Injecting Qualitative Spatial Reasoning into World Models via Logic Tensor Networks

Ajay Ramachandra[1] , **Gesina Schwalbe**[2,3] , **Zoe Falomir**[1]
[1]Umeå University
[2]Ulm University
[3]University of Lübeck
{ajay.ramachandra, zoe.falomir}@umu.se, gesina.schwalbe@pheerai.de

Abstract

World models (WM) learn predictive representations from high-dimensional sensory input. However, their internal knowledge is implicit and often inadequate for systematic reasoning about object structure. Qualitative Spatial Reasoning (QSR) models provide explicit human-readable object descriptions under transformations. This paper shows that these two forms of modeling can be combined by injecting a QSR model into a WM to improve the WM reasoning abilities. Using a cube rotation setting inspired by the cube comparison test used in mental rotation tasks as a case study, we use a QSR model that captures the geometric relationships and transformation rules describing how features move across faces under discrete rotations. We translate this QSR model into differentiable constraints using Logic Tensor Networks (LTNs) to guide gradient based learning. These constraints are injected into the WM loss function to regularize its latent dynamics to satisfy the spatial constraints of rotation. The results show that under an extreme data scarce regime (10%), baseline model's performance collapses to near zero, whereas our approach achieves above 90% better score relative to the baseline. Our method consistently maintains high scores across all data regimes (≈80%), for both in-distribution and out-of-distribution observations. This paper proposes a novel framework to infuse QSR knowledge inside WM to make learned latent dynamics logical. Code is available at: *https://github.com/ajayrao80/LtnDreamer/*

1 Introduction

Learning predictive models of the world is key to intelligent agents that must plan, act, and adapt in complex environments. Particularly in robotic manipulation tasks, spatial reasoning is essential to anticipate how objects will move and transform under sequences of actions. These tasks demand prospection, the capacity to mentally simulate physically plausible future states that remain spatially coherent over many steps. World models [Ha and Schmidhuber, 2018] aim to achieve this kind of reasoning by learning latent dynamics directly from other high-dimensional sensory inputs (e.g. pixels). Models such as PlaNet [Hafner *et al.*, 2019b] and Dreamer [Hafner *et al.*, 2019a; Hafner *et al.*, 2020; Hafner *et al.*, 2025] have shown that predictive latent dynamics can be learned (end-to-end) and used for planning and control. Despite their success in a range of model-based reinforcement learning tasks, a known limitation is that small one-step errors accumulate, causing imagined rollouts to drift and degrade over long horizons [Lambert *et al.*, 2022]. This becomes a limiting factor for robotics applications that require precise spatial consistency across long horizons.

Recent benchmark studies show that contemporary vision models also struggle at spatial reasoning tasks such as mental rotation, paper folding and navigation [Stogiannidis *et al.*, 2025; Kamath *et al.*, 2023; Chen *et al.*, 2025]. These findings suggest that explicit structural knowledge is necessary. While recent benchmarks for world models [Li *et al.*, 2025] probe geometric (and rotational) dynamics, there is little prior work that directly evaluates how a Dreamer-style latent world model handles deterministic object rotations.

To address this gap, we investigate whether a qualitative spatial model, which contains explicit (and not necessarily complete), human-readable descriptions of how objects behave under transformations, can be used to regularize the latent dynamics of world models. Specifically, we use an existing qualitative model of object rotations that captures the relational structure of how object features move across 3D locations under discrete rotations [Falomir and Costa, 2025; Falomir, 2025]—providing the structured knowledge that traditional world models lack. We hypothesize that incorporating this spatial knowledge can improve imagination rollouts by limiting the world model's latent dynamics using real spatial constraints regulating 3D object rotations.

Related work. Our approach builds on a growing literature in neuro-symbolic learning that seeks to combine symbolic knowledge with neural networks. In particular, Logic Tensor Networks (LTN) [Badreddine *et al.*, 2022] have shown that logical constraints can be embedded into deep networks for tasks such as deductive reasoning [Bianchi and Hitzler, 2019], image classification [Roychowdhury *et al.*, 2018], reinforcement learning [Amador and Gierasimczuk, 2025; Badreddine and Spranger, 2019], and semantic image interpretation [Shindo *et al.*, 2025; Donadello *et al.*, 2017;

Manigrasso *et al.*, 2023]. More recently, frameworks such as logic constraint injection via straight-through estimators and architectures like LogicMP have shown that first-order logic constraints can guide learning across diverse neural architectures, including MLPs, CNNs, and GNNs [Yang *et al.*, 2023; Xu *et al.*, 2024]. There have been attempts aimed at combining logical reasoning with generative models for design generation [Jacobson and Xue, 2025]. While there have been attempts to combine world models with neuro-symbolic methods [Liang *et al.*, 2025; Balloch *et al.*, 2023; Sehgal *et al.*, 2023] and attempts to overcome drifting imagination in world models [Zhang *et al.*, 2021; Zhu *et al.*, 2023], prior works have not used a differentiable logic framework to guide latent world models to improve multi-step object dynamics. As far as we know, our work is the first to make use of a QSR model encoded with LTN into a world model to improve imagination by reasoning about object transformations.

To achieve this, we lean on existing work on linking latent space encodings with explainable concepts [Lee *et al.*, 2025]. Similar to concept-bottleneck [Koh *et al.*, 2020] and concept embedding models [Espinosa Zarlenga *et al.*, 2022; De Santis *et al.*, 2026], we hard-code which latent dimensions should encode an object's face. However, we do not employ (semi-)supervision [Oikarinen *et al.*, 2022; Yuksekgonul *et al.*, 2022], nor direct contrastive [Esser *et al.*, 2020] or statistical disentanglement [Liu *et al.*, 2022] to guide representation learning. Instead, we derive a novel set of qualitative, contrastive, temporal constraints to infuse abstract rotational reasoning capabilities [van Steenkiste *et al.*, 2019; Kim and Mnih, 2018].

2 Preliminaries

2.1 Qualitative model for reasoning about Object Rotations (QOR)

Qualitative Spatial Representations and Reasoning (QSR) [Cohn and Renz, 2007; Ligozat, 2011] models and reasons about properties of *space* (i.e. topology, location, direction, proximity, geometry, intersection, etc.) and their evolution between continuous neighbouring situations. The Qualitative Object Descriptor (QOD) [Falomir and Costa, 2025] abstract general features of 3D objects embedding them in a bounding box with the shape of a cube which has six faces characterized by features with its corresponding locations and orientations. The model for Qualitative Object Rotations (QOR) [Falomir, 2025] formalizes how the features of an object change their location and orientation depending on the rotation applied. These spatial rules encoded by the QOR model provide the symbolic knowledge structure that is later translated into differentiable constraints via LTNs in this paper.

View and Orientation reference system. The fundamental unit of the QOR model in this paper is the object's observable state from a fixed viewpoint. We adopted the FRONT-RIGHT-UP (FRU) perspective as the object canonical viewpoint (Fig. 1). FRONT, RIGHT and UP are neighbouring locations in the QOR model since their features can move between these locations using only a discrete 90-degree rotation. A $view\ V(O, \text{FRU})$ is defined as a set of triplets, one for each visible side, containing a feature on that side and its orientation. Thus, a descriptor for a view $V(O, \text{FRU})$ includes:

$$V(O, \text{FRU}) = \{(P_f, \textit{front}, O_f), (P_r, \textit{right}, O_r), (P_u, \textit{up}, O_u)\}$$

where P_i is a unique feature currently occupying location i (e.g. P_f is '2' in *front* in the first object in Fig. 1); the 2^{nd} element in the triplet specifies a location part of the selected viewpoint (e.g. *right* in FRU); and O_i is the feature orientation on location i. For example, the first object view in Fig. 1 describes O_1 as:{('2', *front*, 0q), ('6', *right*, 0q), ('1', *up*, 0q)}, that is, O_1 is observed from FRU perspective, and feature '2' is located on the *front* (not rotated), feature '6' is on the *right* (not rotated), and feature '1' is *up* (not rotated).

The QOR model [Falomir, 2025] identifies rotation axes going through the center of each pair of opposite sides in the bounding box of the object, that is, x or *pitch* rotation axis goes from *right* to *left* (rl); y or *yaw* rotation axis goes from *up* to *down* (ud); z or *roll* rotation axis goes from *front* to *back* (fb). The QOR defines discrete 90-degree rotations on these axes associated with an intuitive name and an arrow – see [Falomir, 2025] for details.

State-transitions. The QOR reasoning framework infers the next $view\ V'(O, \text{FRU})$ given the current $view\ V(O, \text{FRU})$ and a rotation. The QOR defines a Conceptual Neighborhood Graph (CNG) relating the location and orientation changes of any feature given a rotation action.

Example 1. *Let us consider the object $View_1$ in Fig. 1 which contains a '2' on the* FRONT*, a '6' on the* RIGHT *and a '1' on the* UP *location. Applying the $\curvearrowright$ rotation (i.e. towards-up-right, that is 90º centred in the front-back axis) results in '6' moving to* DOWN *hidden location, whereas '1' changes location from* UP *to* RIGHT*. The feature '2' remains on the* FRONT *but increments its orientation a quarter. Applying the rotation towards-left ($\leftarrow$) results in '9' remaining* UP *but increasing its orientation by a quarter, '1' changes its location from* RIGHT *to* FRONT *and a new feature '8' appears on the* RIGHT *location.*

The CNG provides the deterministic *ground truth* logic: for any node (Location) and any edge (Rotation Action), the CNG dictates the destination node and the required orientation increment $\{0, +q, -q, +2q\}$ (see [Falomir, 2025]). This knowledge is encoded as LTN constraints to regularize the world model's latent dynamics.

2.2 Baseline world model

As a baseline, we implement a modified Dreamer [Hafner *et al.*, 2019a] world model. The key components of the model used are an (i) *encoder (convolutional)*, a (ii) *recurrent state-space model (RSSM)*, and a (iii) *decoder (convolutional)*. The encoder maps each observation to a latent encoding, which is then processed by the RSSM along with a rotation action, to produce the next stochastic latent state. The decoder reconstructs the next observation from these latent states. Apart from the mentioned modifications, the architecture and training regime follow standard Dreamer practice.

2.3 Logic Tensor Networks

LTNs [Serafini and d'Avila Garcez, 2016; Badreddine *et al.*, 2022] are a neuro-symbolic framework that enables the integration of first-order logical knowledge into neural networks

($\curvearrowright$) ($\leftarrow$)

View$_{T1}(O_1,$ FRU$)$	View$_{T2}(O_1,$ FRU$)$	View$_{T3}(O_1,$ FRU$)$
{(‘2’, *front*, 0q), (‘6’, *right*, 0q), (‘1’, *up*, 0q)}	{(‘2’, *front*, 1q), (‘1’, *right*, 1q), (‘9’, *up*, 0q)}	{(‘1’, *front*, 1q), (‘8’, *right*, 1q), (‘9’, *up*, 1q)}

Figure 1: Views of an object O_1 from the FRU perspective at different observation times (T1, T2, T3) containing handwritten numbers as features. At T1, ‘2’, ‘6’ and ‘1’ are located at *front*, *right* and *up*, respectively. Next views correspond to QODs after rotations *towards-up-right* ($\curvearrowright$) and *towards-left* ($\leftarrow$) in the QOR model.

in an end-to-end differentiable manner. LTNs provide semantics for logical formulas in the language of continuous t-norm fuzzy logic [Novák *et al.*, 1999], in which logical connectives, constants, functions, and predicates are grounded in real valued tensors. The truth value of any formula is a real number in the interval $\in [0, 1]$. This allows logical constraints to be incorporated directly into gradient based training as differentiable loss terms.

Real-valued Logic and Tensor Grounding
Tensor grounding forms the core of LTNs. Every symbol of a logical signature (constants, functions, predicates) is associated with a numeric interpretation. Constants and variables correspond to tensors representing data instances. Functions are implemented as differentiable objects [1] that map input tensors to output tensors. Predicates are implemented as differentiable objects that map tensor(s) to a scalar in $\in [0, 1]$. This grounding enables logical formulas to be evaluated over real data.

Real-valued Connectives and Quantifiers
LTNs implement a fully differentiable logical language called Real Logic. Key components of Real Logic are the following:

- Logical Operators: Standard Boolean operators ($\wedge, \vee, \neg, \implies$) are represented as continuous, differentiable functions derived from a t-norm (real-valued conjunction, $\wedge$; see Appendix for classical t-norms).
- Quantifiers operate over the entire domain, represented as differentiable aggregation functions [2]: (i) the Universal Quantifier ($\forall$) is implemented as $\forall : A_{FA}(a_1, ..., a_n) = 1-(\frac{1}{n}\sum_{i=1}^{n}(1-a_i)^p)^{\frac{1}{p}}$; (ii) the Existential Quantifier $\exists : A_E(a_1, ..., a_n) = \frac{1}{n}\sum_{i=1}^{n}(a_i^p)^{\frac{1}{p}}$; where $a_1, ..., a_n$ are n truth-values in $[0, 1]$ and $p \geq 1$.

Learning in LTN
The spatial knowledge injected into the system is formalized as a set of first order logic constraints. For example, the rotation action $\curvearrowright$ (from Fig. 1) where the symbol at UP moves to RIGHT and symbol at FRONT remains in FRONT but changes orientation can be formulated as the following:

Example 2.

$$\phi_1 : \forall obs_1, obs_2 Up_\theta(obs_1) = OC_\theta(Right_\theta(obs_2)) \quad \textit{and}$$
$$\phi_2 : \forall obs_1, obs_2 Front_\theta(obs_1) = OC_\theta(Front_\theta(obs_2))$$

where $Front_\theta(obs_i)$, $Up_\theta(obs_i)$ are functions with learnable parameters[3] whose outpus are encodings and obs can be thought of as images before and after the rotation. $OC_\theta(e)$ can be thought of as a model that takes in an encoding and gives out another encoding corresponding to incremented orientation of the symbol (see Fig. 1). Optimization is done by aggregating truth-values of all the constraints by a chosen *formula aggregating* operator $SatAgg : [0, 1]^* \rightarrow [0, 1]$ [Badreddine *et al.*, 2022]. We train the model by defining the Logic Loss as:

$$\mathcal{L}_{LTN} = 1 - SatAgg(\phi_1, \phi_2)$$

By minimizing $\mathcal{L}_{\text{LTN}}$ the underlying neural networks are forced to learn parameters that maximize the truth value of the imposed constraints. Thus, the learned representations are forced to adhere to the symbolic rules which guides the model towards logically consistent solutions.

3 Methodology

3.1 Logic Injected Dreamer (Our Approach)

For the logic injected model, we introduce architectural modifications (derived from the QOR [Falomir, 2025]) that expose the spatial structure of the object to be learned. We assume that a single observation consists of multiple semantically meaningful side locations, namely FRONT, RIGHT, and UP, as defined by the QOR. Accordingly, each side location is processed by a dedicated encoder. The decoder is assumed to operate on three location specific latent representations to reconstruct the next predicted observation. We need a dynamics component to learn the mechanics of rotation, given the observation and action. However, Dreamer's dynamics model produces a single global latent state for the next timestep (see [Hafner *et al.*, 2019a]), which is then decomposed by an MLP layer into three location specific encodings prior to decoding. In addition, we need an auxiliary network that can enforce equality between the transformed latents of *side-locations* (See example 2).

Multi-view Encoders and Decoders. Instead of a single encoder, we use *view-specific encoders*. This corresponds to the FRONT, RIGHT, and UP, visible side locations in the observation. Thus, the observation x_t in timestep t is processed by 3 separate encoders ($E^f_\theta, E^r_\theta, E^u_\theta$), producing 3 distinct encodings z^F_t, z^R_t, z^U_t. Each encoder processes the full image and produces a latent representation. This enforced disentanglement between each side location encodings unites ideas from concurrent concept embedding models [Espinosa Zarlenga *et al.*, 2022] for vector-based explainable

[1] Neural networks in practice, but not necessarily.

[2] LTN provides multiple ways to compute quantifiers. The ones provided in the paper are only an example.

[3] LTNs don't require predicates and functions to have learnable parameters, they merely need to be differentiable.

concept encodings, and from classical object detection approaches [He *et al.*, 2017] which use different head for several object shape candidates.

Dynamics Model and Splitter Network. The three encodings are concatenated and passed into the Recurrent State Space Model (RSSM, see [Hafner *et al.*, 2019a] for details) along with the action a_t. The RSSM produces a single global stochastic latent S_{t+1}. The RSSM's stochastic latent state is passed through an additional Splitter Network (MLP) that produces three outputs $\hat{z}^F_{t+1}, \hat{z}^R_{t+1}, \hat{z}^U_{t+1}$ with the same dimensionality as a *location encoding*. The decoder takes these three predicted latents as input, interpreting them as location dependent slots (FRONT, RIGHT, UP) to reconstruct the next observation $\hat{x}_{t+1}$. This factorization allows side location latent representations to be explicitly constrained by the qualitative spatial rules in the QOR model.

Orientation Change Model (RotNet$_\theta$). To capture how the orientation of a feature located on an object side changes given a rotation, a small auxiliary network (MLP) RotNet$_\theta$ is used which takes a latent z and an action a and predicts the orientation change $z' = \text{RotNet}_\theta(z, a)$. It is used exclusively within the LTN constraints and only during training.

3.2 Formalizing QOR as constraints

We encode qualitative spatial knowledge about object rotations (QOR) as LTN constraints. These constraints are expressed primarily as two sets of rules. Namely, encoder consistency rules and decoder consistency rules. Additionally, we add a third set of secondary constraints for the splitter network. Each set characterizes the state change upon each of the rotation actions as it is encoded in the QOR. Together, these 3 rules per action enforce that latent representations under rotation actions obey the qualitative rotation model.

The system is trained with a combined loss. Namely, the standard World Model Loss ($\mathcal{L}_{\text{WM}}$) and a Logic Loss ($\mathcal{L}_{\text{LTN}}$) are derived from the satisfaction of three sets of rules.

The logic to formulate the constraints uses t-norm fuzzy logical connectives, and predicates and functions operate on the domains of time steps T, images X, latent representations Z, and rotations actions A—*towards-left* ($\leftarrow$), *towards-right* ($\rightarrow$), *towards-up* ($\uparrow$), *towards-down* ($\downarrow$), *towards-up-right* ($\curvearrowright$) and *towards-up-left* ($\curvearrowleft$) (see Table 1 by [Falomir, 2025] for more details). Fuzzy predicates are *equalities* on X and Z, which we define as mean squared error on X and as scaled cosine similarity $(z = z') := \frac{1}{2}(\frac{z^T z'}{\|z\| \cdot \|z'\|} + 1) \in [0, 1]$. The required functions are *time increment* $t \mapsto t + 1$, obtaining the *observation* $t \mapsto x_t$ at a time point t, and the *DNN components* of the the front (F), up (U), right (R) face encoders $\text{Enc}_{\theta_\bullet} : x_t \to z^\bullet_t$ for $\bullet \in \{F, U, R\}$, the RotNet$_\theta$, the RSSM and Splitter part $A \times Z^3 \to Z, (z^F_t, z^U_t, z^R_t, a_t) \mapsto \hat{z}^\bullet_{t+1}$ that predicts the next timestep's latents, and lastly the decoder $\text{Dec}_\theta : (\hat{z}^F_{t+1}, \hat{z}^R_{t+1}, \hat{z}^R_{t+1}) \mapsto \hat{x}_{t+1}$.

Encoder Rules. Encoder rules constrain the relationship between the *location encoding* of the current observation and those of the next observation after a rotation action (e.g. $View_{T1}(O_1)$ and $View_{T2}(O_1)$ in Fig. 1). These rules enforce that features move correctly between locations across time steps.

$$(z^F_t = z^R_{t+1}) \wedge (\text{RotNet}_\theta(z^U_t, \rightarrow) = z^U_{t+1}) \quad (1)$$

$$(z^R_t = z^F_{t+1}) \wedge (\text{RotNet}_\theta(z^U_t, \leftarrow) = z^U_{t+1}) \quad (2)$$

$$(z^F_t = z^U_{t+1}) \wedge (\text{RotNet}_\theta(z^R_t, \uparrow) = z^R_{t+1}) \quad (3)$$

$$(z^U_t = z^F_{t+1}) \wedge (\text{RotNet}_\theta(z^R_t, \downarrow) = z^R_{t+1}) \quad (4)$$

$$(\text{RotNet}_\theta(z^F_t, \curvearrowleft) = z^F_{t+1}) \wedge (\text{RotNet}_\theta(z^R_t, \curvearrowleft) = z^U_{t+1}) \quad (5)$$

$$(\text{RotNet}_\theta(z^F_t, \curvearrowright) = z^F_{t+1}) \wedge (\text{RotNet}_\theta(z^U_t, \curvearrowright) = z^R_{t+1}) \quad (6)$$

Decoder Rules. These rules enforce the semantic meaning of the decoder's input slots along with the reconstruction. The decoder must be able to reconstruct x_{t+1} given the correct, logically permuted latents. Note that ? stands for an unknown latent since information about the new feature rotating into view is not available from the x_t.

$$x_{t+1} = \text{Dec}_\theta(?, z^F_t, \text{RotNet}_\theta(z^U_t, \rightarrow)) \quad (7)$$

$$x_{t+1} = \text{Dec}_\theta(z^R_t, ?, \text{RotNet}_\theta(z^U_t, \leftarrow)) \quad (8)$$

$$x_{t+1} = \text{Dec}_\theta(z^U_t, \text{RotNet}_\theta(z^R_t, \downarrow), ?) \quad (9)$$

$$x_{t+1} = \text{Dec}_\theta(?, \text{RotNet}_\theta(z^R_t, \uparrow), z^F_t) \quad (10)$$

$$x_{t+1} = \text{Dec}_\theta(\text{RotNet}_\theta(z^F_t, \curvearrowleft), ?, \text{RotNet}_\theta(z^R_t, \curvearrowleft)) \quad (11)$$

$$x_{t+1} = \text{Dec}_\theta(\text{RotNet}_\theta(z^F_t, \curvearrowright), \text{RotNet}_\theta(z^U_t, \curvearrowright), ?) \quad (12)$$

Splitter Network Alignment Rules. To ensure the WM internalizes these dynamics, we impose a final set of rules on the Splitter Network. These rules take the predicted *location latents* ($\hat{z}^F_{t+1}, \hat{z}^R_{t+1}, \hat{z}^U_{t+1}$) from the Splitter and set them equal to the qualitatively predicted face latents from the observation x_t. These constraints mirror the encoder rules, with the only difference being that the location variables on the right-hand side are taken from the splitter network rather than the *location encoders*.

$$(z^F_t = \hat{z}^R_{t+1}) \wedge (\text{RotNet}_\theta(z^U_t, \rightarrow) = \hat{z}^U_{t+1}) \quad (13)$$

$$(z^R_t = \hat{z}^F_{t+1}) \wedge (\text{RotNet}_\theta(z^U_t, \leftarrow) = \hat{z}^U_{t+1}) \quad (14)$$

$$(z^F_t = \hat{z}^U_{t+1}) \wedge (\text{RotNet}_\theta(z^R_t, \uparrow) = \hat{z}^R_{t+1}) \quad (15)$$

$$(z^U_t = \hat{z}^F_{t+1}) \wedge (\text{RotNet}_\theta(z^R_t, \downarrow) = \hat{z}^R_{t+1}) \quad (16)$$

$$(\text{RotNet}_\theta(z^F_t, \curvearrowleft) = \hat{z}^F_{t+1}) \wedge (\text{RotNet}_\theta(z^R_t, \curvearrowleft) = \hat{z}^U_{t+1}) \quad (17)$$

$$(\text{RotNet}_\theta(z^F_t, \curvearrowright) = \hat{z}^F_{t+1}) \wedge (\text{RotNet}_\theta(z^U_t, \curvearrowright) = \hat{z}^R_{t+1}) \quad (18)$$

In order to satisfy these rules, the Splitter Network is forced to disentangle the global RSSM state into geometrically accurate location representations.

3.3 LTN Loss

The above constraints are converted into a loss regularization term by letting $\mathcal{R}_{\text{enc}}$, $\mathcal{R}_{\text{dec}}$ and $\mathcal{R}_{\text{split}}$ denote the sets of rules for encoder, decoder and splitter network, respectively. Each rule $r \in \mathcal{R}$ is mapped to a satisfaction value $\mu_r \in [0, 1]$ using LTNs. The encoder, decoder and splitter losses are defined as the following:

$$\mathcal{L}_{\text{enc}} = 1 - \underset{r \in \mathcal{R}_{\text{enc}}}{\text{SatAgg}} \left(\mu_r(z^F_t, z^R_t, z^U_t, z^F_{t+1}, z^R_{t+1}, z^U_{t+1}, a_t)\right),$$

$$\mathcal{L}_{\text{dec}} = 1 - \underset{r \in \mathcal{R}_{\text{dec}}}{\text{SatAgg}} \left(\mu_r(z^F_t, z^R_t, z^U_t, x_{t+1}, a_t)\right) \text{ and}$$

Algorithm 1 Logic-Injected Dreamer Training

Require: Observation sequences $x_{1:T}$, action sequences $a_{1:T}$, weights $\lambda_{wm}, \lambda_{ltn}$
Ensure: Trained parameters θ

for each sequence $(x_{1:T}, a_{1:T})$ in minibatch **do**
  for $t = 1$ to $T-1$ **do**
    // 1. Feature Extraction
    $z_t^F, z_t^R, z_t^U \leftarrow E_\theta^f(x_t), E_\theta^r(x_t), E_\theta^u(x_t)$
    $z_{t+1}^F, z_{t+1}^R, z_{t+1}^U \leftarrow E_\theta^f(x_{t+1}), E_\theta^r(x_{t+1}), E_\theta^u(x_{t+1})$
    // 2. Dynamics and Splitter Prediction
    $S_{t+1} \leftarrow \text{RSSM}_\theta\big(S_t, \text{stack}(z_t^F, z_t^R, z_t^U), a_t\big)$
    $\hat{z}_{t+1}^F, \hat{z}_{t+1}^R, \hat{z}_{t+1}^U \leftarrow \text{Splitter}_\theta(S_{t+1})$
    // 3. World Model Loss
    $\mathcal{L}_{WM} \leftarrow \text{Recon}\big(\text{Dec}_\theta(\hat{z}_{t+1}^F, \hat{z}_{t+1}^R, \hat{z}_{t+1}^U), x_{t+1}\big) + \text{KL}(S_{t+1})$
    // 4. LTN Constraint Losses
    $\mathcal{L}_{LTN} \leftarrow \mathcal{L}_{enc} + \mathcal{L}_{dec} + \mathcal{L}_{split}$
    // 5. Parameter Update
    $\theta \leftarrow \theta - \nabla_\theta\Big(\lambda_{wm}\mathcal{L}_{WM} + \lambda_{ltn}\mathcal{L}_{LTN}\Big)$
  end for
end for

$$\mathcal{L}_{\text{split}} = 1 - \underset{r \in \mathcal{R}_{\text{split}}}{\text{SatAgg}}\left(\mu_r(z_t^F, z_t^R, z_t^U, \hat{z}_{t+1}^F, \hat{z}_{t+1}^R, \hat{z}_{t+1}^U, a_t)\right).$$

Then, $\mathcal{L}_{\text{LTN}} = \mathcal{L}_{\text{enc}} + \mathcal{L}_{\text{dec}} + \mathcal{L}_{\text{split}}$, and added onto the world model loss $\mathcal{L}_{\text{WM}}$ becomes the overall loss $\mathcal{L} = \lambda_{\text{WM}}\mathcal{L}_{\text{WM}} + \lambda_{\text{LTN}}\mathcal{L}_{\text{LTN}}$.

The detailed algorithm to conduct the LTN learning of our world model architecture is provided in Alg. 1.

4 Experiments and Evaluation Metrics

Dataset and Environment. We implemented a controlled environment using Unity3D[4] with a single object having MNIST digit [LeCun *et al.*, 2010] pasted onto its faces. As object prototype we used a cube because it can be a generic bounding box for any object. Our object was rendered from a fixed camera viewpoint such that exactly three sides (FRONT, RIGHT UP) were visible at any time (see Fig. 1). The observation at each timestep was an RGB image of size $3 \times 128 \times 128$. The MNIST digits used to construct cubes in the training and evaluation sets are drawn from mutually exclusive sets. Each episode consists of sequences of deterministic cube rotations with a fixed length of 5 steps. The training set contains a total of 5,000 episodes, and the evaluation set contains 1,000 episodes. The action space contains six discrete actions, that is, the QOR 90-degree positive and negative rotations in each of the 3 axes defined (see Table 1 by [Falomir, 2025] for more details).

Experimental Setup. We evaluate the proposed logic injected world model against a vanilla Dreamer baseline in the controlled cube rotation environment.

[4]Unity Technologies. Unity 3D. https://unity.com/releases/editor/whats-new/6000.0.38f1

To study data efficiency, logical consistency and robustness under limited supervision, we train both models using three fractions of the available training set for 1000 epochs. Namely, 100% of the training set (5,000 episodes), 50% of the training set (2,500 episodes), 10% of the training set (500 episodes). For all data regimes, the evaluation set consists of the same 1,000 held-out episodes. Model and training configurations are kept identical across training regimes/models.

Out of Distribution One Shot Generalization. In addition to in-distribution evaluation, we assess the ability of both models to generalize to novel visual content in a one shot setting. For this experiment, we construct a separate object rotation dataset in which object sides are textured with handwritten letters from the EMNIST dataset [Cohen *et al.*, 2017] instead of MNIST digits. This dataset contains 1,000 episodes and has the same environment dynamics, action space, episode length, and observation resolution as the MNIST-based dataset. Crucially, neither model is trained on handwritten letters at any point. The handwritten letter cube dataset is used exclusively for evaluation, only to test whether the learned world models can generalize their learned rotational dynamics to unseen visual features. This setting accounts for generalization of rotational reasoning from memorization of visual appearance.

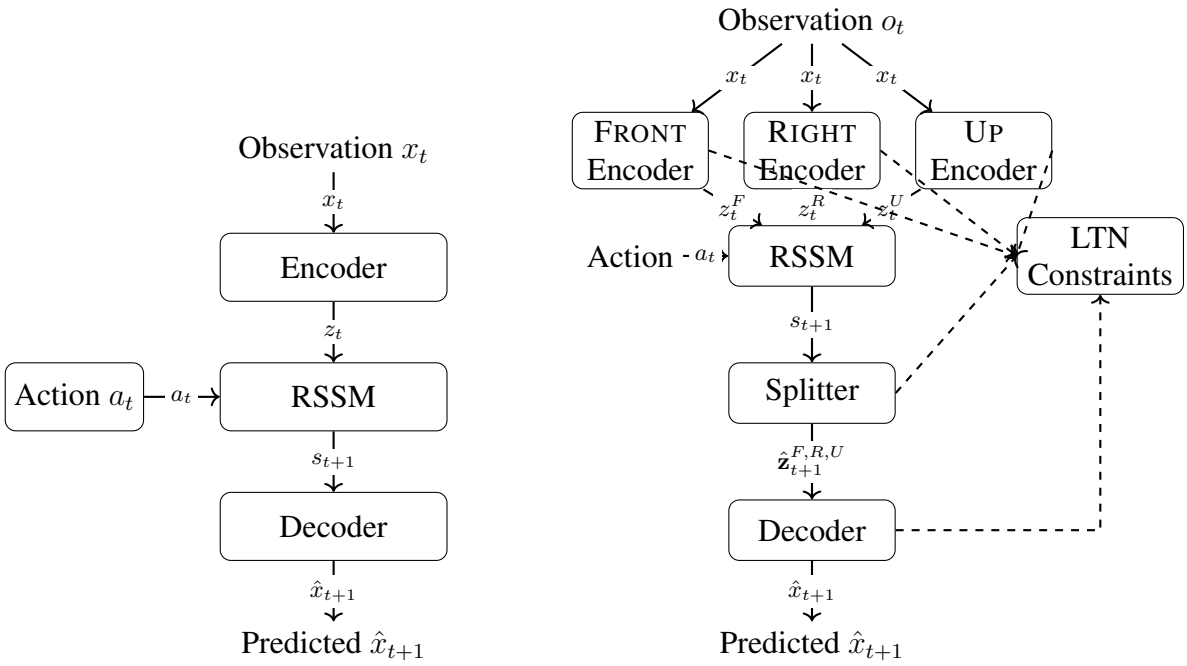
Evaluation Procedure. Our model is evaluated on its ability to imagine future observations under sequences of rotational actions. Given an initial observation, the model performs imagination rollouts by recursively predicting future observations conditioned on the action sequence. Performance is measured over multi-step rollouts, with a specific emphasis on the degradation of imagination over rollout horizon. As the environment dynamics are deterministic, deviations from the *ground-truth* observation at time t reflect failures in rotational reasoning.

Evaluation Metrics. We use three metrics that assess visual accuracy and adherence to qualitative constraints:

M1. Mean Squared Error (MSE) Computed between the predicted observation and the corresponding ground-truth observation at each imagined time step t. MSE provides a baseline measure of pixel-level reconstruction fidelity [Hafner *et al.*, 2019a].

M2. Qualitative Constraint Violation (LTN Loss). To assess adherence to qualitative constraints, we use the standard method of evaluating LTN based models [Badreddine *et al.*, 2022] on imagined rollouts for both the vanilla Dreamer and for our Logic-Dreamer model. For the logic-injected model, this corresponds to the encoder, decoder, and splitter constraint losses used during training. For the vanilla Dreamer model, the same qualitative rules are applied post hoc to its latent representations and reconstructions to measure the degree of constraint violation. The lower the LTN loss, the stronger the adherence to the qualitative constraints.

M3. Adjusted Cosine Similarity (object-side-wise). To measure structural similarity independently of global pixel scale, we compute an adjusted cosine similarity on cropped object location sides. For each predicted and ground-truth observation, we extract image crops corresponding to the visible FRONT, RIGHT and UP locations. Cosine similarity is



(a) Vanilla Dreamer (b) Logic-Dreamer

Figure 2: Vanilla Dreamer vs. our Logic-Dreamer architecture.

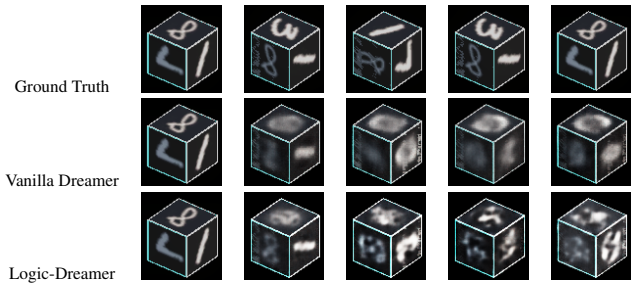


Figure 3: Imagination rollouts for models trained on the full dataset. Top row: ground-truth environment rollout. Middle and bottom rows: imagined rollouts conditioned on the first ground-truth observation. As the rollout progresses, symbols increasingly move outside the agent's field of view, preventing the model from predicting symbols that were not visible in the initial observation.

computed independently for each cropped side and linearly rescaled to lie within the range $[0, 1]$. The final score is obtained by averaging across visible locations. This object-side evaluation isolates spatial reasoning errors, such as incorrect location permutations or rotations. Results for both in-/out-distribution evaluations are reported in the following section.

5 Results

We evaluate vanilla Dreamer and our Logic-Dreamer under both in-distribution and out-of-distribution settings. Performance is measured over a 4-step imagination rollout using MSE, adjusted face-wise cosine similarity, and LTN loss. Results are reported for models trained with 100%, 50%, and 10% of the available training data.

In-Distribution Evaluation. Tab. 1 reports in-distribution imagination rollout performance on objects with MNIST digit features for vanilla Dreamer and our Logic-Dreamer, respectively. Across all training regimes, vanilla Dreamer exhibits stable pixel-level reconstruction error. However, its ability to maintain spatial coherence degrades significantly with fewer training samples. In particular, when trained on only 10% of the data, vanilla Dreamer's similarity score collapses to zero, suggesting a failure to preserve any coherence in its imagination. In contrast, our Logic-Dreamer maintains high similarity scores across multiple rollout steps, even as the dataset gets smaller. At 10% training data, the LTN augmented model achieves relatively high similarity scores, even at the final rollout step. This shows an improvement over the vanilla model (Tab. 1). This effect is particularly pronounced across the rollout steps, where LTN constraints reduce the accumulating drift observed in the baseline. It seems that training with 50% data has a regularizing effect on vanilla Dreamer. MSE and Similarity values are complementary as MSE alone can in extreme cases (as in the 10% vanilla Dreamer) paint a slightly misleading picture. In the 10% case the collapsed imagination (vanilla Dreamer) gives out black images matching with the background resulting in a lower MSE. Face-wise similairty score as result is zero (by definition).

Logical constraint adherence differentiates the two models further. Vanilla Dreamer violates the constraints indicated by high LTN loss across horizons and data regimes. By contrast, our Logic-Dreamer achieves consistently lower LTN loss across horizons. This suggests that the injected LTN constraints are better satisfied throughout the imagination rollout.

Metric	Horizon	100%		50%		10%	
		vanilla Dreamer	Logic-Dreamer(ours)	vanilla Dreamer	Logic-Dreamer(ours)	vanilla Dreamer	Logic-Dreamer(ours)
MSE ↓	t=1	0.216 ± 0.001	$\mathbf{0.004 \pm 0.001}$	$\mathbf{0.007 \pm 0.001}$	0.021 ± 0.002	$\mathbf{0.038 \pm 0.003}$	0.061 ± 0.015
	t=2	0.218 ± 0.002	$\mathbf{0.015 \pm 0.002}$	$\mathbf{0.010 \pm 0.002}$	0.013 ± 0.002	0.038 ± 0.003	$\mathbf{0.029 \pm 0.007}$
	t=3	0.218 ± 0.002	$\mathbf{0.016 \pm 0.002}$	$\mathbf{0.011 \pm 0.002}$	0.012 ± 0.002	0.037 ± 0.003	$\mathbf{0.022 \pm 0.004}$
	t=4	0.218 ± 0.002	$\mathbf{0.016 \pm 0.002}$	0.012 ± 0.002	0.012 ± 0.002	0.038 ± 0.004	$\mathbf{0.019 \pm 0.003}$
LTN Loss ↓	t=1	0.865 ± 0.0003	$\mathbf{0.853 \pm 3.91 \times 10^{-6}}$	$\mathbf{0.853 \pm 0.0001}$	0.855 ± 0.0004	$0.895 \pm 3.32 \times 10^{-7}$	$\mathbf{0.859 \pm 0.001}$
	t=2	0.866 ± 0.0003	$\mathbf{0.854 \pm 2.95 \times 10^{-4}}$	0.854 ± 0.0002	0.854 ± 0.0002	$0.895 \pm 2.50 \times 10^{-7}$	$\mathbf{0.855 \pm 0.001}$
	t=3	0.866 ± 0.0003	$\mathbf{0.854 \pm 2.96 \times 10^{-4}}$	0.854 ± 0.0002	0.854 ± 0.0002	$0.895 \pm 2.60 \times 10^{-7}$	$\mathbf{0.855 \pm 0.001}$
	t=4	0.866 ± 0.0003	$\mathbf{0.854 \pm 3.21 \times 10^{-4}}$	0.854 ± 0.0003	0.854 ± 0.0002	$0.895 \pm 2.66 \times 10^{-7}$	$\mathbf{0.854 \pm 0.001}$
Similarity ↑	t=1	0.794 ± 0.010	$\mathbf{0.956 \pm 0.006}$	$\mathbf{0.903 \pm 0.019}$	0.791 ± 0.017	0.0001 ± 0.0004	$\mathbf{0.763 \pm 0.013}$
	t=2	0.777 ± 0.012	$\mathbf{0.843 \pm 0.023}$	$\mathbf{0.865 \pm 0.023}$	0.861 ± 0.023	0.000 ± 0.000	$\mathbf{0.796 \pm 0.017}$
	t=3	0.776 ± 0.011	$\mathbf{0.835 \pm 0.023}$	0.848 ± 0.023	$\mathbf{0.863 \pm 0.020}$	0.000 ± 0.000	$\mathbf{0.814 \pm 0.016}$
	t=4	0.777 ± 0.012	$\mathbf{0.833 \pm 0.024}$	0.839 ± 0.027	$\mathbf{0.865 \pm 0.020}$	0.000 ± 0.000	$\mathbf{0.827 \pm 0.016}$

Table 1: In-distribution imagination rollout for vanilla Dreamer models vs Logic-Dreamer (ours) trained with data fractions 100%, 50%, and 10%. Results are reported as mean ± standard deviation over a 4-step rollout for MSE (↓), adjusted cosine similarity (↑), and LTN loss (↓).

Metric	Horizon	100%		50%		10%	
		vanilla Dreamer	Logic-Dreamer(ours)	vanilla Dreamer	Logic-Dreamer(ours)	vanilla Dreamer	Logic-Dreamer(ours)
MSE ↓	t=1	0.212 ± 0.002	$\mathbf{0.006 \pm 0.002}$	$\mathbf{0.012 \pm 0.002}$	0.021 ± 0.002	$\mathbf{0.044 \pm 0.005}$	0.056 ± 0.015
	t=2	0.213 ± 0.003	$\mathbf{0.017 \pm 0.002}$	$\mathbf{0.013 \pm 0.002}$	0.015 ± 0.002	0.045 ± 0.005	$\mathbf{0.027 \pm 0.006}$
	t=3	0.213 ± 0.003	$\mathbf{0.018 \pm 0.003}$	0.015 ± 0.002	0.015 ± 0.002	0.045 ± 0.005	$\mathbf{0.020 \pm 0.004}$
	t=4	0.213 ± 0.003	$\mathbf{0.018 \pm 0.003}$	0.015 ± 0.002	0.015 ± 0.002	0.045 ± 0.005	$\mathbf{0.017 \pm 0.003}$
LTN Loss ↓	t=1	0.865 ± 0.0004	$\mathbf{0.853 \pm 1.31 \times 10^{-5}}$	$\mathbf{0.854 \pm 0.0003}$	0.855 ± 0.0004	$0.895 \pm 4.4 \times 10^{-7}$	$\mathbf{0.859 \pm 0.002}$
	t=2	0.866 ± 0.0004	$\mathbf{0.854 \pm 3.1 \times 10^{-4}}$	0.854 ± 0.0003	0.854 ± 0.0003	$0.895 \pm 3.3 \times 10^{-7}$	$\mathbf{0.855 \pm 0.001}$
	t=3	0.866 ± 0.0004	$\mathbf{0.854 \pm 3.5 \times 10^{-4}}$	0.854 ± 0.0003	0.854 ± 0.0002	$0.895 \pm 3.5 \times 10^{-7}$	$\mathbf{0.855 \pm 0.001}$
	t=4	0.866 ± 0.0004	$\mathbf{0.854 \pm 3.8 \times 10^{-4}}$	0.854 ± 0.0004	0.854 ± 0.0003	$0.895 \pm 3.6 \times 10^{-7}$	$\mathbf{0.854 \pm 0.0004}$
Similarity ↑	t=1	0.819 ± 0.016	$\mathbf{0.936 \pm 0.013}$	$\mathbf{0.860 \pm 0.024}$	0.801 ± 0.018	0.0001 ± 0.0005	$\mathbf{0.792 \pm 0.020}$
	t=2	0.806 ± 0.022	$\mathbf{0.832 \pm 0.022}$	0.842 ± 0.023	$\mathbf{0.845 \pm 0.025}$	0.000 ± 0.000	$\mathbf{0.821 \pm 0.022}$
	t=3	0.805 ± 0.020	$\mathbf{0.819 \pm 0.026}$	0.828 ± 0.023	$\mathbf{0.850 \pm 0.021}$	0.000 ± 0.000	$\mathbf{0.835 \pm 0.021}$
	t=4	0.807 ± 0.022	$\mathbf{0.820 \pm 0.026}$	0.822 ± 0.024	$\mathbf{0.854 \pm 0.021}$	0.000 ± 0.000	$\mathbf{0.846 \pm 0.020}$

Table 2: Out-of-distribution (handwritten letters) imagination rollout performance of vanilla Dreamer models vs. Logic-Dreamer models trained with data fractions (100%, 50%, 10%). Results are reported as mean ± standard deviation over a 4-step rollout for MSE (↓), adjusted cosine similarity (↑), and LTN constraint violation (↓).

Out-of-Distribution Evaluation. Tab. 2 reports results on the objects containing handwritten letters as features, which are never seen during training by either model. The out of distribution results mirror the trends observed in-distribution. Vanilla Dreamer again shows a collapse in similarity with model trained with 10% of the training data (Tab. 2).

Our Logic-Dreamer generalizes to unseen handwritten letter textures: Even though the model has never seen these symbols during training, it retains an average similarity score across rollout steps of 0.852 ± 0.022 (across 1–4 horizons and 100% training data), as compared to the baseline which is at 0.81 ± 0.02 (Tab. 2). The same is visible in MSE. This suggests that the model has learned the underlying rotational dynamics rather than surface level visual statistics. LTN Loss seems to be sensitive to small changes. We suspect this is due to the *formula-aggregating operator* forcing the values to be between $[0, 1]$ causing small changes on average to have an amplifying effect on similarity. Notably, **constraint adherence remains stable under distribution shift**.

While LTN loss and similarity scores are not affected by distribution shift, MSE sees an increase for both models. The relative change is small for models trained on 100% training data (13% ours, -2% baseline), but under low training data regimes Logic-Dreamer see a decrease across horizons (9%), while the baseline method sees a considerable increase of 16% at 10% training data. This evidences that the LTN rules help generalize beyond the trained domain.We observe a nuanced interaction between architectural inductive bias, LTN constraints, dataset size and complexity. When trained on the full dataset, Logic-Dreamer model outperforms the vanilla Dreamer baseline. The effect of QOR-derived architecture and LTN constraints provide the inductive prior to learn latent dynamics. Even in the most data scarce setting (10%), the advantage persist where the architectural inductive bias alone is able to still show improved scores.

These results suggest that injecting QSR knowledge into a latent world model noticeably improves multi-step imagination, even under data scarcity and distribution shift. While vanilla Dreamer can achieve low reconstruction error, it lacks the structural knowledge for reliable spatial reasoning. In contrast, Logic-Dreamer setup satisfies constraints and generalizes to unseen visual domains, highlighting the benefits of integrating explicit, human-readable models of object dynamics into world models.

6 Discussing Generalization

A key property of the Qualitative Object Descriptor (QOD) is that it is independent of object shape and serves as a structural representation container [Falomir, 2015; Zhao *et al.*, 2021]. Intuitively, this is analogous to a carton box. When the box is

rotated, its internal structure rotates accordingly. This allows the same compact rule set to be naturally generalized to other shapes. That way, we here demonstrated how a fairly compact set of rules already covers dynamics that may describe arbitrarily complex object rotation dynamics.

A key consideration is that exhaustively specifying constraints is impractical. However, our approach doesn't require complete symbolic specification of the underlying dynamics. The induced rules act as complementary constraints that guide the latent dynamics while still allowing the model to learn from data. As demonstrated by our results even a partial set of rules can substantially improve performance. Fortunately, many real-world systems are governed by a compact set of rules, where a limited number of constraints describe rich dynamics. The controlled cube setting is intentionally chosen to isolate the effect of our approach.

While the constraints used in our method are manually specified, they follow a structured and reusable decomposition into three categories. Namely, encoder, decoder and dynamics (splitter) constraints. This is not domain specific and can transfer other settings where latent representations correspond to meaningful components. Notably, our approach does not rely on access to full QOR graph. Due to partially observability, constraints are applied ony to the visible subset of components (FRONT, RIGHT and UP). As implied above, the effectiveness comes from expressiveness rather than the quantity of rules. A small number of constraints can capture rich behaviours, similar in spirit to physics-informed networks. Extending these principles toward broader and more realistic domains is an important avenue for future work.

7 Conclusion

Our results show that QSR knowledge can be used in world models to improve longer horizon imagination and spatial reasoning. By encoding a QOR model [Falomir and Costa, 2025; Falomir, 2025] as differentiable logical constraints using LTNs and injecting these constraints into a world model, we showed that learned latent dynamics can be anchored to object geometry.

The logic injected model maintains spatial coherence over multi-step imagination rollouts, even under severe data scarcity and distribution shift. In contrast, a standard world model exhibits compounding errors. The reduction in constraint violations observed across evaluations shows that qualitative constraints can act as a form of inductive bias for learned dynamics.

Bridging perception and dynamics is fundamental to building robust world models. A key challenge is enabling them to learn latent dynamics that remain consistent under long rollouts. In this work, we explored how grounding latent representations with explicit qualitative constraints can improve imagination consistency. Integrating constraints to ground latent dynamics of world models provides a promising direction for enforcing consistency and improving generalization beyond training distribution.

A Appendix

A.1 t-norm Fuzzy Logic

A listing of classical t-norms (real-valued logical conjunction) and respectively matching logical connectives is provided in Tab. 3 and Tab. 4. These constitute a generating system of all continuous t-norms [Novák *et al.*, 1999].

Logic	$\neg a$	$a \wedge b$	$a \vee b$
Łukasiewicz	$1-a$	$\max(0, a+b-1)$	$\min(1, a+b)$
Goedel/Minimum	$1-a$	$\min(a, b)$	$\max(a, b)$
Product/Goguen	$1-a$	$a \cdot b$	$a+b-a \cdot b$

Table 3: Basic fuzzy connectives for standard t-norm fuzzy logics.

Logic	$a \rightarrow_R b$	$(\neg a) \vee b$
Łukasiewicz	$\min(1, 1-a+b)$	$\min(1, 1-a+b)$
Goedel/Minimum	1 if $a \leq b$ else b	$\max(1-a, b)$
Product/Goguen	$\min(1, \frac{b}{a})$	$1-a+a \cdot b$

Table 4: Residuated implication (R-implication) and strong implication (S-implication) for standard t-norm fuzzy logics.

Acknowledgements

This work has been supported by the Knut and Alice Wallenberg foundation and the Wallenberg AI, Autonomous Systems and Software Program (WASP). The computational resources are provided by the University of Lübeck and National Academic Infrastructure for Supercomputing in Sweden (NAISS).

References

[Amador and Gierasimczuk, 2025] Ivo Amador and Nina Gierasimczuk. Symdqn: Symbolic knowledge and reasoning in neural network-based reinforcement learning. In Leilani H. Gilpin, Eleonora Giunchiglia, Pascal Hitzler, and Emile van Krieken, editors, *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*, volume 284 of *Proceedings of Machine Learning Research*, pages 70–85. PMLR, 08–10 Sep 2025.

[Badreddine and Spranger, 2019] Samy Badreddine and Michael Spranger. Injecting prior knowledge for transfer learning into reinforcement learning algorithms using logic tensor networks. *ArXiv*, abs/1906.06576, 2019.

[Badreddine *et al.*, 2022] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

[Balloch *et al.*, 2023] Jonathan C. Balloch, Zhiyu Lin, Robert William Wright, Xiangyu Peng, Mustafa Hussain, Aarun Srinivas, Julia Kim, and Mark O. Riedl. Neurosymbolic world models for adapting to open world novelty. *ArXiv*, abs/2301.06294, 2023.

[Bianchi and Hitzler, 2019] Federico Bianchi and Pascal Hitzler. On the capabilities of logic tensor networks for deductive reasoning. In *AAAI Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019)*, Stanford University, Palo Alto, California, USA, March 25-27 2019.

[Chen *et al.*, 2025] Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. Why is spatial reasoning hard for VLMs? An attention mechanism perspective on focus areas. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proc. 42nd Int. Conf. on Machine Learning*, volume 267 of *Proc. of Machine Learning Research*, pages 9910–9932. PMLR, 13–19 Jul 2025.

[Cohen *et al.*, 2017] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

[Cohn and Renz, 2007] A. G. Cohn and J. Renz. *Qualitative Spatial Reasoning, Handbook of Knowledge Representation*. Elsevier, Wiley-ISTE, London, 2007.

[De Santis *et al.*, 2026] Francesco De Santis, Gabriele Ciravegna, Philippe Bich, Danilo Giordano, and Tania Cerquitelli. V-CEM: Bridging Performance and Intervenability in Concept-Based Models. In Riccardo Guidotti, Ute Schmid, and Luca Longo, editors, *Explainable Artificial Intelligence*, pages 48–67, Cham, 2026. Springer Nature Switzerland.

[Donadello *et al.*, 2017] Ivan Donadello, Luciano Serafini, and Artur d'Avila Garcez. Logic tensor networks for semantic image interpretation. In *Proc. 26th Int. Joint Conf. Artificial Intelligence, IJCAI-17*, pages 1596–1602, 2017.

[Espinosa Zarlenga *et al.*, 2022] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept Embedding Models: Beyond the Accuracy-Explainability Trade-Off. *Advances in Neural Information Processing Systems*, 35:21400–21413, December 2022.

[Esser *et al.*, 2020] Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proc. 2020 IEEE Conf. Comput. Vision and Pattern Recognition*, pages 9220–9229, Seattle, WA, USA, June 2020. IEEE.

[Falomir and Costa, 2025] Zoe Falomir and Vicent Costa. Comparing qualitative object descriptors using a visual similarity measure. In Vicenç Torra, Yasuo Narukawa, and Josep Domingo-Ferrer, editors, *Modeling Decisions for Artificial Intelligence - 22nd Int. Conf., MDAI 2025, València, Spain, September 15-18, 2025, Proceedings*, volume 15957 of *Lecture Notes in Computer Science*, pages 303–314. Springer, 2025.

[Falomir, 2015] Zoe Falomir. A Qualitative Model for Reasoning about 3D Objects using Depth and Different Perspectives. In T. Lechowski, P. Walega, and M. Zawidzki, editors, *Proc. of 2015 Workshop on Logics for Qualitative Modelling and Reasoning (LQMR@FedCSIS 2015), Lódz, Poland,*, volume 7 of *Annals of Computer Science and Information Systems*, pages 3–11, 2015.

[Falomir, 2025] Zoe Falomir. A qualitative model to reason about object rotations (QOR) applied to solve the cube comparison test (CCT). *ArXiv arXiv.2601.08382*, 2025.

[Ha and Schmidhuber, 2018] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Proc. 32nd Int. Conf. Neural Information Processing Systems*, NIPS'18, page 2455–2467, Red Hook, NY, USA, 2018. Curran Associates Inc.

[Hafner *et al.*, 2019a] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[Hafner *et al.*, 2019b] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Int. Conf. machine learning*, pages 2555–2565. PMLR, 2019.

[Hafner *et al.*, 2020] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. *arXiv preprint arXiv:2010.02193*, 2020.

[Hafner *et al.*, 2025] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

[He *et al.*, 2017] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. 2017 IEEE Int. Conf. Comput. Vision*, pages 2980–2988, October 2017.

[Jacobson and Xue, 2025] Maxwell J. Jacobson and Yexiang Xue. Integrating symbolic reasoning into neural generative models for design generation. *Artificial Intelligence*, 339:104257, 2025.

[Kamath *et al.*, 2023] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. *ArXiv*, abs/2310.19785, 2023.

[Kim and Mnih, 2018] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proc. 2018 Int. Conf. Machine Learning*, pages 2649–2658. PMLR, July 2018.

[Koh *et al.*, 2020] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proc. 2020 Int. Conf. Machine Learning*, pages 5338–5348. PMLR, November 2020.

[Lambert *et al.*, 2022] Nathan Lambert, Kristofer S. J. Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models. *ArXiv*, abs/2203.09637, 2022.

[LeCun *et al.*, 2010] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs*

[Online]. Available: http://yann.lecun.com/exdb/mnist, 2, 2010.

[Lee *et al.*, 2025] Jae Hee Lee, Georgii Mikriukov, Gesina Schwalbe, Stefan Wermter, and Diedrich Wolter. Concept-Based Explanations in Computer Vision: Where Are We and Where Could We Go? In Alessio Del Bue, Cristian Canton, Jordi Pont-Tuset, and Tatiana Tommasi, editors, *Computer Vision – ECCV 2024 Workshops*, pages 266–287, Cham, 2025. Springer Nature Switzerland.

[Li *et al.*, 2025] Xinyi Li, Zaishuo Xia, Weyl Lu, Chenjie Hao, and Yubei Chen. Smallworlds: Assessing dynamics understanding of world models in isolated environments, 2025.

[Liang *et al.*, 2025] Yichao Liang, Nishanth Kumar, Hao Tang, Adrian Weller, Joshua B. Tenenbaum, Tom Silver, João F. Henriques, and Kevin Ellis. Visualpredicator: Learning abstract world models with neuro-symbolic predicates for robot planning, 2025.

[Ligozat, 2011] G. Ligozat. *Qualitative Spatial and Temporal Reasoning*. MIT Press, Wiley-ISTE, London, 2011.

[Liu *et al.*, 2022] Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q. O'Neil, and Sotirios A. Tsaftaris. Learning disentangled representations in the imaging domain. *Medical Image Analysis*, 80:102516, August 2022.

[Manigrasso *et al.*, 2023] Francesco Manigrasso, Lia Morra, and Fabrizio Lamberti. Fuzzy logic visual network (flvn): A neuro-symbolic approach for visual features matching, 2023.

[Novák *et al.*, 1999] Vilém Novák, Irina Perfilieva, and J. Mockor. *Mathematical Principles of Fuzzy Logic*. The Springer Int. Series in Engineering and Computer Science. Springer US, January 1999.

[Oikarinen *et al.*, 2022] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free Concept Bottleneck Models. In *The Eleventh Int. Conf. Learning Representations*, September 2022.

[Roychowdhury *et al.*, 2018] Soumali Roychowdhury, Michelangelo Diligenti, and Marco Gori. Image classification using deep learning and prior knowledge. In *AAAI Workshops*, pages 336–343, 2018.

[Sehgal *et al.*, 2023] Atharva Sehgal, Arya Grayeli, Jennifer J. Sun, and Swarat Chaudhuri. Neurosymbolic grounding for compositional world models. *ArXiv*, abs/2310.12690, 2023.

[Serafini and d'Avila Garcez, 2016] Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge, 2016.

[Shindo *et al.*, 2025] Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, and Kristian Kersting. Learning differentiable logic programs for abstract visual reasoning, 2025.

[Stogiannidis *et al.*, 2025] Ilias Stogiannidis, Steven McDonagh, and Sotirios A. Tsaftaris. Mind the gap: Benchmarking spatial reasoning in vision-language models, 2025.

[van Steenkiste *et al.*, 2019] Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are Disentangled Representations Helpful for Abstract Visual Reasoning? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Xu *et al.*, 2024] Weidi Xu, Jingwei Wang, Lele Xie, Jianshan He, Hongting Zhou, Taifeng Wang, Xiaopei Wan, Jingdong Chen, Chao Qu, and Wei Chu. Logicmp: A neuro-symbolic approach for encoding first-order logic constraints. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *Int. Conf. Representation Learning*, volume 2024, pages 4181–4209, 2024.

[Yang *et al.*, 2023] Zhun Yang, Joohyung Lee, and Chiyoun Park. Injecting logical constraints into neural networks via straight-through estimators, 2023.

[Yuksekgonul *et al.*, 2022] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc Concept Bottleneck Models. In *ICLR 2022 Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, March 2022.

[Zhang *et al.*, 2021] Lunjun Zhang, Ge Yang, and Bradly C Stadie. World model as a graph: Learning latent landmarks for planning. In *Int. Conf. Machine Learning*, pages 12611–12620. PMLR, 2021.

[Zhao *et al.*, 2021] Meihua Zhao, Gang Xiong, MengChu Zhou, Zhen Shen, and Fei-Yue Wang. 3D-RVP: A method for 3D object reconstruction from a single depth view using voxel and point. *Neurocomputing*, 430:94–103, 2021.

[Zhu *et al.*, 2023] Deyao Zhu, Li Erran Li, and Mohamed Elhoseiny. Value memory graph: A graph-structured world model for offline reinforcement learning, 2023.