Neuro-Symbolic Planning under Uncertainty in Unknown Models

David Hudák
Brno University of Technology
Brno, Czech Republic
ihudak@fit.vutbr.cz

Maris F. L. Galesloot
Radboud University
Nijmegen, The Netherlands
maris.galesloot@ru.nl

Martin Tappler
TU Wien
Vienna, Austria
martin.tappler@tuwien.ac.at

Milan Češka
Brno University of Technology
Brno, Czech Republic
ceskam@fit.vutbr.cz

ABSTRACT

Autonomous planning in uncertain environments poses a significant challenge. While model-free deep reinforcement learning scales to large problems, it does not provide any theoretical or practical guarantees; its scalability depends on high computational demands and the use of highly efficient simulators; and its results are hard to explain. On the other hand, formal model-based methods provide strong theoretical guarantees but do not scale to standard real-world planning problems and require significant external input in the form of the model. Recent research, primarily represented by Dreamer-like architectures, has proposed leveraging learnable abstractions to efficiently learn policies via policy gradient methods over imagined trajectories. This approach has led to significantly better sampling efficiency, but it remains a deep learning approach without guarantees or explanations, as the learned abstractions are represented by intricate neural networks. In this work-in-progress paper, we propose a novel model-learning methodology to obtain smaller, fully discrete abstractions, enabling us to use both model-based and model-free methods within a tight neuro-symbolic planning loop to achieve more robust and explainable behavior under model uncertainty.

KEYWORDS

Model Learning; Planning under Uncertainty; Neuro-Symbolic AI; Abstraction; Safety; Explainability; POMDP; Model-Based RL

1 INTRODUCTION

Planning or sequential decision-making in environments under uncertainty, formally modeled by a partially observable Markov decision process (POMDP) [24], is an active research area. The general framework of POMDPs can model several real-world tasks, ranging from healthcare [46] to robotics [27] and autonomous driving [36]. The solution to a POMDP is represented by a policy, a function that maps an observation history to an action. However, the problem of solving POMDPs, i.e., computing a policy that maximizes the expected reward from the POMDP, is intractable in general [29]. The primary reason is the potentially infinite memory to infer optimal actions.

In practice, there exist two orthogonal methodologies for planning under uncertainty [26]. The first option, formal model-based methods, uses the exact model's knowledge to compute a suboptimal policy by solving the POMDP using belief-based approximations [21] or by exploring a set of finite-state controllers [2, 45]. Those methods, combined with the known model, provide strong theoretical guarantees but scale only to relatively small models. On the other hand, model-free methods, now most notably represented by various deep reinforcement learning (DRL) algorithms such as proximal policy optimization (PPO) [35] or soft actor-critic (SAC) [14], combined with recurrent neural networks, scale to realistic problem sizes. However, DRL offers no practical theoretical guarantees, and the resulting neural polices are opaque.

Recently, the gap between the model-based and model-free approaches has been decreasing. Most notably, as we further describe in related work, algorithms such as MuZero [34] and Dreamer [18, 19] introduce model-based approaches that learn an abstracted model using neural networks. Those algorithms have achieved state-of-the-art performance across multiple challenging benchmarks. Indeed, those methods still rely on hard-to-explain functional approximations of the true environments. On the contrary, the concept of neuro-symbolic AI [40] leverages the synergy between deep learning and formal symbolic methods, e.g., by deep learning of finite symbolic representations combined with symbolic model-based reasoning over them, leading to a more explainable process given the symbolic nature of learned model representations and policies. For example, Hudák et al. [22] introduced a robust planning loop that combines DRL with finite-state controller extraction and symbolic model-based analysis to achieve scalable, robust, and explainable policies.

Planned Contributions. While the direction of neuro-symbolic AI provides a strong foundation for robust and scalable planning, the core limitation of the neuro-symbolic concept in general [7, 8, 12, 22, 40] is the challenge of learning suitable models. Such models are crucial to provide explainable policies [44], to further improve safety guarantees of existing policies with shielding [1, 13] or robust improvement over worst-case scenarios [11]. The ultimate goal of this work is to reduce this gap by learning (robust) discrete model abstractions from continuous systems with weaker assumptions about how to encode observations into an abstract space or about the true state of the original system. Particularly, we plan those three main contributions:

(1) Designing a novel neural network architecture with the aim of emulating discrete and finite POMDPs.

Proc.@ of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

(2) Designing a novel robust model-learning procedure with the aim to provide probably approximately correct (PAC) guarantees of the learned representation.
(3) Learning robust and explainable policies for unknown environments using the learned abstractions.

1.1 Related Work

Model-Based Reinforcement Learning. The general idea of combining deep learning and planning over abstract state spaces has recently achieved state-of-the-art performance across various tasks. MuZero [34] proposed a model-based planning approach using learned representations and dynamics networks to predict behavior via a Monte Carlo tree search (MCTS) algorithm with success across various games such as shogi, chess, go, and Atari. More recently, the family of Dreamer algorithms [15, 17–19] achieved significant success in complex long-term planning tasks through learned world models and succeeded in the challenge of mining diamonds in Minecraft without any prior user inputs compared to former model-based approaches, e.g. VPT [4], which used data from human contractors to beat the same benchmarks. In our work, we plan to push the concept of model-based reinforcement learning further by learning completely discrete models, enabling compatibility with formal model-based tools for stronger explainability.

Learning Discrete (Finite) Models. The method most closely related to discrete model learning is the framework of Wasserstein Autoencoded MDPs (WAE-MDPs) [10]. It converts complex continuous states into a discrete binary representation using continuous relaxations of the Bernoulli distribution and Gumbel Softmax reparametrization for latent-action encoding. The training process employs bisimulation between the real and the abstract environments to minimize the Wasserstein distance between trajectory distributions, thereby avoiding several fatal issues, such as posterior collapse or poor dynamics estimation. However, the limitation of computing the Wasserstein distance is its cubic $O(n^3)$ complexity [43], which means the distance must be estimated for complex models, leading to inaccurate abstractions. Regarding partial observability, which is more challenging, Avalos et al. [3] proposes Wasserstein Believers, which extends Wasserstein Autoencoders to learn belief representations; however, the state space of the underlying environment must be known during training. The aim of our work, compared to the mentioned approaches, is to develop novel discrete and finite model-learning methods that extract POMDPs without requiring access to the underlying hidden state during training or a handcrafted feature-extraction function. The main assumption of our planned work is that there is a finite set of distinguishable scenarios that allow us to describe potential behavior in the environment and produce (sub-)optimal policies.

Safe Planning with (Safety) Model Approximations. The idea of learning model representations to improve the safety guarantees is an active research field. Notably, the concept of world models similar to DreamerV3 architecture was used for shielding [1] in the approach of approximate model-based shielding (AMBS) [13]. It uses model approximation to improve the safety guarantees of learned policies by computing expected outcomes in the model abstraction using simulations. Alternatively, in the context of partially observable stochastic games (POSG), Yan et al. [44] proposes a different notion of neuro-symbolic POSGs (NS-POSG) to enable the use of model-based heuristic search value iteration methods in complex continuous domains; indeed, as the authors mention, the method is still highly computationally demanding.

Algorithm 1 Overview of neuro-symbolic planning loop

```
1:  function LearnModel(π_Exp : Policy, S: Simulator)
2:      T ← Simulate(S, π)                          ▷ Batch of real trajectories
3:      GSSM ← LearnModel(T)                        ▷ See Section 3
4:      M ← ConstructModel(GSSM)                    ▷ POMDP def. in [9]
5:      return M, GSSM

6:  function GetAction(E: Encoder, π_Abs: Policy, z : Observation)
7:      o ← E(z)                                    ▷ Map observation to abstract space
8:      a ← π_Abs(o)                                ▷ Simplified Memoryless Policy
9:      return a

10: function EvalAbstract(S: Simulator, GSSM, π_Abs : Policy)
11:     i ← 0
12:     r_total ← 0
13:     z ← S.reset()
14:     while ¬ StepLimit(i) do
15:         a ← GetAction(GSSM.encoder, π_Abs, z)
16:         z, r ← S.step(a)
17:         r_total ← r_total + r
18:         i ← i + 1
19:     return r_total

20: function PlanningLoop(S: Simulator)
21:     π_Exp ← Init()
22:     M, GSSM ← LearnModel(π_Exp, S)              ▷ Initial abstraction
23:     while ¬ TimeOut() do
24:         π_Abs ← Synthesize(M)                   ▷ Using Storm
25:         r_new ← EvalAbstract(S, GSSM, π_Abs)
26:         π_Exp ← UpdateExplorer(π_Exp, π_Abs, GSSM)
27:         M, GSSM ← LearnModel(π_Exp, S)
28:     return GSSM, π_Abs
```

2 TOWARDS NEURO-SYMBOLIC PLANNING

The Idea. Algorithm 1 sketches the proposed neuro-symbolic planning loop. The high-level process resembles the learning processes of the Dreamer [15], as we initially approximate the model using an exploration policy to generate data for the first model approximation for abstract policy synthesis. Next, as shown in the while loop starting at Line 23, we iteratively collect more data using an exploration policy to improve the model abstraction and, in turn, the abstract policy. The only assumption about the training process is the existence of a simulator or another interactive environment representation providing trajectories. Furthermore, we assume that the problem can be formulated as multiple distinguishable abstract scenarios, i.e., practically, that the discrete representation of the unknown model contains at least sub-optimal policies in the original environment.

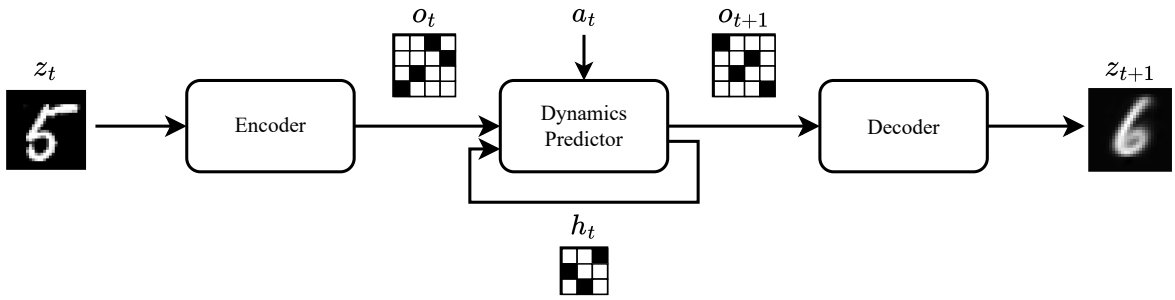


Figure 1: High-Level Architecture of the Gumbel state space model (GSSM).

The Model. One of the crucial aspects of the model learning is the final form of the learned model. While Delgrange et al. [10] treat the output as a fully specified MDP, the Dreamer [18] architectures consider the general framework of POMDPs. However, for planning, the Dreamer models assume the agent knows the hidden state of the learned model and thus learns memoryless policies as if it were an MDP. Furthermore, since Hafner et al. [17] introduced discrete latent representations produced by stochastic encoders, we might also consider the task as a belief-MDP [24]. In our work, we aim to explore the most suitable formulation of the learned models to achieve the safest behavior.

Exploration Policy. The model-learning algorithms are not strictly offline or online and can be executed with any reasonable, well-distributed trajectory buffer. However, we differ from existing training loops in our focus on high-fidelity exploration policies, since both Wasserstein Autoencoders and Dreamer primarily prioritize maximizing performance and improving model abstraction in the most-visited regions of the model. In our setting, we focus on building a precise representation of the unknown model of the overall environment to enable more complex model-based analysis methods and achieve both explainable and robust policies. In this regard, we plan to experiment with the ideas of curiosity-driven learning [6] and robust reinforcement learning [32], with the aim of generating more data where the model's abstraction is least precise.

Synthesis and Evaluation of Abstract Policies. The policy synthesis in the abstract space is independent of the abstraction encoding process, and the policy might be trained or synthesized with any (PO)MDP solving method. In our preliminary experiments, we use the Storm [20] model checker. However, it should be noted that the evaluation of a policy depends on the abstraction encoding process, as shown in Algorithm 1 at Line 6. Furthermore, the policy might leverage the learned underlying dynamics as an additional input to the decision-making process, e.g., via belief estimation.

3 MODEL LEARNING

In this section, we describe the recent model-learning method used in the DreamerV3 algorithm [18] and then discuss how to modify the architecture towards compact discrete model representations, inspired by the recent self-interpretable network approach proposed in [22].

3.1 Recurrent State Space Model

To represent a model abstraction, Hafner et al. [16] propose a general, learnable architecture for a recurrent state space model (RSSM), which was later extended in the Dreamer architectures, most notably in the DreamerV2 [17] and the DreamerV3 [18]. The architecture consists of 6 main components:

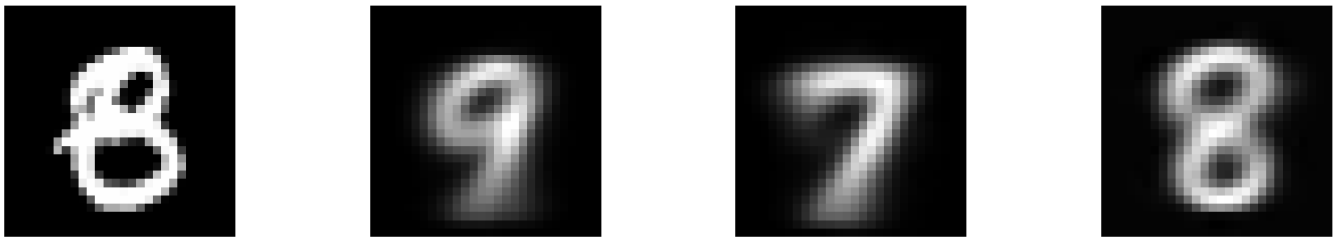
- Sequence model: $h_t = f_\theta(h_{t-1}, o_{t-1}, a_{t-1})$
- Encoder: $o_t \sim q_\theta(o_t|h, a_t)$
- Dynamics predictor: $o_t \sim p_\theta(o_t|h_t)$
- Reward predictor: $r_t \sim p_\theta(r_t|h_t, o_t)$
- Continue predictor: $c_t \sim p_\theta(c_t|h_t, o_t)$
- Decoder: $z_t \sim p_\theta(z_t|h_t)$

where z is an observation from an original environment, h is hidden recurrent feedback, o is an abstract observation, a is an action same for both original and abstract space, r is the immediate reward, and c is the continue flag providing information about the final states of the model. While the decoder is necessary only for the learning process and visual demonstrations, the remaining components, starting with the encoder, can serve as an imaginary trajectory generator, providing data to standard reinforcement learning. The core benefit of this world-model representation lies in its ability to efficiently generate low-dimensional data that compresses information from the real environment.

3.2 Deep Learning of Discrete Models

While RSSM provides a reasonable abstraction for various complex tasks, its latent (or abstract) observation space is too huge for model-based reasoning as it has, at least in the DreamerV2 implementation, 32^{32} possible states [17]. Despite this significant compression from the original, infinite-like state space, it remains beyond the practical and theoretical boundaries of formal model-based methods. Furthermore, because it relies on complex recurrent feedback, the state space S and the transition function T of the learned POMDP are difficult to extract.

In this work, we propose a novel Gumbel State Space Model (GSSM) architecture that represents a compact, abstract neuro-symbolic model representation of the training data. The core idea relies on Gumbel Softmax [23] (or Concrete [30]) distribution used for both latent representations and recurrent feedback, providing a more natural framework for learning discrete stochastic representations. Compared to DreamerV3 [18], our aim is to downscale



(a) Input (b) Action *Increment* (c) Action *Decrement* (d) Action *Identity*

Figure 2: Demonstration of a simple MNIST arithmetic with learned GSSM.

the size of the latent space, and, inspired by Hudák et al. [22], replace the recurrent memory with discrete feedback, allowing us to completely represent the underlying abstract POMDP in a finite discrete form.

Architecture. As we show in Figure 1, the model consists of three neural networks. First, the stochastic abstract encoder maps the input observation z_t to a vector of multiple discrete categorical variables representing the distribution over an abstract observation o_t. It resembles the RSSM architecture from [17], but now with the aim of providing a small discrete representation using factored Gumbel Softmax reparametrization rather than biased straight-through estimators. Second, we propose a novel discrete sequence model. Similarly to the encoder, it uses the Gumbel reparametrization to learn a distribution over a new latent observation o_t and a new discrete hidden state h_t. The product of the hidden space H and the observation space O gives us a complete abstract state space S of the learned MDP M. The final part of the GSSM, decoder, maps the discrete observation to a predicted output successor observation z_{t+1}. To represent the continue predictor, i.e., the distribution $c_t \sim p_\theta(c_t|o_t, h_t)$, and the reward predictor $r_t \sim p_\theta(r_t|o_t, h_t)$, we propose the same approach as Hafner et al. [18] using a standard multi-layer perceptron.

Learning. The main benefit of our novel architecture is the compatibility with any standard model-learning techniques. In our preliminary prototype implementation, we largely followed the training procedure described in the section on world model learning of the DreamerV3 algorithm [18], but we aim to further improve the robustness of predictions in the face of uncertain outcomes across a wider range of behaviors.

Finite POMDP Extraction. The crucial benefit of our GSSM representation over RSSMs is the ability to extract a finite POMDP by repeated inference of the GSSM. That is, we enumerate over abstract states $\overline{S} : O \times H$, which is possible given the fully enumerable nature of both the abstract observation space O and the discrete recurrent feedback H, and then repeatedly perform the inference over the dynamics predictor to get a transition function of type $T = \overline{S} \times A \to dist(\overline{S})$. Similarly, we can infer final-state and reward predictors to assign each abstract state its corresponding reward and determine whether it is a final state. It defines a POMDP since, while O is observable to the agent, H is hidden yet influences the transition function.

3.3 Automata Learning

As an alternative to our deep learning approach for discrete model learning, we propose using standard automata learning theory for MDPs, based on the IOAlergia algorithm [31], to learn underlying MDPs that align with the trajectories in the abstract observation space. The idea is to remove the dynamics predictor from Figure 1 and move the latent discrete recurrent feedback $h_t \in H$ and the action $a_t \in A$ from the predictor to both the encoder and the decoder. Compared to the learning of GSSM, we plan to learn the encoder and decoder in a standard auto-encoder manner, aggregating the observation $o_t \in O$ and the observation memory $h_t \in H$ to represent the abstract states $\overline{S} = O \times H$. This would enable us to use a standard automata-learning tool, e.g., AALpy [33], to learn POMDPs with observation space $\overline{S}$ and its own hidden state space $\mathcal{S}$.

4 PRELIMINARY EXPERIMENTAL SETTING

In our preliminary experiments, we implemented an initial prototype of the algorithm described in Section 3.2, for now focusing on fully observable environments. In this section, we provide a brief summary of this experimental setting. Details regarding the outcomes of those experiments are further discussed in Section 5.

4.1 MNIST Discretization

Inspired by autoencoding variational Bayes [25] and vector quantized variational autoencoders [39] (VQ-VAE), which focus on stochastic and discrete representation learning, respectively, we conducted preliminary experiments on learning discrete representations of digits from an MNIST dataset to confirm the ability of multivariate Gumbel softmax reparametrization to learn reasonable representations for visual tasks. In this experiment, we defined three preliminary research questions:

(1) Are the factored Gumbel softmax representations suitable for a standard image benchmark, i.e., can we auto-regress MNIST digits through our GSSM architecture?
(2) Can we learn a simple discrete Markov chain over a sequence of MNIST digits, i.e., emulate a sequence of ordered MNIST images with labels 0, 1,..., 9?
(3) Can we learn a discrete pseudo-MDP emulating a simple arithmetic over MNIST digits with actions identity, increment, and decrement?

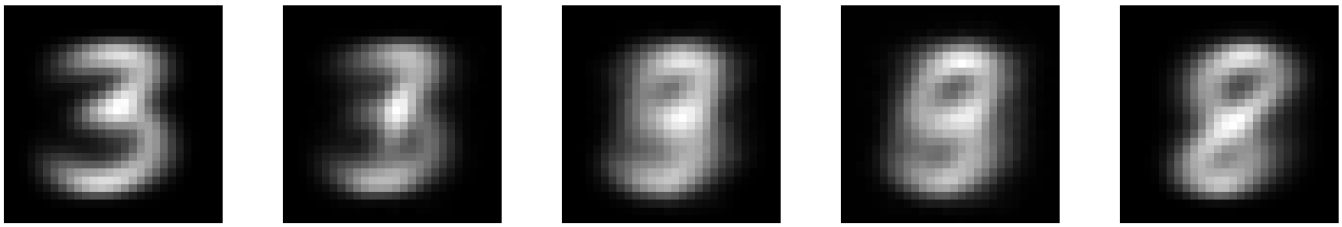


Figure 3: Examples of a decoded latent space to the original MNIST image space. The first figure shows the main candidate used in the MNIST arithmetic experiment, while the others show numbers with a close latent representation. The number continuously changes from a digit 3 to a digit 8.

In those experiments, we included a small perturbation in the labels, i.e., a small probability of incorrect transitions. For example, with a probability of 10 %, we replaced the result of the increment operation with the result of the decrement operation. Furthermore, in our experiments, we limited the discrete space to 256 possible latent states and used a 3-layer multi-layer perceptron (MLP) for the encoder, decoder, and the dynamics predictors. For those experiments, we omitted the reward, the continue predictors, and the recurrent feedback h_t. We demonstrate an example of a test digit not seen during the training over the last research question in Figure 2. We should note that the ground-truth labels are used only to generate the dataset and are completely ignored during the training.

Overall, our experiments showed that the GSSM is suitable for these preliminary problems. However, the number of required states for a minimal Markov model is significantly lower than the state space; specifically, we need precisely 10 discrete states to represent each unique digit for each task, while even our small version of GSSM provides up to 256 unique discrete states for stable learning.

4.2 Prototype MDP Learning

Following our preliminary experiments with the MNIST dataset, in which we demonstrated that factored Gumbel representations can learn a simple discrete latent space with dynamics from visual data, we implemented a prototype planning loop as described in Algorithm 1. To synthesize a policy in an abstract space, we used the Storm [20] model checker, producing a finite, deterministic policy over a learned abstract MDP. The initial exploration policy is implemented as a uniform distribution over the discrete action space, and in subsequent iterations, we explore using the abstract Storm policy with the learned encoder. In our initial experiments with a prototype training loop, we proposed two research questions:

(1) Can we use standard model-checking tools to synthesize policies in an abstract space to get a competitive policy in the original environment?
(2) What are the core challenges of deployment of our GSSM architecture and its training loop?

We implemented two initial benchmarks. We used a standard continuous low-dimensional Gymnasium [38] Cartpole benchmark and a VecStorm simulator from [22] to simulate discrete and finite MDP models defined in the PRISM language [28].

From our first experiment, we found that while Storm produces completely deterministic policies only over an imprecise abstract approximation of the MDP, it can still synthesize policies that are competitive with those learned by a standard PPO algorithm. However, it has two caveats. First, the original DreamerV3 algorithm [18] uses an entropy-oriented actor loss to encourage stochastic exploration of the original environment. As we found, the exploration is a crucial part of the algorithm. Playing deterministic policies can only lead to learning a stable model if the initial Storm policy exhibits sufficiently diverse behavior. To investigate further, we adjusted the exploration process by adding a small amount of noise to the abstract policy, i.e., playing a random action with a small probability, which somewhat improved learning stability, but the method remains unstable and highlights a core challenge in our neuro-symbolic planning loop.

Furthermore, we observed problematic behavior in the learned abstract space across both benchmarks. The Dreamer was designed to optimize over a high-dimensional observation space, where the reconstruction loss plays a significantly more important role than the dynamics. With low-dimensional spaces, we found that dynamics optimization often leads to mode collapse in the abstract space. That is, the algorithm, instead of learning a reasonable abstract representation, overfits to learning the dynamics and learns an encoder that maps to only a few abstract states. This was clearly observed in the experiments with the PRISM models, where the quality of the learned abstraction heavily relies on the original separation of the abstract space. I.e., if only a few abstract states are relevant to the original problem at the start of the algorithm, the learning process does not learn a more diverse state space in later training epochs. While the problem might be partially mitigated by tuning each part of the loss function, in our safe planning setting, it is a core priority to learn a reasonable neuro-symbolic abstract space without relying on such tuning.

5 CHALLENGES AND RESEARCH GOALS

Learning discrete models is an open research problem, and in this section, we outline challenges reported in the literature and those observed in our preliminary experiments.

5.1 Posterior Collapse

As described by Wang et al. [42], one significant challenge in learning compact abstraction with autoencoders is posterior collapse.

It occurs when the learning procedure fails to learn a reasonable discrete space, and instead focuses on optimizing a prior; i.e., the learned model optimizes its prior to generate the most common observation, failing to learn a proper model of the environment. To mitigate this issue, recent methods propose various regularizations and annealing schemes, or design novel optimization formulas, with the aim of avoiding the possibility of a simple solution, as in the case of Wasserstein autoencoders [10]. Thus, one of our main goals is to design model learning to avoid posterior collapse in the partially observable setting, where it can be even more significant, as the model may rely solely on its internal memory rather than on real observations.

5.2 Scalability of the Latent Space

While continuous relaxations of discrete random variables, e.g., the categorical distribution [23, 30] or Bernoulli distribution [41], provide the ability to learn discrete distributions in a deep learning setting, their scalability is limited by the necessity of sampling the specific discrete values to optimize the behavior of neural networks. I.e., the larger the discrete latent space, the more challenging it becomes to explore and optimize the complete state space, leading to a significant portion of states with no clear interpretation.

To demonstrate the issue, we used the preliminary MNIST experiment, in which we learned a simple abstract MDP to model simple arithmetic operations on MNIST digit images using a simplified GSSM implementation. We found that if we enumerate and decode all possible latent representations through the decoder (see some of those in Figure 3), the resulting images are completely clear for only about 10 of them, corresponding to the number of real labels in the dataset. The rest of the latent space ranges from blurry versions of those numbers to a mixture of multiple digits at once. While we did not observe a significant negative impact in our experiments, as the transitions from initial abstractions mostly led to the explored parts of the latent space, even a small probability of reaching unexplored states might be fatal for a formal model-checking approach, leading to unexpected behavior. Furthermore, in this example, we found that limiting the size of the discrete-state space reduces the ability to learn proper digit representations, while larger spaces lead to more uninterpretable latent representations, which adds a novel safety-scalability challenge. We aim to explore existing and propose novel methods to improve the granularity of the state space, e.g., by leveraging recent research on hierarchical discrete representations [37]. Furthermore, scalability could be improved by post-processing learned POMDPs by pruning unlikely states.

5.3 Symmetry of the Latent Space

A general discrete POMDP has only a single discrete state space. In contrast, the world models used in the Dreamer architectures [16–18] rely on a learning-based approach, where the algorithm utilizes the KL divergence to minimize the distance between abstractions of the same encoded observation and its predicted from. However, in our preliminary experiments with a significantly downscaled discrete latent space compared to the ones in Dreamer models, we found that the KL divergence's precision is insufficient, and that the encoder's latent-space ordering differs from the expected ordering at the level of the decoder. I.e., if we generate a sequence starting from an initial observation and multiple selected actions, the imaginary trajectories start diverging. However, we found that using the decoder to generate imaginary outputs in the original space and then feeding them back as inputs to continue the sequence produces the desired behavior. Indeed, this behavior is not ideal, since at the lowest level, we want an independent POMDP abstraction to perform the model checking. Our aim is to redesign the learning method to enforce symmetry in the input-output space.

5.4 Exploration and the Reliability of the Model

While model-based planning methods, led by the recent DreamerV3 [18] approach, are significantly more sample-efficient compared to standard model-free reinforcement learning methods, such as PPO [35], in our safety-oriented setting, it is crucial to precisely represent the complete environment. Furthermore, the model-checking methods might be more sensitive to the quality of the learned model. In this regard, there are various options for collecting data, both in offline and online planning settings. The approach of Wasserstein autoencoder MDPs [10] samples only from the supervising policy combined with ϵ-restarts as proposed in [5]. On the other hand, the first three Dreamer models introduced in DreamerV1 [15] use an iterative optimization over imagined models, followed by sampling from the original environment with a new policy. Lastly, DreamerV4 focuses on a completely offline setting [19] using expert data, previously collected by contractors in the VPT [4] experiment. In contrast to the mentioned approaches, our goal and principal challenge in our setting is to achieve a robust behavior supported by collecting as much diverse knowledge of the environment as possible.

5.5 Non-Uniformity of the Environment

In our preliminary experiments on an MDP setting modeled in a PRISM language [28], i.e., in a setting where we know the model to construct the simulator. Our aim was to reconstruct the abstraction of the original model using only simulations. We observed that the learning has difficulty assigning spiking values to the final-state detection and reward approximation because of their sparsity. Furthermore, the issue scales up because those environments usually have a dynamic action space: in each state, you can play a different set of actions. Surprisingly, as observed by Hudák et al. [22], those models are challenging to solve even with unconstrained deep reinforcement learning methods. On the contrary, existing methods, such as Wasserstein Autoencoder MDPs [10], consider only standard Gymnasium tasks, where the complexity lies in the continuous state space rather than in the unpredictable behavior of the underlying model.

6 CONCLUSION

In this work-in-progress paper, we propose a novel neuro-symbolic planning loop for learning robust, explainable policies in complex partially observable environments with unknown models. Our main planned contribution relies on learning robust, discrete, and finite model representations to enable compatibility with conventional model-based techniques with the main benefit in terms of the explainability of the overall training process and the learned policies.

REFERENCES

[1] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. 2018. Safe Reinforcement Learning via Shielding. *AAAI* (2018).

[2] Roman Andriushchenko, Milan Ceska, Sebastian Junges, and Joost-Pieter Katoen. 2022. Inductive synthesis of finite-state controllers for POMDPs. In *UAI*.

[3] Raphaël Avalos, Florent Delgrange, Ann Nowe, Guillermo Perez, and Diederik M Roijers. 2024. The Wasserstein Believer: Learning Belief Updates for Partially Observable Environments through Reliable Latent Space Models. In *ICLR*.

[4] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video pretraining (VPT): learning to act by watching unlabeled online videos. In *NeurIPS*.

[5] Huang Bojun. 2020. Steady State Analysis of Episodic Reinforcement Learning. In *NeurIPS*.

[6] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. 2019. Large-Scale Study of Curiosity-Driven Learning. In *ICLR*.

[7] Steven Carr, Nils Jansen, and Ufuk Topcu. 2020. Verifiable RNN-Based Policies for POMDPs Under Temporal Logic Constraints. In *IJCAI*.

[8] Steven Carr, Nils Jansen, and Ufuk Topcu. 2021. Task-Aware Verifiable RNN-Based Policies for Partially Observable Markov Decision Processes. *J. Artif. Intell. Res.* (2021).

[9] Krishnendu Chatterjee, Martin Chmelík, Raghav Gupta, and Ayush Kanodia. 2016. Optimal cost almost-sure reachability in POMDPs. *Artificial Intelligence* (2016).

[10] Florent Delgrange, Ann Nowé, and Guillermo A. Pérez. 2023. Wasserstein Auto-encoded MDPs: Formal Verification of Efficiently Distilled RL Policies with Many-sided Guarantees. In *ICLR*.

[11] Maris F. L. Galesloot, Roman Andriushchenko, Milan Češka, Sebastian Junges, and Nils Jansen. 2025. Robust finite-memory policy gradients for hidden-model POMDPs. In *IJCAI*.

[12] Maris F. L. Galesloot, Marnix Suilen, Thiago D. Simão, Steven Carr, Matthijs T. J. Spaan, Ufuk Topcu, and Nils Jansen. 2025. Pessimistic Iterative Planning with RNNs for Robust POMDPs. In *ECAI (Frontiers in Artificial Intelligence and Applications)*. IOS Press, 4823–4831.

[13] Alexander W. Goodall and Francesco Belardinelli. 2023. Approximate Model-Based Shielding for Safe Reinforcement Learning. In *ECAI*.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.. In *ICML*.

[15] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to Control: Learning Behaviors by Latent Imagination. In *ICLR*.

[16] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning Latent Dynamics for Planning from Pixels. In *ICML*. PMLR.

[17] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2021. Mastering Atari with Discrete World Models. In *ICLR*.

[18] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2025. Mastering diverse control tasks through world models. *Nature* (2025).

[19] Danijar Hafner, Wilson Yan, and Timothy Lillicrap. 2025. Training Agents Inside of Scalable World Models. arXiv:2509.24527 [cs.AI]

[20] Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. 2022. The probabilistic model checker Storm. *Int. J. Softw. Tools Technol. Transf.* (2022).

[21] Karel Horák, Branislav Bosanský, and Krishnendu Chatterjee. 2018. Goal-HSVI: Heuristic Search Value Iteration for Goal POMDPs. In *IJCAI*.

[22] David Hudák, Maris F.L. Galesloot, Martin Tappler, Martin Kurečka, Nils Jansen, and Milan Češka. 2026. Finite-State Controllers for (Hidden-Model) POMDPs using Deep Reinforcement Learning. In *AAMAS*.

[23] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.

[24] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* (1998).

[25] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

[26] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. 2022. *Algorithms for Decision Making*. MIT Press.

[27] Hanna Kurniawati. 2022. Partially Observable Markov Decision Processes and Robotics. *Annual Review of Control, Robotics, and Autonomous Systems* (2022).

[28] Marta Kwiatkowska, Gethin Norman, and David Parker. 2009. PRISM: probabilistic model checking for performance and reliability analysis. *SIGMETRICS Perform. Eval. Rev.* (2009).

[29] Omid Madani, Steve Hanks, and Anne Condon. 1999. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI*.

[30] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *ICLR*.

[31] Hua Mao, Yingke Chen, Manfred Jaeger, Thomas D Nielsen, Kim G Larsen, and Brian Nielsen. 2012. Learning Markov decision processes for model checking. *arXiv preprint arXiv:1212.3873* (2012).

[32] Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. 2022. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction* (2022).

[33] Edi Muškardin, Bernhard K. Aichernig, Ingo Pill, Andrea Pferscher, and Martin Tappler. 2022. AALpy: an active automata learning library. *Innovations in Systems and Software Engineering* (2022).

[34] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* (2020).

[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347

[36] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. 2018. Planning and Decision-Making for Autonomous Vehicles. *Annual Review of Control, Robotics, and Autonomous Systems* (2018).

[37] Yuhta Takida, Yukara Ikemiya, Takashi Shibuya, Kazuki Shimada, Woosung Choi, Chieh-Hsin Lai, Naoki Murata, Toshimitsu Uesaka, Kengo Uchida, Wei-Hsiang Liao, and Yuki Mitsufuji. 2024. HQ-VAE: Hierarchical Discrete Representation Learning with Variational Bayes. *Transactions on Machine Learning Research* (2024).

[38] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. 2025. Gymnasium: A Standard Interface for Reinforcement Learning Environments. arXiv:2407.17032 [cs.LG]

[39] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *NeurIPS*.

[40] Alvaro Velasquez, Neel Bhatt, Ufuk Topcu, Zhangyang Wang, Katia Sycara, Simon Stepputtis, Sandeep Neema, and Gautam Vallabha. 2025. Neurosymbolic AI as an antithesis to scaling laws. *PNAS Nexus* (2025).

[41] Xi Wang and Junming Yin. 2020. Relaxed Multivariate Bernoulli Distribution and Its Applications to Deep Generative Models. In *UAI*.

[42] Yixin Wang, David Blei, and John P Cunningham. 2021. Posterior Collapse and Latent Variable Non-identifiability. In *NeurIPS*.

[43] Makoto Yamada, Yuki Takezawa, Ryoma Sato, Han Bao, Zornitsa Kozareva, and Sujith Ravi. 2022. Approximating 1-Wasserstein Distance with Trees. *Transactions on Machine Learning Research* (2022).

[44] Rui Yan, Gabriel Santos, Gethin Norman, David Parker, and Marta Kwiatkowska. 2025. Partially Observable Stochastic Games with Neural Perception Mechanisms. In *FM*.

[45] Yang You, Vincent Thomas, Francis Colas, and Olivier Buffet. 2021. Solving infinite-horizon Dec-POMDPs using finite state controllers within JESP. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 427–434.

[46] Wenqian Zhang and Haiyan Wang. 2022. Diagnostic Policies Optimization for Chronic Diseases Based on POMDP Model. *Healthcare* (2022).