging the agent to encode those elements of the observation that have consistent predictive value for future states. By concentrating on temporally persistent and predictive features, the learned latent space becomes more robust to nuisance variability (e.g., background distractions) and more useful for downstream decision-making. Similarly, approaches like Self-Predictive Representations (SPR) in RL use an auxiliary loss to predict the agent's own next latent state, which has been shown to accelerate learning by capturing task-relevant information in the latent dynamics [8].

2.3 Predictive Coding and Sequential Memory

Predictive coding is a theory originating from neuroscience which postulates that brains achieve efficient information processing by continually predicting incoming sensory signals and adjusting internal beliefs based on the ensuing prediction errors [7, **?**]. This framework has inspired computational models wherein each layer of a neural network attempts to predict the activity of the next layer or the sensory input, and only the deviation (error) is propagated for learning [7]. Such models naturally extend to sequential data by iteratively minimizing prediction errors over time, which aligns well with the needs of an agent in a dynamic environment. In machine learning, predictive coding principles have been applied to build unsupervised sequence learning architectures. For instance, deep predictive coding networks have been used for video prediction, where a model learns to anticipate future frames and thereby learns robust visual features [**?**]. Contrastive predictive coding [6] similarly trains a network to predict future latent representations of observations, yielding powerful representations for downstream tasks.

In the context of memory and sequential pattern learning, predictive coding has shown promise as the basis for storing temporal information. The Temporal Predictive Coding model proposed by Tang et al. [9] is a recent example: it is a recurrent network trained under a predictive coding objective to remember sequences of inputs. At each time step, the TPC network generates a prediction of the next input and uses the prediction error to update its latent state, effectively "locking in" information about the sequence. This mechanism enables the TPC to encode rich temporal contexts and to accurately recall or regenerate the sequence from partial cues [9]. Moreover, analytical results suggest that TPC can be interpreted as an Asymmetric Hopfield Network performing a form of whitening on its inputs, which gives it a greater capacity and stability for sequential memory compared to standard recurrent autoencoders [9]. In summary, TPC provides a principled way to learn a recurrent latent state that preserves information over long sequences by continually minimizing prediction errors. Notably, predictive coding networks like TPC rely on local learning rules (each

prediction error only updates adjacent layers), a property that is attractive for aligning with biological plausibility and for potential on-line learning advantages [10].

2.4 Predictive Representations for Reinforcement Learning

The convergence of ideas from world models in RL and predictive coding suggests a promising direction: using predictive coding networks as internal world models for agents. If an agent's internal model is trained to minimize the prediction error of its sensory observations (or features thereof), the model's latent state should capture precisely those aspects of the environment state that are necessary to predict future observations. This concept is closely related to the notion of predictive state representations (PSRs) in reinforcement learning, where the state is represented by predictions of future observable events [**?**]. Modern deep RL methods have echoed this idea: by learning models that predict observations or latent features, agents implicitly learn belief states that carry task-relevant information [**?**,**?**]. The temporal predictive coding approach by Nguyen et al. [5] discussed above exemplifies this synergy, as it integrates a predictive learning objective into model-based RL.

In our proposed approach, we take this idea further by explicitly incorporating a TPC-based world model into the agent. The TPC network serves as a sequence model that continually predicts the next observation (or encoded latent), and the resulting prediction errors update its internal state. By training the agent with this auxiliary predictive coding objective alongside the standard RL objective, we aim to ensure that the agent's latent state (provided by the TPC model) functions as a high-fidelity belief state of the POMDP. In essence, the TPC is encouraged to fill in the missing pieces of the state from observations, using prediction to enforce consistency over time. We hypothesize that this leads to better policy performance in partially observable tasks, as the agent can rely on the predictive world model to recall important information and ignore ephemeral distractions. In the following, we detail the architecture and training procedure of the TPC-augmented agent and evaluate its performance on benchmark tasks.

3 Temporal Predictive Coding in a RL Setup

We use a Temporal Predictive Coding Network (tPCN), extending classical Predictive Coding Networks (PCNs) into the temporal domain (Millidge et al., 2024; Tang et al., 2024), for learning spatial representations in grid-world tasks. The architecture of the two-layer tPCN is shown in Fig. 1.

The model receives as input a sequence of encoded agent states represented as one-hot vectors corresponding to the color-coded positions in a grid-world environment, denoted as $s_1, \ldots, s_T$, along with actions taken by the agent at each timestep, denoted as $a_1, \ldots, a_T$. The tPCN minimizes the following loss function at each time step t:

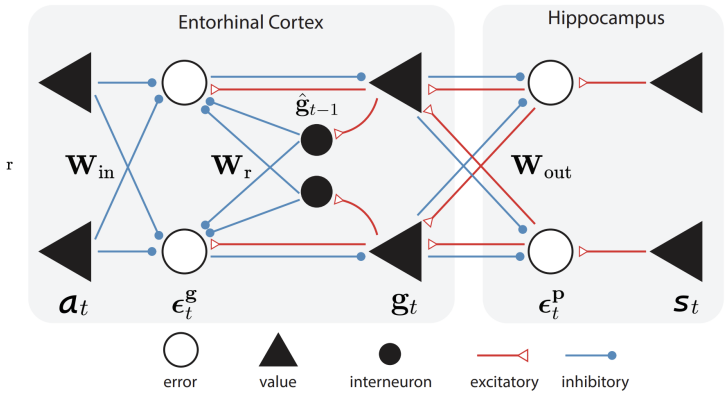

Fig. 1. Architecture of the two-layer Temporal Predictive Coding Network (tPCN).

$$L_{tPCN,t} = \|s_t - f(W_{out}g_t)\|_2^2 + \|g_t - h(W_r\hat{g}_{t-1} + W_{in}a_t)\|_2^2, \tag{1}$$

where f and h are nonlinear activation functions, and W_{in}, W_r, W_{out} are weight matrices forming predictions. Prediction errors are defined as:

$$\epsilon_t^s := s_t - f(W_{out}g_t), \quad \epsilon_t^g := g_t - h(W_r\hat{g}_{t-1} + W_{in}a_t). \tag{2}$$

Training occurs in two stages. First, the prediction error with respect to the latent state g_t is minimized via gradient descent:

$$\Delta g_t \propto -\nabla_{g_t} L_{tPCN,t} = -\epsilon_t^g + W_{out}^\top f'(W_{out}g_t)\epsilon_t^s, \tag{3}$$

then weights are updated according to:

$$\{\Delta W_{out}, \Delta W_r, \Delta W_{in}\} \propto -\nabla_{\{W_{out},W_r,W_{in}\}} L_{tPCN,t} \tag{4}$$

$$= \{f'(W_{out}g_t)\epsilon_t^s g_t^\top,\ h'(\tilde{g}_t)\epsilon_t^g \hat{g}_{t-1}^\top,\ h'(\tilde{g}_t)\epsilon_t^g a_t^\top\}, \tag{5}$$

where f' and h' are Jacobians of the nonlinear functions f and h, and $\tilde{g}_t := W_r\hat{g}_{t-1} + W_{in}a_t$.

After inference, the value $\hat{g}_t$ is updated to the optimized g_t to minimize the loss at the next timestep, $L_{tPCN,t+1}$. The model is trained on numerous trajectories $\{a_t, s_t\}$ and tested on unseen trajectories, computing predicted states:

$$g_t = h(W_r g_{t-1} + W_{in}a_t), \quad \hat{s}_t = f(W_{out}g_t). \tag{6}$$

The model's performance is evaluated based on the accuracy of predicted positions $\hat{s}_t$ and the latent activations g. In the case of linear f and h, these computations can be implemented with local neural circuitry and Hebbian learning rules, plausibly mapping onto hippocampal formation circuitry.

Importantly, this setup supports both offline and online prediction modes, making it applicable to a wide range of temporal inference tasks.

4 Experiments

In this section, we empirically investigate the ability of Temporal Predictive Coding (TPC) models to represent hidden states in partially observable environments. Our initial experiments focus on random agent trajectories in grid-world environments. The aim here is preliminary validation of the TPC's capability to internally reconstruct and represent an agent's true state in a POMDP setting, despite only partial and ambiguous observations.

4.1 Experimental Setup

We constructed grid-world environments with varying sizes (specifically, 3×3 and 5×5) where each cell was assigned a color from a limited palette randomly. The colors represent partial observations available to the agent and, crucially, the number of colors was smaller than the number of grid cells, creating ambiguity in observations. An example of a randomly colored environment (3×3 with 3 colors) is shown in Figure 2.

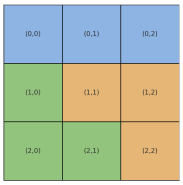

Fig. 2. Example of a randomly colored 3×3 grid-world environment with 3 distinct colors. Colors represent partial observations available to the agent.

Agents moved randomly, generating trajectories for analysis. The primary question was whether TPC could implicitly learn to distinguish between the true underlying states despite observing only ambiguous colors.

4.2 Analysis of Transition Accuracy

We first analyzed the accuracy of the TPC model's predictions about future observations given the current partial observation and action. Figure 3 illustrates the accuracy of predicting subsequent positions along random trajectories in a 5×5 environment across various numbers of colors.

From Figure 3, we observe a clear trend: the accuracy significantly improves after the first few steps, indicating the model gradually learns to leverage historical information to infer the true state more reliably.

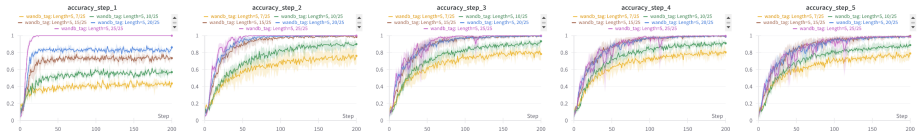

Fig. 3. Prediction accuracy of TPC model transitions for sequential steps in a 5 × 5 environment with different numbers of colors. Each subplot shows accuracy when predicting the subsequent position given current observation (color) and action.

Further insights into these predictions are provided by heatmaps representing prediction accuracy at each transition step (Figure 4). We specifically illustrate the scenario with 15 colors in a 25-position environment.

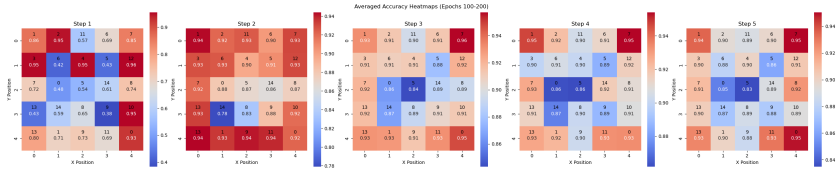

Fig. 4. Heatmaps illustrating TPC prediction accuracies for each transition step in a 5 × 5 environment with 15 colors.

The heatmaps further demonstrate that, as the agent progresses through subsequent steps, the model's internal representations become increasingly precise in identifying specific grid locations despite ambiguous inputs.

4.3 Latent Representation Clustering

To verify that TPC internal states effectively capture and differentiate true positions over time, we applied t-SNE clustering to latent states accumulated during inference in the 3 × 3 environment (Figure 5).

The t-SNE visualization clearly illustrates that with each subsequent step, latent states become increasingly well-separated into distinct clusters corresponding to true grid positions.

Quantitative clustering metrics presented in Figure 6 support this observation, demonstrating convergence and stabilization of latent representations across steps.

4.4 Generalization of Latent Representations

Finally, we assessed the generalization capability of the latent representations by training logistic regression classifiers to predict true states from latent states. Figure 7 shows the resulting accuracies.

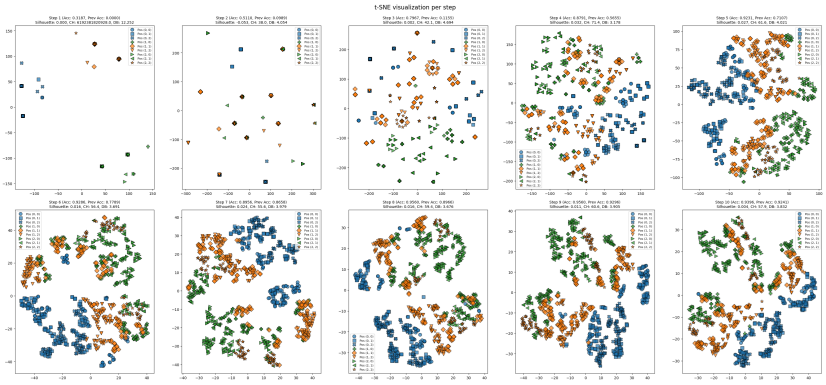

Fig. 5. t-SNE visualization of TPC latent states in a 3×3 environment. Points correspond to latent states collected at each step, colored by their true positions.

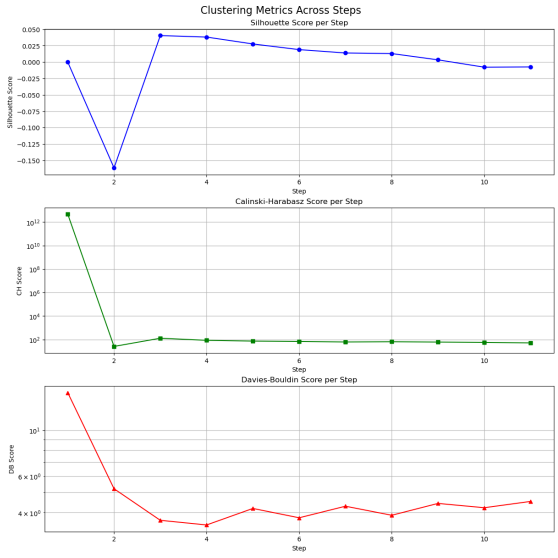

Fig. 6. Quantitative clustering metrics of TPC latent states over sequential steps in the 3×3 environment, demonstrating progressive stabilization.

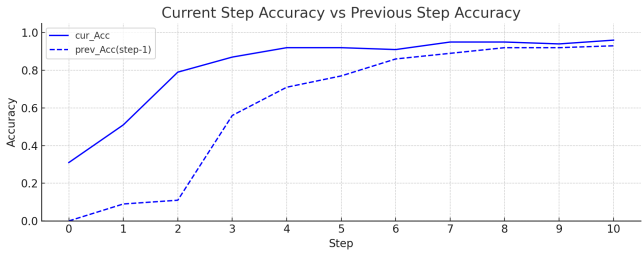

Fig. 7. Accuracy of logistic regression classifiers trained to predict true states from TPC latent representations at different steps in a 3×3 grid-world environment. Training on latent states from step n and testing on states from step $n-1$ demonstrates progressive generalization.

We observe from Figure 7 that predictive accuracy steadily improves across steps, and crucially, latent representations trained on later steps generalize increasingly well to previous steps. This result strongly suggests that TPC latent states evolve to robustly encode underlying environment structure, effectively generalizing beyond immediate temporal contexts.

Taken together, these experiments highlight the potential of TPC as a powerful internal representation model for RL agents in partially observable environments, motivating further exploration in more complex scenarios.

References

1. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
2. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: International Conference on Learning Representations (ICLR) (2020)
3. Hausknecht, M., Stone, P.: Deep recurrent q-learning for partially observable mdps. AAAI Fall Symposium Series (2015)
4. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101**(1-2), 99–134 (1998)
5. Nguyen, T., Tachet des Combes, R., Durand, A., Pineau, J.: Temporal predictive coding for model-based planning in latent space. arXiv preprint arXiv:2106.07156 (2021)
6. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. In: NeurIPS. pp. 1–12 (2018)
7. Rao, R.P., Ballard, D.H.: Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience **2**(1), 79–87 (1999)
8. Schwarzer, M., Anand, A., Goel, R., Hjelm, D., Courville, A., Bachman, P.: Data-efficient reinforcement learning with self-predictive representations. In: International Conference on Learning Representations (ICLR) (2021)
9. Tang, E., Mattar, M.G., Daw, N.D.: Sequential memory with temporal predictive coding. arXiv preprint arXiv:2306.09852 (2023)

10. Whittington, J.C., Bogacz, R.: Theories of error back-propagation in the brain. Trends in Cognitive Sciences **23**(3), 235–250 (2019)