# Towards Zero-Shot Task Transfer with Neurosymbolic World Models

**Isidoro Tamassia, Lennert De Smet, Giuseppe Marra**

KU Leuven, Department of Computer Science, Belgium
{isidoro.tamassia, lennert.desmet, giuseppe.marra}@kuleuven.be



**Abstract**

State-of-the-art model-based reinforcement learning methods learn neural world models that allow policy improvement by planning in a latent space, without assumptions on the structure of the underlying environment. While expressive, these models are generally task-dependent: they learn uninterpretable latent representations that are tied to the training task and thus hard to generalize to new tasks. In this work, we present a novel world model formulation where the reward prediction only depends on a subset of structured, symbolic components of the whole latent state. Decoupling observation reconstruction and reward prediction allows us to learn world models that can adapt zero-shot, i.e. without further environment interactions, to new reward functions defined over the same symbolic state space. We discuss the main advantages and challenges of learning these neurosymbolic world models and demonstrate the strong generalisation properties of our approach over purely neural methods.

## 1 Introduction

World models are motivated by the idea that agents can learn compact internal representations of their environments from experience and use them to guide decision-making (Ha and Schmidhuber 2018). In reinforcement learning (RL), popular methods typically learn latent representations of states, dynamics, and reward functions directly from high-dimensional observations such as sequences of images (Schrittwieser et al. 2020; Hafner et al. 2019, 2020, 2021, 2025; Hansen, Wang, and Su 2022; Hansen, Su, and Wang 2024). These learned representations can be used to optimise behaviour in several ways, including model-predictive control (Hansen, Wang, and Su 2022; Hansen, Su, and Wang 2024), Monte-Carlo Tree Search (Schrittwieser et al. 2020), or actor-critic learning on imagined trajectories (Hafner et al. 2020, 2021, 2025).

Despite their ability to learn compact representations of environment dynamics, RL world models are not straightforwardly reusable across tasks. A particularly important case is *task transfer under shared dynamics*: the environment and its transition structure remain the same, but the reward function changes. This setting arises naturally in domains where the same environment supports multiple objectives, such as reaching different goals, placing objects in different configurations, or avoiding different hazards. Current RL world models are poorly suited to this setting because they typically learn a reward predictor jointly with the latent state and dynamics model. As a result, the learned representations may depend fully (Schrittwieser et al. 2020; Hansen, Wang, and Su 2022; Hansen, Su, and Wang 2024) or partially (Hafner et al. 2020, 2021, 2025) on the rewards observed during training. When the task changes, the learned reward predictor remains tied to the training objective and is therefore not aligned with the new reward function, even though the underlying dynamics are unchanged. At the same time, the latent states learned by these models do not provide an interpretable interface on which a new reward function can be specified directly. Consequently, reusing a task-dependent world model for a new task typically requires learning or finetuning a new reward predictor *using data from the test environment*, or by *relabeling training trajectories* with the downstream task rewards (Sekar et al. 2020).

Unlike transition dynamics, reward functions often encode the desiderata of a user or task designer rather than an intrinsic property of the environment. Hence, they are frequently specified in terms of high-level semantic abstractions of the state, such as an agent reaching a target location, objects satisfying a desired configuration, or unsafe states being avoided (Icarte et al. 2022). When a reward function is given, replacing it with a learned predictor can unnecessarily entangle the task objective with the latent dynamics model.

However, directly using an explicit reward function requires the variables on which it depends to be available to the model. If these symbolic properties were available as part of the model state, the same learned model could be reused for new tasks whose reward functions are defined over the same properties, even when the reward specification itself changes. Crucially, exposing such symbolic properties should not require replacing learned latent dynamics with a fully symbolic model. The symbolic component only needs to capture the reward-relevant aspects of the state, while the remaining information required for control can remain encoded in latent representations learned from raw, subsymbolic observations.

We instantiate this principle with **Neurosymbolic World Models (NeSy-WMs)**, a new class of world models that combine latent dynamics learning with an explicit symbolic interface for reward evaluation. NeSy-WMs build on Recurrent State-Space Models (RSSMs) (Hafner et al. 2019), which learn compact latent dynamics from high-dimensional observations. In addition, a NeSy-WM predicts a set of symbolic state properties that are chosen to capture the variables on

which task rewards depend. Rewards are then predicted only through these symbolic properties, rather than directly from the latent state. This design disentangles latent modeling of environment dynamics from symbolic, interpretable reward prediction. Importantly, the symbolic state does not need to provide a complete symbolic description of the environment. It only needs to expose the reward-relevant properties, leaving the remaining information needed for prediction and control in the learned latent representation.

Our contributions are as follows. **(1)** We formalise NeSy-WMs and their use for task transfer under shared dynamics, where new reward functions are defined over the same symbolic state properties as the training task. **(2)** We propose different symbolic supervision regimes for NeSy-WMs and evaluate their training performance and sample-efficiency against the state-of-the-art DreamerV3 (Hafner et al. 2025). **(3)** We evaluate the zero-shot adaptation of NeSy-WMs to new test-time tasks in two different settings: one where pure imagination planning is performed to solve the task (no further learning), and one where the model is finetuned in imagination. **(4)** We provide an open-source and highly configurable PyTorch (Paszke et al. 2019) implementation of NeSy-WMs.

## 2 Related Work

Our work lies at the intersection of world models for model-based RL (MBRL) and probabilistic neurosymbolic learning.

World models aim to predict, explicitly or implicitly, how an environment (the world) evolves under actions. While reward-free world models are typically trained from observations alone through predictive or generative objectives (Assran et al. 2023; Zhou et al. 2025; Micheli, Alonso, and Fleuret 2023), several task-dependent world models used in MBRL instead aim for optimal control by learning latent dynamics directly through reward prediction, value prediction, and policy improvement objectives (Schrittwieser et al. 2020; Danihelka et al. 2022; Hansen, Wang, and Su 2022; Hansen, Su, and Wang 2024). Generative world models (Ha and Schmidhuber 2018; Alonso et al. 2024; Hafner et al. 2020, 2021, 2025) additionally shape their representations by *reconstructing* observations or sensory inputs. In particular, Recurrent State-Space Models (RSSMs) (Hafner et al. 2019) can successfully model high-dimensional, partially observable environments and form the backbone of the Dreamer family of algorithms (Hafner et al. 2020, 2021, 2025). Moreover, their probabilistic formulation is naturally compatible with probabilistic neurosymbolic methods.

Neurosymbolic AI (Garcez and Lamb 2023; Marra et al. 2024) integrates symbolic knowledge and reasoning into machine learning models. Various probabilistic neurosymbolic models (Manhaeve et al. 2018; Yang, Ishay, and Lee 2020) now integrate generative models to constrain generations and perform test-time interventions. They mostly use variational autoencoders (Kingma and Welling 2014) with a static (Misino, Marra, and Sansone 2022; De Smet et al. 2023) or dynamic (De Smet et al. 2025) encoding space combining latent and symbolic features. Their probabilistic nature is not only used for compatibility with generative models, but also to enable end-to-end training via probabilistic inference. A number of neurosymbolic approaches to world modeling have recently been proposed. Cano et al. (2025) learn reusable neural world-model primitives from offline data and assembles them into a finite-state-machine world model to facilitate control across environments. Sehgal et al. (2024) introduce an object-centric neurosymbolic world model that augments object representations with symbolic attributes to improve compositional generalisation in dynamics prediction. In both cases, the primary goal is to leverage symbolic structure to improve modeling and generalisation of environment dynamics, rather than our goal of enabling adaptation to new tasks under shared dynamics.

Finally, our work is related to goal-conditioned RL (GCRL), which learns policies or value functions conditioned on a goal representation to solve a family of tasks (Schaul et al. 2015; Andrychowicz et al. 2017; Liu, Zhu, and Zhang 2022). GCRL typically amortizes adaptation by *jointly training on many goals* and executing a goal-conditioned policy at test time. In contrast, we learn a NeSy-WM *on a single training task* and perform the desired task adaptation in the imagination of the world model, without the need to pre-train in a multi-task setting.

## 3 Methodology

### 3.1 Overview

Our goal is *task transfer* under two main assumptions. **(A1)** Training and test reward functions are explicitly defined in terms of *a limited number* of high-level abstractions of the state. Such abstractions will be called the symbolic properties of the state *and the vocabulary of such abstractions is assumed to be given by the user*. **(A2)** The underlying dynamics of all tasks is the same. For instance, consider a navigation environment where the training task is to reach a fixed goal. Whether this task has been achieved only depends on the location of the agent. No other properties of the environment, e.g. agent orientation, matter for assessing task-achievement. A possible test task would be to shift the goal to any other position in the map; the reward function still depends only on the agent location, despite inducing a task that is completely different from the training task.

Current world models *learn purely latent representations* of the state which cannot directly exploit user-specified reward functions, even when the test tasks are naturally defined on symbolic properties. As a result, a new reward predictor has to be learned, e.g. by manually relabeling replayed observations (Sekar et al. 2020).

**NeSy-WMs avoid learning new reward functions without harming representation learning.** They instead make the symbolic properties an explicit part of the world model. The symbolic properties form a bottleneck for the reward prediction to facilitate substitution of any other reward function defined on the same properties. More specifically, the model augments DreamerV3 (Hafner et al. 2025) with *neurosymbolic reward and continue predictors* (Figure 1).

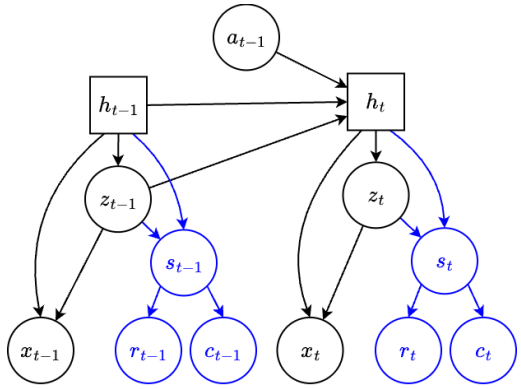


Figure 1: Overview of NeSy-WMs. The symbolic-bottlenecked reward and continue predictors are shown in blue; the rest follows the RSSM structure of (Hafner et al. 2019). At test time, new reward and continue functions can be defined over the same symbols to enable zero-shot planning or imagination training of a new actor-critic.

$$\begin{aligned}
\textbf{Sequence model: } & h_t = f_\phi(z_{t-1}, h_{t-1}, a_{t-1}) \\
\textbf{Encoder: } & z_t \sim q_\phi(z_t \mid h_t, x_t) \\
\textbf{Dynamics predictor: } & \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t) \\
\textbf{NeSy reward predictor: } & \hat{r}_t \sim p_\phi^{\text{sym}}(\hat{r}_t \mid z_t, h_t) \\
\textbf{NeSy continue predictor: } & \hat{c}_t \sim p_\phi^{\text{sym}}(\hat{c}_t \mid z_t, h_t) \\
\textbf{Decoder: } & \hat{x}_t \sim p_\phi(\hat{x}_t \mid z_t, h_t)
\end{aligned}$$

Here, $x_t$ denotes the observation at time $t$, $a_t$ the action, $h_t$ the deterministic recurrent hidden state, $z_t$ the stochastic latent state. The "continue" predictor models the probability that the episode continues at the next step, i.e. that the current state is non-terminal. The function $f_\phi$ is the deterministic recurrent transition of the RSSM. Next, we specify the neurosymbolic predictors and how to compute their probabilities.

**Neurosymbolic predictors combine neural parametrisations with functions on symbols.** The neurosymbolic reward predictor first neurally predicts a distribution $p_\phi(s_t \mid z_t, h_t)$ over the symbolic properties using a simple MLP. The symbolic properties are assumed to be the only input to a given reward function $p^{sym}(r_t \mid s_t)$ (Assumption **(A1)**). The uncertainty on $s_t$ parametrised by $p_\phi(s_t \mid z_t, h_t)$ induces a probability distribution over the rewards $r_t$. Indeed, the distribution $p_\phi^{\text{sym}}(r_t \mid z_t, h_t)$ is given by marginalising over the symbolic states $s_t$ as

$$\sum_{s_t} p^{sym}(r_t \mid s_t) p_\phi(s_t \mid z_t, h_t) \tag{1}$$

$$\text{or} \qquad \int p^{sym}(r_t \mid s_t) p_\phi(s_t \mid z_t, h_t)\, \mathrm{d}s_t, \tag{2}$$

for discrete or continuous symbolic properties, respectively. Equations 1 and 2 also explain the generalisation ability of NeSy-WMs; $p^{sym}(r_t \mid s_t)$ can simply be replaced with a new reward function. The same holds for the neurosymbolic continue predictor, which can be defined analogously.

**Example 3.1** (Discrete Navigation)**.** *Consider a discrete grid world where the symbolic property $s_t = (x_t, y_t)$ models the discrete coordinates of the player at time step $t$. The reward function $p^{sym}(r_t \mid s_t)$ takes the coordinates $s_t$ as input and returns 1 if $s_t$ is the goal $(x_g, y_g)$ and 0 otherwise. If we model the distribution over the symbolic properties $p_\phi(s_t \mid z_t, h_t)$ as two independent categorical distributions, then $p_\phi^{\text{sym}}(r_t = 1 \mid z_t, h_t)$ (Equation 1) reduces to*

$$\Pr_\phi(x_t = x_g \mid z_t, h_t) \cdot \Pr_\phi(y_t = y_g \mid z_t, h_t).$$

**Example 3.2** (Continuous Navigation)**.** *Consider a continuous grid world that has a continuous symbolic state $s_t = (x_t, y_t) \in \mathbb{R} \times \mathbb{R}$. Here, a goal is reached when the symbolic state $s_t$ lies in a square target area $\{(x, y) \mid a_x^- \le x \le a_x^+, a_y^- \le y \le a_y^+, \}$ . If we model the distribution over the symbolic properties $p_\phi(s_t \mid z_t, h_t)$ as two independent Gaussian distributions*

$$\begin{aligned}
x_t \mid z_t, h_t &\sim \mathcal{N}(\mu_x(z_t, h_t), \sigma_x^2(z_t, h_t)), \\
y_t \mid z_t, h_t &\sim \mathcal{N}(\mu_y(z_t, h_t), \sigma_y^2(z_t, h_t)),
\end{aligned}$$

*then $p_\phi^{sym}(r_t = 1 \mid z_t, h_t)$ (Equation 2) is given by*

$$\Pr_\phi(a_x^- < x_t < a_x^+) \cdot \Pr_\phi(a_y^- < y_t < a_y^+),$$

*where $\Pr_\phi(a_x^- < x_t < a_x^+)$ is computable using the Gaussian CDF (supplementary material).*

In the relevant case where the reward function checks whether $s_t$ satisfies logical properties, computing Equations 1 and 2 reduces to the standard inference tasks in probabilistic neurosymbolic AI: Weighted Model Counting (WMC) (Chavira and Darwiche 2008) for discrete $s_t$ or Weighted Model Integration (WMI) (Belle, Passerini, and Van den Broeck 2015) for continuous $s_t$. WMC and WMI have highly-optimised inference strategies (Chavira and Darwiche 2008; Zeng et al. 2020; Maene, Derkinderen, and Zuidberg Dos Martires 2025) to better handle the intractability of computing Equations 1 and 2 (Roth 1996).

## 3.2 Model Learning

Our model-learning setup differs from DreamerV3 in two ways. **(1)** The neurosymbolic predictors $p_\phi^{\text{sym}}(r_t \mid z_t, h_t)$ and $p_\phi^{\text{sym}}(c_t \mid z_t, h_t)$ replace the standard MLPs and **(2)** symbolic supervision losses that can be added to ensure aligned symbolic representations.

**Standard Loss** Given a sequence of observations $x_{1:T}$, actions $a_{1:T}$, rewards $r_{1:T}$ and continuation flags $c_{1:T}$, we minimize the overall world model loss

$$\mathcal{L}_{\text{W}}(\phi) = \mathbb{E}_{q_\phi}\Bigg[\sum_{t=1}^{T}\Big(\beta_{\text{pred}}\mathcal{L}_{\text{pred}}^t(\phi) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}^t(\phi) + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}^t(\phi) + \beta_{\text{rec}}\mathcal{L}_{\text{rec}}^t(\phi)\Big)\Bigg],$$

where $\beta_{\text{pred}}, \beta_{\text{dyn}}, \beta_{\text{rep}}, \beta_{\text{rec}}$ are fixed hyperparameters and the individual losses are defined as

$$\begin{aligned}
\mathcal{L}^t_{\text{rec}}(\phi) &= -\log p_\phi(x_t \mid z_t, h_t), \\
\mathcal{L}^t_{\text{pred}}(\phi) &= -\log p^{\text{sym}}_\phi(r_t \mid z_t, h_t) - \log p^{\text{sym}}_\phi(c_t \mid z_t, h_t), \\
\mathcal{L}^t_{\text{dyn}}(\phi) &= \max\big(1, \text{KL}[\text{sg}(q_\phi(z_t \mid h_t, x_t)) \,||\, p_\phi(z_t \mid h_t)]\big), \\
\mathcal{L}^t_{\text{rep}}(\phi) &= \max(1, \text{KL}[q_\phi(z_t \mid h_t, x_t) \,||\, \text{sg}(p_\phi(z_t \mid h_t))]),
\end{aligned}$$

where $\text{sg}(\cdot)$ is the stop-gradient operator. The dynamics loss $\mathcal{L}^t_{\text{dyn}}(\phi)$ and representation loss $\mathcal{L}^t_{\text{rep}}(\phi)$ optimise the transitions via a KL divergence between prior and posterior. Reconstruction loss $\mathcal{L}^t_{\text{rec}}(\phi)$ and prediction loss $\mathcal{L}^t_{\text{pred}}(\phi)$ ensure accurate observations and reward/continuation signals.

**Neurosymbolic predictors are differentiable.** The neurosymbolic reward and continuation predictors are computed exactly via Equation 1 or 2. Exact neurosymbolic inference remains end-to-end differentiable (Manhaeve et al. 2018) to propagate gradients from $s_t$ to $z_t$ and $h_t$. Consequently, the neurosymbolic $p^{\text{sym}}_\phi(r_t \mid z_t, h_t)$ and $p^{\text{sym}}_\phi(c_t \mid z_t, h_t)$ are drop-in replacements for the MLPs used by DreamerV3.

**Symbolic supervision losses make symbolic states identifiable.** Solely relying on task reward prediction as a source of distant supervision for learning symbolic states can be insufficient to learn aligned representations (Marconato et al. 2023). Identifiability of the symbols impacts task generalisation; if the symbols are learned wrongly, symbolic states may be assigned wrong rewards at test-time. For this reason, we consider and test three different supervision schemes **(S)**.

**(S1) Full supervision.** Direct symbolic supervision on *all* the states visited throughout world model training. This requires the agent to directly observe the symbolic properties $s_{1:T}$. The per-step symbolic supervision loss is the negative log-likelihood of the ground-truth symbol

$$\mathcal{L}^t_{\text{sym}}(\phi) = -\log p_\phi(s_t \mid z_t, h_t).$$

This kind of supervision can be seen as a form of privileged-information training (Lambrechts, Bolland, and Ernst 2024; Huang et al. 2025). However, such approaches either provide the full Markovian state during training, or a surrogate that is sufficient for reconstructing image observations (Lambrechts, Bolland, and Ernst 2024). This is a stronger assumption than observing the symbolic properties considered in our work, which neither constitute a Markovian state representation nor suffice to reconstruct image observations.

**(S2) Partial supervision.** A weaker assumption is to supervise only a subset $\mathcal{S}'$ of all symbolic states. This limits the generalisation of NeSy-WMs to novel tasks involving relevant symbolic states, e.g. potential goal states, in $\mathcal{S}'$. Concretely, partial supervision adds the log-likelihood of the ground-truth symbols only when a state in $\mathcal{S}'$ is visited:

$$\mathcal{L}^t_{\text{sym}}(\phi) = -\mathbb{1}_{\{s_t \in \mathcal{S}'\}} \cdot \log p_\phi(s_t \mid z_t, h_t).$$

Intuitively, this corresponds to having sensors in limited parts of the environment. A potential issue is that the predictor may *collapse* to always predicting supervised symbols; after a reward intervention, a state incorrectly mapped to a rewarded supervised symbol may produce a spurious reward and cause transfer failure. A simple solution inspired by Marconato et al. (2024) is to add an entropy regularisation term to discourage overly confident predictions. For a symbolic state composed of $M$ categorical variables, we minimise

$$\mathcal{L}^t_{\text{ent}}(\phi) = -\beta_{\text{ent}} \frac{1}{M} \sum_{j=1}^{M} \frac{\mathcal{H}\Big[p_\phi(s^{(j)}_t \mid z_t, h_t)\Big]}{\log |\mathcal{S}_j|},$$

where $\mathcal{H}$ denotes Shannon entropy, $\mathcal{S}_j$ is the domain of the $j$-th symbolic variable, and $\beta_{\text{ent}}$ controls the regularisation.

**(S3) No supervision.** When no supervision is provided, the symbolic predictors can learn any mapping as long as it suffices to solve the training task. Nonetheless, we still consider this setting because the imposed symbolic structure may still help stabilise training and facilitate downstream finetuning.

### 3.3 Test-Time Interventions and Generalisation

At training time, control follows the actor-critic scheme of DreamerV3, summarised in the supplementary material. Once a NeSy-WM is trained, it can be adapted to different tasks *without access to the test environments*. To do so, it is sufficient to replace the training predictors $p^{sym}(r_t \mid s_t)$ and $p^{sym}(c_t \mid s_t)$ in Equation 1 or 2 with the given test predictors. This requires new tasks to be defined over the same symbols (Assumption **(A1)**) and assumes fixed dynamics (Assumption **(A2)**). Given the replaced reward function, there are two options **(O)** to adapt a NeSy-WM to a novel test task. In both cases, *there is no interaction with the test environment*.

**(O1) Adaptation by pure planning.** Monte-Carlo Tree Search (MCTS) (Coulom 2006; Kocsis and Szepesvári 2006) can be employed to plan at test time using the learned model and the new reward/continue predictors *without any further learning*. We stress that this is not possible for purely neural models such as Dreamer, since they cannot replace the previously learned reward/continue predictors without a finetuning phase. While we experiment on environments with discrete action spaces, NeSy-WMs can also be used for planning in continuous action spaces using compatible algorithms, such as the Cross-Entropy Method (CEM) (Rubinstein 1997) or appropriate extensions of MCTS (Couetoux 2013).

**(O2) Adaptation by imagination finetuning.** NeSy-WMs can be finetuned on a new task entirely in the world model's imagination *without any further interaction with the environment*. By manually replacing the predictors $p^{\text{sym}}(r_t \mid s_t)$ and $p^{\text{sym}}(c_t \mid s_t)$ as described above, a new actor-critic can be trained without collecting new task-specific data or performing the time-consuming replay-buffer relabeling required by previous Dreamer adaptation approaches (Sekar et al. 2020).

### 3.4 Limitations

The main limitation of our approach surrounds the alignment of the symbolic predictions $p_\phi(s_t \mid z_t, h_t)$ with the ground-truth properties they are meant to represent. If the predicted symbolic distribution is misaligned, then intervening on the reward function can assign rewards according

to the wrong symbolic interpretation, leading to adaptation failure. Symbolic supervision aligns the symbols with the intended semantics and facilitates reliable task transfer, while purely reward-based training may lead to symbolic shortcuts sufficient only for the training task. That is, NeSy-WMs make explicit the relation between the symbolic information provided during learning and the range of reward interventions that can be trusted at test time. Purely neural world models do not provide this possibility because they have no interpretable interface for directly specifying new reward functions.

A separate limitation concerns the availability and sufficiency of the symbolic language. Our formulation assumes that training and test rewards are defined over the same symbolic properties. If a new task depends on properties outside this vocabulary, the latent RSSM state may still contain useful predictive information, but provides no interpretable mechanism for wiring the new reward function on top of it. Transfer is therefore limited to tasks whose rewards can be expressed sufficiently in the chosen symbolic vocabulary.

# 4 Experimental Evaluation

In this section, we propose experiments to test the following three main claims **(C)** of our work.

**(C1) Sample-efficient training.** We expect NeSy-WMs to improve sample efficiency with respect to DreamerV3 due to the provided structure of the reward/continue predictors, especially when some symbolic supervision is provided.

**(C2) Zero-shot adaptation by imagination planning.** We expect NeSy-WMs to enable planning on new test tasks under the same dynamics without further learning, unlike existing finetuning approaches.

**(C3) Zero-shot adaptation by imagination finetuning.** We expect NeSy-WMs to enable zero-shot finetuning on novel test tasks completely in the model imagination. This ability contrasts with existing adaptation settings of DreamerV3, which either require online test-task experience or replay buffer relabeling (Sekar et al. 2020).

To test our claims, we propose a set of corresponding experiments **(E)**:

**(E1) Training sample efficiency.** We train NeSy-WMs in fully supervised (S1), partially supervised (S2), and unsupervised (S3) settings to assess their training sample efficiency compared to standard DreamerV3.

**(E2) Zero-shot adaptation by planning.** The trained NeSy-WMs from E1 are adapted to a variety of test tasks of increasing difficulty *without any further training* through MCTS planning in model imagination (O1). Since Dreamer adaptation requires further training, it cannot be compared in this setting. Instead, we compare against the same MCTS planning algorithm using a perfect environment simulator.

**(E3) Zero-shot adaptation by imagination finetuning.** Using the models trained in E1, NeSy-WMs are adapted to a selected set of hard test tasks through imagination finetuning (O2). Specifically, NeSy-WMs always freeze their latent dynamics model and train a new actor-critic if symbols were fully supervised (S1) or partially supervised (S2) during training (NEW AC). If symbols were unsupervised during training (S3), NeSy-WMs instead learn a new actor-critic and reward/continue predictors through replay buffer relabeling following Sekar et al. (2020). For DreamerV3, we employ the same adaptation scheme by training a new actor-critic and reward/continue predictors from the relabeled buffer (NEW AC + R + C). We also evaluate a baseline where the entire DreamerV3 world model is unfrozen (WM UNFROZEN).

## 4.1 Training Environments

We describe the environments used for the training experiments in E1 (Figure 2), and their symbolic abstractions.

**MiniGrid 2D-FourRooms.** We employ two versions of a 2D Four-Rooms environment implemented using MiniGrid (Chevalier-Boisvert et al. 2023). In particular, MINIGRID-SMALL has four $3 \times 3$ rooms, whereas MINIGRID-LARGE has $5 \times 5$ rooms. The observations are images of the whole grid, and the agent has 3 available actions: FORWARD, TURN-LEFT, and TURN-RIGHT. The agent only receives a positive reward of +1 if the goal is reached and 0 otherwise. At training time, the initial agent position is randomized, and the goal position is fixed as the center of the top-right room. The $(x, y)$ coordinate of the agent is used as the task-relevant symbolic abstraction. This abstraction is sufficient for reward prediction, but incomplete for control since the orientation of the agent also matters for choosing the correct action. In experiments with partial symbolic supervision, we only supervise the center of each room.

**MiniWorld 3D-FourRooms.** We employ two versions of a 3D partially observable four-rooms environment with continuous state space, implemented in MiniWorld (Chevalier-Boisvert et al. 2023). Figure 2 shows a top-down view (the egocentric view is provided in the supplementary material). The action space matches MiniGrid, except that FORWARD moves the agent by $0.3$ and $0.15$ world units in MINIWORLD-SMALL and MINIWORLD-LARGE, respectively, while turning rotates the agent by $30°$ and $15°$. A reward of $+1$ is received only when the agent enters a goal region. During training, the initial position is randomized and the goal is fixed at the center of the top-right room. We use the continuous $(x, y)$ position as the symbolic abstraction. In experiments with partial symbolic supervision, we only supervise the four potential goal areas centered in each room.

**Sokoban.** Here, the agent needs to push all the boxes to the indicated goal positions in a customized version of the game Sokoban (Schrader 2018). In this version, *it only receives a reward ($+1$) when the task is completed.* The agent has 9 available actions: MOVE-DIR and PUSH-DIR for each direction DIR $\in$ {LEFT, RIGHT, UP, DOWN}, and a NULL action which does nothing. We employ a task-relevant symbolic abstraction comprising *only the coordinates of the boxes*, since the reward depends only on their locations. In experiments with partial symbolic supervision, we only provide supervision when all boxes are on training or test targets.

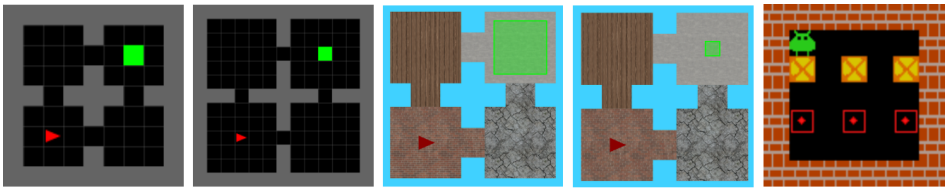

Figure 2: Training tasks. The initial position of the agent is random. From left to right: MiniGrid-small, MiniGrid-large, a top view of MiniWorld-small and MiniWorld-large differing in movement granularity and goal area size, and Sokoban.

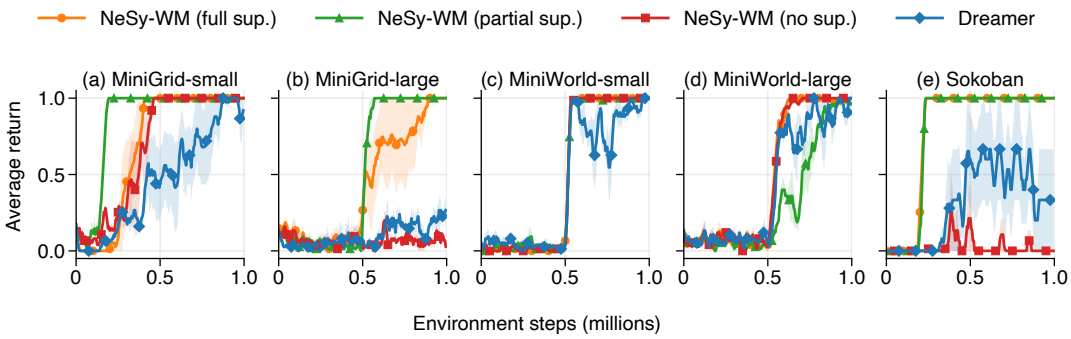


Figure 3: Training results. Curves show the average return across three seeds and their standard error. Both NeSy-WMs and Dreamer start by an exploration phase in the MiniWorld environments and in MiniGrid-large (see supplementary material).

### 4.2 Test Tasks

**Planning Tasks** For the experiments in E2, we select a set of test tasks for each training environment, visualized in the supplementary material. In particular, we take the center of the bottom-left room as test goal, opposite the training goal, for MiniGrid and MiniWorld, and move the boxes to opposite (top) target positions in Sokoban. The tasks are crafted to identify at what point the planning-adapted model fails as the distance between the initial position of the agent and the goal increases. Specifically, the Easy, Medium and Hard challenges for MiniGrid and MiniWorld feature an increasingly distant starting position of the agent from the goal position. Similarly, the Easy task in Sokoban requires placing only one box in a correct test-task configuration, while Hard requires moving all boxes to new target positions.

**Imagination Finetuning Tasks** For the experiments in E3, we consider again the opposite map position with respect to training as the goal for MiniGrid and MiniWorld, and moving all three boxes to opposite target positions in Sokoban. While finetuning the model, we take snapshots of the policy to evaluate it on the test environment with randomized agent initial position at each evaluation episode.

### 4.3 Results

**(C1) NeSy-WMs improve the stability and sample efficiency of world-model training in sparse-reward environments, particularly under symbolic supervision.** Figure 3 shows the training performance of NeSy-WMs under different supervision regimes S1-S3, compared to DreamerV3. Across the five training tasks, NeSy-WMs with either full or partial supervision mostly outperform the baseline in terms of sample-efficiency and stability. The only exception is the partially supervised agent in MiniWorld-large that converged some steps later. Supervised NeSy-WMs also converge significantly faster in MiniGrid-small, and outperform the baseline in MiniGrid-large and Sokoban, where Dreamer is unstable and does not converge. In the case where no symbolic supervision is provided, NeSy-WMs still perform comparably to Dreamer, outperforming it on MiniGrid-small but matching Dreamer's failures on Sokoban and MiniGrid-large. The benefits provided by the supervision in these experiments suggest that symbol prediction may by itself be a strong learning signal for RL world models. We further investigated this hypothesis in an ablation reported in the supplementary material, where we show that our performance under symbolic supervision remains equally strong without reconstruction gradients, unlike Dreamer which dramatically fails.

**(C2) NeSy-WMs enable zero-shot MCTS test-time adaptation, but are constrained by the limitations of uninformed planning.** Table 1 shows the results of the zero-shot MCTS planning with fully supervised and partially supervised NeSy-WMs on the selected test tasks. With the same planning budget of 256 tree expansions per step, the fully supervised model solves most of the Easy and Medium challenges in an optimal or close-to-optimal number of steps. However, pure planning is not enough to consistently solve the Hard tasks in MiniGrid-large, MiniWorld-small/large, and Sokoban. The partially supervised model generally achieves comparable performance, despite lower success rates in MiniGrid-small. These results were compared to planning with perfect environment simulators using the same planning budget and MCTS parameters. Aggre-

| | MG-SMALL | | MG-LARGE | | MW-SMALL | | MW-LARGE | | SOKOBAN | |
|---|---|---|---|---|---|---|---|---|---|---|
| Challenge | Full | Partial | Full | Partial | Full | Partial | Full | Partial | Full | Partial |
| EASY | $1.00 \pm 0.00$<br>$\Delta = 0.0$ | $1.00 \pm 0.00$<br>$\Delta = 2.7$ | $1.00 \pm 0.00$<br>$\Delta = 0.0$ | $1.00 \pm 0.00$<br>$\Delta = 0.0$ | $0.78 \pm 0.15$<br>$\Delta = 8.7$ | $0.89 \pm 0.11$<br>$\Delta = 7.4$ | $0.89 \pm 0.11$<br>$\Delta = 5.1$ | $0.44 \pm 0.18$<br>$\Delta = 24.5$ | $1.00 \pm 0.00$<br>$\Delta = 0.0$ | $1.00 \pm 0.00$<br>$\Delta = 1.1$ |
| MEDIUM | $1.00 \pm 0.00$<br>$\Delta = 0.0$ | $0.67 \pm 0.17$<br>$\Delta = 0.0$ | $1.00 \pm 0.00$<br>$\Delta = 0.0$ | $1.00 \pm 0.00$<br>$\Delta = 8.2$ | $1.00 \pm 0.00$<br>$\Delta = 8.9$ | $1.00 \pm 0.00$<br>$\Delta = 8.2$ | $0.11 \pm 0.11$<br>$\Delta = 33.0$ | $0.00 \pm 0.00$<br>$\Delta = -$ | $1.00 \pm 0.00$<br>$\Delta = 1.8$ | $1.00 \pm 0.00$<br>$\Delta = 0.2$ |
| HARD | $1.00 \pm 0.00$<br>$\Delta = 9.8$ | $0.67 \pm 0.17$<br>$\Delta = 9.0$ | $0.22 \pm 0.15$<br>$\Delta = 33.0$ | $0.44 \pm 0.18$<br>$\Delta = 20.0$ | $0.67 \pm 0.17$<br>$\Delta = 35.3$ | $0.67 \pm 0.17$<br>$\Delta = 16.5$ | $0.00 \pm 0.00$<br>$\Delta = -$ | $0.00 \pm 0.00$<br>$\Delta = -$ | $0.67 \pm 0.17$<br>$\Delta = 72.5$ | $0.67 \pm 0.17$<br>$\Delta = 21.8$ |

Table 1: MCTS evaluation of fully and partially supervised NeSy-WMs. MG and MW denote MiniGrid and MiniWorld, respectively. We report success rate $\pm$ standard error over nine episodes (three episodes for each of three trained models) and the optimality gap $\Delta = L - L^\star$, where $L$ is the mean length of successful plans and $L^\star$ is the optimal plan length for that challenge.

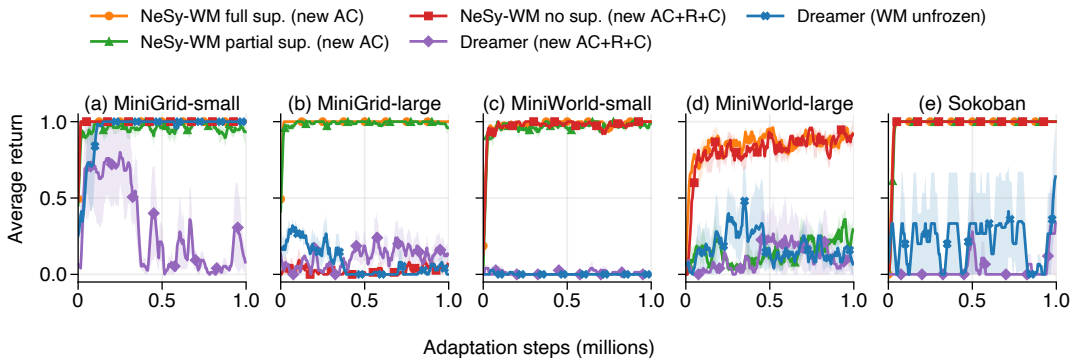


Figure 4: Imagination finetuning results. Curves show the average return across three seeds and their standard error. Adaptation is entirely offline; during finetuning, we periodically evaluate the policy in the test environment solely to measure performance.

gated across all challenges, fully and partially supervised NeSy-WMs achieve success rates of 75.6% and 69.6%, respectively, compared with 79.3% for perfect-simulator planning. The complete comparison is reported in the supplementary material. While increasing the planning budget may improve the results, doing so with a learned model without value bootstrapping is slow (Schrittwieser et al. 2020) and prone to dynamics degradation (Talvitie 2017; Asadi et al. 2019).

**(C3) NeSy-WMs outperform DreamerV3 in the zero-shot imagination finetuning setting.** Figure 4 shows the adaptation performance of NeSy-WMs against the two Dreamer adaptation regimes. We remark that while supervised NeSy-WMs only need to train a fresh actor-critic in imagination *without any replay buffer relabeling*, both Dreamer baselines and unsupervised NeSy-WMs need to train new reward/continue predictors on the relabeled buffer. The results clearly show how Dreamer consistently fails to adapt to the test-time tasks, with the only exception being MINIGRID-SMALL for the WM UNFROZEN regime. Conversely, fully supervised NeSy-WMs adapt almost immediately to all the test-time tasks, followed by partially supervised NeSy-WMs which only struggle in MINIWORLD-LARGE. Unsupervised NeSy-WMs also result in a substantially more reliable adaptation approach than Dreamer, successfully adapting in four out of the five challenges. This shows that even when symbolic supervision is not available, the structure of the neurosymbolic predictors can still make NeSy-WMs a convenient alternative for efficient task-adaptation by imagination finetuning.

## 5 Conclusion

In this work, we presented Neurosymbolic World Models (NeSy-WMs) that build on top of generative world models by incorporating symbolic reward and continuation predictors over learned, task-relevant symbolic properties of the state. NeSy-WMs demonstrate a consistently more sample-efficient training than DreamerV3 on the sparse-reward tasks considered in this work. They can also be employed for zero-shot test-time planning on different tasks than the one they were trained on, i.e. new reward functions defined over the same set of symbolic properties. Finally, NeSy-WMs demonstrated effective zero-shot finetuning on novel tasks in imagination without any interaction with the test environment. Importantly, they consistently outperformed the common approach of finetuning DreamerV3 with a relabeled replay buffer. Overall, we showed how prior symbolic knowledge of the reward function can be incorporated into generative world models in a way that is simple, benefits the training process, and provides distinctive generalisation properties.

Promising directions for extending NeSy-WMs include robust dynamics learning for reliable test-time planning over long horizons, testing other forms of symbolic alignment such as temporal consistency constraints, and exploration strategies that expose the model to the symbolic states needed for reliable task transfer in hard-exploration environments.

## 6 Acknowledgments

This research has received funding from the KU Leuven Research Funds (C14/24/092) and from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

## References

Alonso, E.; Jelley, A.; Micheli, V.; Kanervisto, A.; Storkey, A.; Pearce, T.; and Fleuret, F. 2024. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37: 58757–58791.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, P.; and Zaremba, W. 2017. Hindsight experience replay. *Advances in neural information processing systems*, 30.

Asadi, K.; Misra, D.; Kim, S.; and Littman, M. L. 2019. Combating the compounding-error problem with a multi-step model. *arXiv preprint arXiv:1905.13320*.

Assran, M.; Duval, Q.; Misra, I.; Bojanowski, P.; Vincent, P.; Rabbat, M.; LeCun, Y.; and Ballas, N. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15619–15629.

Belle, V.; Passerini, A.; and Van den Broeck, G. 2015. Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2770–2776. IJCAI Inc.

Cano, L. H.; Perroni-Scharf, M.; Dhir, N.; Ramamurthy, A.; and Solar-Lezama, A. 2025. Neurosymbolic World Models for Sequential Decision Making. In *Forty-second International Conference on Machine Learning*.

Chavira, M.; and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7): 772–799.

Chevalier-Boisvert, M.; Dai, B.; Towers, M.; de Lazcano, R.; Willems, L.; Lahlou, S.; Pal, S.; Castro, P. S.; and Terry, J. 2023. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *CoRR*, abs/2306.13831.

Couetoux, A. 2013. *Monte Carlo tree search for continuous and stochastic sequential decision making problems*. Ph.D. thesis, Université Paris Sud-Paris XI.

Coulom, R. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*, 72–83. Springer.

Danihelka, I.; Guez, A.; Schrittwieser, J.; and Silver, D. 2022. Policy improvement by planning with Gumbel. In *International Conference on Learning Representations*.

De Smet, L.; Dos Martires, P. Z.; Manhaeve, R.; Marra, G.; Kimmig, A.; and De Raedt, L. 2023. Neural probabilistic logic programming in discrete-continuous domains. In *Uncertainty in Artificial Intelligence*, 529–538. PMLR.

De Smet, L.; Venturato, G.; De Raedt, L.; and Marra, G. 2025. Relational neurosymbolic Markov models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 16181–16189.

Garcez, A. d.; and Lamb, L. C. 2023. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11): 12387–12406.

Ha, D.; and Schmidhuber, J. 2018. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31.

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2020. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565. PMLR.

Hafner, D.; Lillicrap, T. P.; Norouzi, M.; and Ba, J. 2021. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2025. Mastering diverse control tasks through world models. *Nature*, 640(8059): 647–653.

Hansen, N.; Su, H.; and Wang, X. 2024. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *The Twelfth International Conference on Learning Representations*.

Hansen, N.; Wang, X.; and Su, H. 2022. Temporal Difference Learning for Model Predictive Control. In *International Conference on Machine Learning, PMLR*.

Huang, D.; Wang, J.; Li, Y.; Xia, C.; Zhang, T.; and Zhang, K. 2025. PIGDreamer: Privileged information guided world models for safe partially observable reinforcement learning. In *Forty-second International Conference on Machine Learning*.

Icarte, R. T.; Klassen, T. Q.; Valenzano, R.; and McIlraith, S. A. 2022. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73: 173–208.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations ICLR*.

Kocsis, L.; and Szepesvári, C. 2006. Bandit Based Monte-Carlo Planning. In *European conference on machine learning*, 282–293. Springer.

Lambrechts, G.; Bolland, A.; and Ernst, D. 2024. Informed POMDP: Leveraging Additional Information in Model-Based RL. *Reinforcement Learning Journal*.

Liu, M.; Zhu, M.; and Zhang, W. 2022. Goal-Conditioned Reinforcement Learning: Problems and Solutions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5502–5511. International Joint Conferences on Artificial Intelligence Organization.

Maene, J.; Derkinderen, V.; and Zuidberg Dos Martires, P. 2025. KLay: Accelerating Arithmetic Circuits for Neurosymbolic AI. In *The Thirteenth International Conference on Learning Representations*.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31.

Marconato, E.; Bortolotti, S.; van Krieken, E.; Vergari, A.; Passerini, A.; and Teso, S. 2024. BEARS Make Neuro-Symbolic Models Aware of their Reasoning Shortcuts. In *Uncertainty in Artificial Intelligence*, 2399–2433. PMLR.

Marconato, E.; Teso, S.; Vergari, A.; and Passerini, A. 2023. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. *Advances in Neural Information Processing Systems*, 36: 72507–72539.

Marra, G.; Dumančić, S.; Manhaeve, R.; and De Raedt, L. 2024. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, 328: 104062.

Micheli, V.; Alonso, E.; and Fleuret, F. 2023. Transformers are Sample-Efficient World Models. In *The Eleventh International Conference on Learning Representations*.

Misino, E.; Marra, G.; and Sansone, E. 2022. Vael: Bridging variational autoencoders and probabilistic logic programming. *Advances in Neural Information Processing Systems*, 35: 4667–4679.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Roth, D. 1996. On the hardness of approximate reasoning. *Artificial intelligence*, 82(1-2): 273–302.

Rubinstein, R. Y. 1997. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1): 89–112.

Schaul, T.; Horgan, D.; Gregor, K.; and Silver, D. 2015. Universal value function approximators. In *International conference on machine learning*, 1312–1320. PMLR.

Schrader, M.-P. B. 2018. gym-sokoban. https://github.com/mpSchrader/gym-sokoban.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Sehgal, A.; Grayeli, A.; Sun, J. J.; and Chaudhuri, S. 2024. Neurosymbolic Grounding for Compositional World Models. In *The Twelfth International Conference on Learning Representations*.

Sekar, R.; Rybkin, O.; Daniilidis, K.; Abbeel, P.; Hafner, D.; and Pathak, D. 2020. Planning to explore via self-supervised world models. In *International conference on machine learning*, 8583–8592. PMLR.

Talvitie, E. 2017. Self-correcting models for model-based reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Yang, Z.; Ishay, A.; and Lee, J. 2020. NeurASP: Embracing neural networks into answer set programming. In *29th International Joint Conference on Artificial Intelligence, IJCAI 2020*, 1755–1762. International Joint Conferences on Artificial Intelligence.

Zeng, Z.; Morettin, P.; Yan, F.; Vergari, A.; and Van den Broeck, G. 2020. Scaling up hybrid probabilistic inference with logical and arithmetic constraints via message passing. In *International Conference on Machine Learning*, 10990–11000. PMLR.

Zhou, G.; Pan, H.; LeCun, Y.; and Pinto, L. 2025. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *International Conference on Machine Learning*, 79115–79135. PMLR.