

Relational Neurosymbolic Markov Models

Lennert De Smet^{*1}, Gabriele Venturato^{*1}, Luc De Raedt^{†1,2}, Giuseppe Marra^{†1}

¹KU Leuven, Belgium

²Örebro University, Sweden

firstname.lastname@kuleuven.be

Abstract

Sequential problems are ubiquitous in AI, such as in reinforcement learning or natural language processing. State-of-the-art deep sequential models, like transformers, excel in these settings but fail to guarantee the satisfaction of constraints necessary for trustworthy deployment. In contrast, neurosymbolic AI (NeSy) provides a sound formalism to enforce constraints in deep probabilistic models but scales exponentially on sequential problems. To overcome these limitations, we introduce relational neurosymbolic Markov models (NeSy-MMs), a new class of end-to-end differentiable sequential models that integrate and provably satisfy relational logical constraints. We propose a strategy for inference and learning that scales on sequential settings, and that combines approximate Bayesian inference, automated reasoning, and gradient estimation. Our experiments show that NeSy-MMs can solve problems beyond the current state-of-the-art in neurosymbolic AI and still provide strong guarantees with respect to desired properties. Moreover, we show that our models are more interpretable and that constraints can be adapted at test time to out-of-distribution scenarios.

Code — <https://github.com/ML-KULeuven/nesy-mm>

Extended version — <https://arxiv.org/abs/2412.13023>

1 Introduction

Markov models are the theoretical foundation for many successful applications of artificial intelligence, such as speech recognition (Juang and Rabiner 1991), meteorological predictions (Khatani and Ghose 2017), games (Schrittwieser et al. 2020), music generation (Austin et al. 2021), sports analytics (Van Roy et al. 2023) and many more (Mor, Garhwal, and Kumar 2020). They are so popular mainly because they naturally factorise a sequential problem into step-wise probability distributions. Such a decomposition leads to better predictions in terms of bias and variance compared to models that do not incorporate the sequential nature of the problem (Bishop 2006).

Neurosymbolic AI (NeSy) has also enjoyed a tremendous increase in attention. Its general goal is to combine the generalisation potential of symbolic, *i.e.* logical, reasoning with

the representational learning prowess of neural networks. This integration can improve interpretability (Ciatto et al. 2024) and provably satisfy logical constraints. For example, to guarantee the safety of an autonomous agent (Yang et al. 2023), to constrain autoregressive language generation (Zhang et al. 2023) or to impose physical modelling into temporal forecasting (Reichstein et al. 2019).

Such a combination already exists in many different flavours, using either fuzzy logic (Badreddine et al. 2022) or probabilistic logic (Manhaeve et al. 2021; Yang, Ishay, and Lee 2020; Huang et al. 2021; De Smet et al. 2023), and either propositional or relational logic (Marra et al. 2024). The probabilistic case is of special interest, as probabilistic NeSy systems provide a sound semantics to handle uncertainty, as well as to tackle generative tasks. The relational case is also of special interest as relational logic is a popular and very expressive representation for representing states in, for instance, databases and planning (Russell and Norvig 2020). Moreover, relational representations facilitate strong generalisation behaviour (Hummel and Holyoak 2003). Unfortunately, existing probabilistic or relational NeSy models can not exploit the sequential decomposition inherent to temporal reasoning tasks, thereby limiting their applicability in complex sequential problems. Therefore, there are still no inference algorithms for such NeSy models that are tailored to scale in sequential settings.

In order to overcome these limitations, we identify four desiderata that a model and its inference algorithm should satisfy. **(D.I)** It must be able to model and exploit relational logical constraints on states and transition functions. It should use relational states as in planning, and ideally it can cope with both continuous and discrete aspects of reality. **(D.II)** It has to exploit sequential dependencies without restricting the modelling power, allowing it to scale further than existing NeSy systems. **(D.III)** It must be properly neurosymbolic, that is, it must support transition functions that are logical, neural or purely probabilistic in nature, or any combination thereof. Moreover, it must be end-to-end differentiable to allow for the optimisation of any neural components of the model. **(D.IV)** It can tackle both discriminative and generative tasks in a probabilistic fashion.

Both existing probabilistic techniques and neurosymbolic AI are insufficient. On the neurosymbolic side, scalability **(D.II)** remains the biggest problem, and genera-

^{*}Equal first authorship

[†]Equal last authorship

tive capabilities (D.IV) are also lacking. Purely exact techniques (Manhaeve et al. 2021; Yang, Ishay, and Lee 2020) do not scale to non-trivial time horizons, while approximate techniques (Huang et al. 2021; van Krieken et al. 2024; de Penning et al. 2011; Tran 2021; Tran and Garcez 2023) are still limited, *e.g.* they are statistically biased, lack guarantees, or do not support generative tasks. Only few exact NeSy systems can tackle generative tasks (De Smet et al. 2023; Misino, Marra, and Sansone 2022) and those that can, are very limited in scalability. Moreover, most NeSy systems use neural predicates only for perception. While some NeSy methods for structure learning exist (Towell and Shavlik 1994; Yang, Yang, and Cohen 2017), NeSy-MMs can learn both perception and structure at the same time through parameter learning. On the side of probabilistic models, only desideratum (D.IV) can be fully met. Non-parametric techniques can infer any generic hidden Markov model (Koller and Friedman 2009) and have been applied in the statistical relational setting (Nitti, De Laet, and De Raedt 2016). However, their integration with the neural paradigm is often paired with strong distributional assumptions (Krishnan, Shalit, and Sontag 2017), such as requiring Gaussian densities. In particular, gradient-based optimisation is often difficult for general approximate Bayesian inference methods (Ścibior, Masrani, and Wood 2021; Corenflos et al. 2021; Younis and Sudderth 2023).

To fulfil all desiderata, we introduce *relational neurosymbolic Markov models* (NeSy-MMs), the first integration of deep sequential probabilistic models with NeSy. In particular, (i) we provide a formal definition of NeSy-MMs, (ii) we introduce a new differentiable neurosymbolic particle filter that combines Rao-Blackwellised (Liu et al. 2019) inference and state-of-the-art discrete gradient estimation, (iii) we provide an implementation of such models, and (iv) we introduce two new benchmarks for generative and discriminative learning, and run an extensive experimental analysis on both. The results show that NeSy-MMs satisfy all the desiderata (D.I) - (D.IV).

2 Preliminaries

2.1 Markov Models

Hidden Markov models (HMMs) are sequential probabilistic models for discrete-time Markov processes (Baum and Petrie 1966). Given sequences of *states* $\mathbf{X} = (\mathbf{X}_t)_{t \in \mathbb{N}}$ and *observations* $\mathbf{Z} = (\mathbf{Z}_t)_{t \in \mathbb{N}}$, an HMM factorises the joint probability distribution $p(\mathbf{X}, \mathbf{Z})$ as,

$$p(\mathbf{X}_0)p(\mathbf{Z}_0 | \mathbf{X}_0) \prod_{t \in \mathbb{N}} p(\mathbf{X}_{t+1} | \mathbf{X}_t)p(\mathbf{Z}_{t+1} | \mathbf{X}_{t+1}), \quad (1)$$

where \mathbf{X}_t is a fully latent state (Figure 1a). If \mathbf{X}_t has a known factorisation in the form of a Bayesian network (BN) (Pearl 1988), then the process and its observations encode a *Markovian dynamic Bayesian network* (DBN) (Dean and Kanazawa 1989). Note that \mathbf{X}_t and \mathbf{Z}_t are random vectors that can have both discrete and continuous components. In all that follows, a specific assignment of a random variable or vector will be written in lowercase. For example, $\mathbf{x}_t = (x_{t,1}, \dots, x_{t,D})$ is an assignment of the random vector $\mathbf{X}_t = (X_{t,1}, \dots, X_{t,D})$ of dimension D .

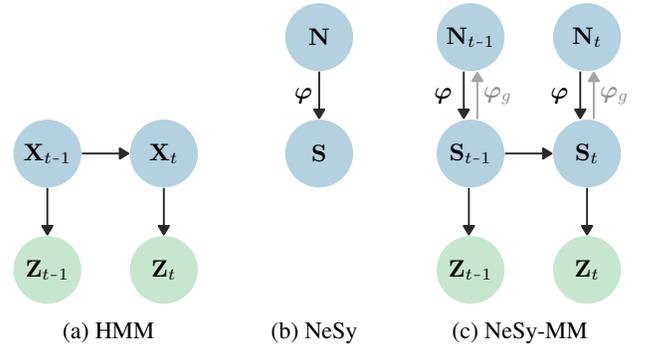


Figure 1: Probabilistic graphical model representations of the different systems considered in this work. Blue represents the states (N and S indicate their *neural* and *symbolic* nature), green the observations.

2.2 Probabilistic Neurosymbolic AI

Probabilistic NeSy methods originate from the field of statistical relational AI (StarAI) that integrates statistical AI with logic (De Raedt et al. 2016; Marra et al. 2024). This integration leads to systems capable of performing inference and learning with uncertainty over *symbolic*, *i.e.* logical, knowledge. For example, the logical relations `player_at(player1, location1)` and `monster_at(monster1, location1)` can be used to apply the rule `hit(P,M) :- player_at(P,L), monster_at(M,L)` and deduce that `player1` is hit by `monster1` because they are in the same location. Moreover, this knowledge is often uncertain in practice, resulting in uncertainty on whether the deduced logical relations hold. For example, consider the case of a sneaky monster. If we are unsure whether `monster_at(monster1, location1)` is true or not, we will also be uncertain whether the player is hit or not. Notice that uncertain logical relations can be modelled as binary random variables.

While StarAI assumes knowledge to be neatly represented as a symbolic state \mathbf{S} , such an assumption does not always hold. Images, sound waves or natural language are usually represented as *subsymbolic* data, *i.e.* tensors, that are not directly usable by relational AI. Therefore, probabilistic NeSy methods use neural predicates φ to map subsymbolic data \mathbf{N} to a probability distribution over a symbolic state \mathbf{S} that can be used by StarAI. Figure 1b depicts a probabilistic graphical model (PGM) (Koller and Friedman 2009) representation of a NeSy system. More formally, given a boolean variable Y from \mathbf{S} with domain $\{y, \neg y\}$ and a set of rules \mathcal{R} on the symbols in \mathbf{S} , inference in NeSy computes the probability that the query Y is true via weighted model integration (WMI) (Morettin et al. 2021)

$$p_\varphi(y | \mathbf{n}) = \int \mathbb{1}_{\mathbf{s} \models_{\mathcal{R}} y} p_\varphi(\mathbf{s} | \mathbf{n}) d\mathbf{s}, \quad (2)$$

where the distribution of \mathbf{S} is parametrised by a neural network φ from the subsymbolic state \mathbf{n} .

A prominent way of representing neurosymbolic models is via probabilistic logic programming (PLP) (De Raedt, Kimmig, and Toivonen 2007).

Algorithm 1 Logic programming encoding of Example 2.1.

```

player(Im, P) ~ normal(noisy_player(Im)).
monster(Im, M) ~ normal(noisy_monster(Im)).
clumsy ~ bernoulli(0.75).

```

```

hits(M, P) :-
  distance(M, P, D), D < 2, not clumsy.
game_over(Im) :-
  player(Im, P), monster(Im, M), hits(M, P).

```

Example 2.1. Consider a simple game where a monster M and player P interact with each other (Figure 1). Both entities are represented by their normally distributed locations, which are parametrised by neural networks from a given image Im . Their interactions are in the form of `hits`, where the monster can hit the player if it is close and not `clumsy`. The clumsiness of the monster is uncertain and modelled by a separate Bernoulli random variable. Finally, the game ends whenever the monster succeeds in hitting the player. Because of the uncertainty on locations and clumsiness, it also follows that whether the game is over is uncertain.

Example 2.1 is encoded in Algorithm 1. The first two lines are *neural predicates* that represent deep random variables modelling the normally distributed locations of the monster M and the player P . Each of the neural predicates has a named neural network that takes the image Im as input and outputs the parameters of its random variable. The third line introduces a Bernoulli random variable `clumsy` indicating that the monster will be clumsy with a 75% chance. The final two lines express two rules of the game that determine when the monster `hits` the player P and when the image Im depicts a lost game, *i.e.* when the image depicts the monster hitting the player.

3 Relational Neurosymbolic Markov Models

Relational neurosymbolic Markov models (NeSy-MMs) combine the sequential and partially observable nature of HMMs and DBNs (Figure 1a) with neurally parametrised relational probability distributions (Figure 1b). That is, we consider Markov processes $\mathbf{X} = (\mathbf{X}_t)_{t \in \mathbb{N}}$ with observations $\mathbf{Z} = (\mathbf{Z}_t)_{t \in \mathbb{N}}$ where the state \mathbf{X}_t is now a *neurosymbolic state* $\mathbf{X}_t = (\mathbf{N}_t, \mathbf{S}_t)$. Figure 1c depicts the graphical model of this novel integration. NeSy-MMs represent joint probability distributions $p_\varphi(\mathbf{N}, \mathbf{S}, \mathbf{Z})$ that factorise as

$$p_\varphi(\mathbf{S}_0 | \mathbf{N}_0)p(\mathbf{N}_0)p(\mathbf{Z}_0 | \mathbf{S}_0) \prod_{t \in \mathbb{N}} p_\varphi(\mathbf{S}_{t+1} | \mathbf{S}_t, \mathbf{N}_{t+1})p(\mathbf{N}_{t+1})p(\mathbf{Z}_{t+1} | \mathbf{S}_{t+1}). \quad (3)$$

Despite the similarity with Eq. 1, NeSy-MMs are complex models that define a wide variety of distributions, taking into account our four desiderate of interest **(D.I)** - **(D.IV)**.

NeSy-MMs explicitly model symbols and their relations. Having a NeSy state means we perform inference in a symbolic state space where *relational logic rules* \mathcal{R} govern the relationship between symbols, both within a single time slice

and in the transition between states. This relational symbolic space allows NeSy-MMs to incorporate human knowledge into our reasoning process, giving guarantees on how the sequential process evolves, *e.g.* we can guarantee safety properties throughout the entire sequence (see Example 3.1). Additionally, the relational aspect significantly enhanced the out-of-distribution generalisation potential (Section 5).

NeSy-MMs factorise symbols over sequences. Standard NeSy systems (Figure 1b) must model the full joint distribution over time, *i.e.* $p(\mathbf{S}) = p(\mathbf{S}_1, \dots, \mathbf{S}_t)$. In contrast, we can factorise the distribution thanks to the Markovian neurosymbolic transition function $p_\varphi(\mathbf{S}_{t+1} | \mathbf{S}_t, \mathbf{N}_{t+1})$, allowing for the definition of probabilistic temporal relations between symbols. Moreover, such a factorisation dramatically simplifies the symbolic space by exploiting the sequential dependencies that standard NeSy systems ignore.

NeSy-MMs integrate neural and logical parametrisations. The symbols of a NeSy-MM and their transitions need not be purely logical and can be parametrised by neural networks. This flexibility in parametrisation not only bridges the gap between subsymbols and symbols, but also allows for neural nets to fill in gaps in background knowledge. For example, when faced with learning the behaviour of another entity in a game while being constrained by the rules of the game (Section 5.2). In essence, NeSy-MMs place symbols and logic where knowledge is available, while using neural nets to parametrise symbols and structure where necessary.

NeSy-MMs express discriminative and generative neurosymbolic models. When given a target variable Y , which can be any of the symbols in \mathbf{S} or a logical derivation thereof, a NeSy-MM can answer conditional *discriminative NeSy queries* of the form $p_\varphi(y | \mathbf{n}, \mathbf{z})$ via

$$\int_{\mathbf{s} \models \mathcal{R}y} p_\varphi(\mathbf{s}_0 | \mathbf{n}_0, \mathbf{z}_0) \prod_{t \in \mathbb{N}} p_\varphi(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{n}_{t+1}, \mathbf{z}_{t+1}) d\mathbf{s}. \quad (4)$$

Alternatively, we can assume a generative perspective (Goodfellow et al. 2020; Dinh, Sohl-Dickstein, and Bengio 2016; Ho, Jain, and Abbeel 2020), *i.e.* the inverted φ_g edges in Figure 1c. This leads to the factorisation

$$\int p_\varphi(\mathbf{s}_0, \mathbf{N}_0 | \mathbf{z}_0) \prod_{t \in \mathbb{N}} p_\varphi(\mathbf{s}_{t+1}, \mathbf{N}_{t+1} | \mathbf{s}_t, \mathbf{z}_{t+1}) d\mathbf{s}, \quad (5)$$

of $p_\varphi(\mathbf{N} | \mathbf{z})$. That is, NeSy-MMs can tackle *generative tasks* where samples \mathbf{n} from $p_\varphi(\mathbf{N} | \mathbf{z})$ that satisfy the possibly logical evidence \mathbf{z} are asked. We showcase this functionality in Section 5.3, where we use a VAE (Kingma and Welling 2013) to generate sequences of images of a game that adhere to the rules of the game.

Example 3.1. Figure 2 shows a new version of the game from Example 2.1, and Algorithm 2 its logic programming encoding. The player can now move in the environment with a Markovian transition function `player_move` based on the player’s previous location and the static monster’s position. The observation rule `safe` guarantees the player’s safety at every time step within the horizon $\Theta : T$. Finally, we can

Algorithm 2 Logic programming encoding of Example 3.1.

```
player(Im, P)0 ~ normal(noisy_player(Im)).  
player(Im, P)t ~ normal(Next) :-  
    player(Im, P)t-1, monster(Im, M),  
    player_move(P, M, Next).  
monster(Im, M) ~ normal(noisy_monster(Im)).  
clumsy ~ bernoulli(0.75).  
  
hits(M, P)t :-  
    distance(M, P, D)t, D < 2, not clumsy.  
  
game_overt(Im) :-  
    player(Im, P)t, monster(Im, M),  
    hits(M, P)t.  
  
safet(Im, P) :-  
    player(Im, P)t, monster(Im, M),  
    distance(M, P, D)t, D > 2.  
  
observe(safe0:T, true).
```

query `game_lost(image.png)t` for any $t \in \{0, \dots, T\}$. Notice that this NeSy-MM depends only on the first image at time $t=0$ and that the projection into the future is done via the logical transition rules.

4 Inference and Learning

To bridge the gap between NeSy and sequential probabilistic models, we propose a new, differentiable inference technique that combines non-parametric approximate Bayesian inference with exact NeSy inference. In the following sections, we will distinguish between random variables with finite and infinite domains. The latter includes both countably infinite and continuous (uncountable) domains.

4.1 Differentiable NeSy-MM Particle Filtering

Traditional particle filters are not differentiable because they perform resampling. Resampling is needed because the observations \mathbf{Z}_t are separated from the transitions $p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t)$, which means the conditional distribution $p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{Z}_{t+1})$ is not readily available. The current state-of-the-art solution is to recover the differentiability of resampling (Ścibior, Masrani, and Wood 2021; Corenflos et al. 2021; Younis and Sudderth 2023). On the contrary, we propose a novel solution that takes advantage of the neurosymbolic nature of a NeSy-MM. In particular, we circumvent the problem of differentiating through resampling by using a Rao-Blackwellised particle filter (RBPF) (Murphy and Russell 2001). A RBPF assumes $p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{Z}_{t+1})$ can be computed exactly and uses it to recursively compute $p_\varphi(\mathbf{X}_{t+1} | \mathbf{Z}_{0:t+1})$ as

$$\int p_\varphi(\mathbf{X}_{t+1} | \mathbf{x}_t, \mathbf{Z}_{t+1}) p_\varphi(\mathbf{x}_t | \mathbf{Z}_{0:t}) d\mathbf{x}_t. \quad (6)$$

We claim it is viable to compute $p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{Z}_{t+1})$ in our NeSy setting because, when \mathbf{X}_t is purely discrete, computing these probabilities can leverage the advances in exact inference from both neurosymbolic AI (Kisa et al. 2014;

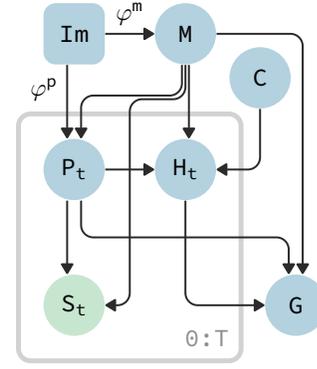


Figure 2: Graphical model of Example 3.1. We use plate notation to indicate a Markov transition.

Tsamoura et al. 2021) and probabilistic AI (Darwiche 2020; Holtzen, Van den Broeck, and Millstein 2020).

By removing resampling and having access to the exact transition probabilities, we can exploit an up-until-now unexplored synergy with gradient estimation. State-of-the-art unbiased discrete gradient estimation algorithms (Kool, van Hoof, and Welling 2019; De Smet, Sansone, and Zuidberg Dos Martires 2023) use samples and the gradients of the probability of those samples to approximate the gradients of finite distributions. In other words, they need the exact probabilities of these distributions to function. Hence, since our RBPF computes $p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{Z}_{t+1})$ exactly, gradient estimation can be used to recursively get approximate gradients for the distribution $p_\varphi(\mathbf{X}_{t+1} | \mathbf{Z}_{0:t+1})$. For example, using the Log-Derivative trick (Williams 1992)

$$\begin{aligned} \nabla_\varphi p_\varphi(\mathbf{X}_{t+1} | \mathbf{Z}_{0:t+1}) & \quad (7) \\ &= \mathbb{E}_{\mathbf{X}_t} [\nabla_\varphi p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{Z}_{t+1})] \\ &+ \mathbb{E}_{\mathbf{X}_t} [p_\varphi(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{Z}_{t+1}) \nabla_\varphi \log p_\varphi(\mathbf{X}_t | \mathbf{Z}_{0:t})]. \end{aligned}$$

In our implementation, we opted for the state-of-the-art performance of RLOO (Kool, van Hoof, and Welling 2019) for gradient estimation.

4.2 NeSy Inference via Cluster Factorisation

Unfortunately, computing $p_\varphi(\mathbf{X}_{t+1} | \mathbf{x}_t, \mathbf{Z}_{t+1})$ exactly when \mathbf{X}_{t+1} also contains variables with an infinite domain is generally only possible under strict assumptions such as Gaussian densities. Moreover, it can still become prohibitively expensive in the purely finite case when ignoring the internal dependency structure of \mathbf{X}_{t+1} . We mitigate these problems by factorising the NeSy-MM further into clusters of variables that become independent when conditioning on \mathbf{Z} . Specifically, a conditional probability distribution $p(\mathbf{X} | \mathbf{Z})$ can be factorised as

$$p(\mathbf{X} | \mathbf{Z}) = \prod_{i=1}^B p(\mathbf{X}^i | \mathbf{Z}), \quad (8)$$

where B is the maximal number of clusters. Intuitively, variables within the same cluster must always be sampled together and hence comprise minimal subproblems to be

solved. The distribution $p(\mathbf{X}^i | \mathbf{Z})$ of each of the subproblems can be computed separately to alleviate the computational bottleneck of computing $p(\mathbf{X} | \mathbf{Z})$ exactly.

Applying the cluster factorisation to the conditional probability distribution $p_\varphi(\mathbf{X}_{t+1} | \mathbf{x}_t, \mathbf{Z}_{t+1})$ with clusters $\{\mathbf{X}_{t+1}^i\}_{i=1}^B$ yields

$$p_\varphi(\mathbf{X}_{t+1} | \mathbf{x}_t, \mathbf{Z}_{t+1}) = \prod_{i=1}^B p_\varphi(\mathbf{X}_{t+1}^i | \mathbf{x}_t, \mathbf{Z}_{t+1}). \quad (9)$$

If we split every cluster \mathbf{X}_t^i into a finite part \mathbf{F}_t^i and infinite part \mathbf{I}_t^i , this factorisation can be further refined into

$$\prod_{i=1}^B p_\varphi(\mathbf{F}_{t+1}^i | \mathbf{I}_{t+1}^i \mathbf{x}_t, \mathbf{Z}_{t+1}) p_\varphi(\mathbf{I}_{t+1}^i | \mathbf{x}_t, \mathbf{Z}_{t+1}). \quad (10)$$

By first obtaining samples for the infinite random variables \mathbf{I}_t^i in every i^{th} cluster using a traditional particle filter, we are again left with a purely finite probability distribution $p_\varphi(\mathbf{F}_{t+1}^i | \mathbf{I}_{t+1}^i \mathbf{x}_t, \mathbf{Z}_{t+1})$ that we compute exactly such that discrete gradient estimation can be applied.

For infinite random variables, where our RBPF-based method cannot be applied, we can recover differentiability using any of the proven and tailored gradient estimation algorithms (Ścibior, Masrani, and Wood 2021; Corenflos et al. 2021; Younis and Sudderth 2023). In our case, we followed the work of Ścibior, Masrani, and Wood (2021) as it provides strong baseline performance. In summary, we recover gradient-based optimisation of infinite, finite and binary logical variables by joining local exact inference with specialised gradient estimation. *The result is a novel Rao-Blackwellised particle filter for NeSy-MMs that handles hybrid domains and exploits the inner conditional dependency structure of the NeSy states \mathbf{X}_t .*

5 Experiments

We present here our generative and discriminative benchmarks, and show that NeSy-MMs are capable of tackling both settings (D.IV). We also clearly show how the presence of relational logic in NeSy-MMs significantly and positively impacts both in- and out-of-distribution performance compared to state-of-the-art deep (probabilistic) models (D.I). In doing so, we show that NeSy-MMs scale to problem settings far beyond the horizon of existing NeSy methods (D.II). In total, NeSy-MMs are successful neurosymbolic models capable of optimising various neural components while adhering to logical constraints (D.III).

5.1 Generative

Our generative experiment is inspired by the Mario experiment of Misino, Marra, and Sansone (2022), extended using MiniHack (Samvelyan et al. 2021), a flexible framework to define environments of the open-ended game NetHack (Küttler et al. 2020). The dataset consists of trajectories of images of length T representing an agent moving T steps in a grid world of size $N \times N$ surrounded by walls. The starting position of the agent is randomly initialised and the actions the agent takes are uniformly sampled among the four cardinal directions, *i.e.* up, down, left, right (Figure 3a). The

actions the agent took at every time step are also given in the trajectory. During training, the model takes sequences of both MiniHack images and actions and learns to reconstruct the given images. At test time, the model should then be able to generate sequences of MiniHack images that follow a given sequence of actions and satisfy the rules of NetHack.

We use VAE (Misino, Marra, and Sansone 2022) as a neurosymbolic baseline and two other fully neural baselines: a variational transformer architecture (VT); and a NeSy-MM without logical rules and with neural networks as transition function (Deep-HMM). Since Deep-HMMs are subsumed by NeSy-MMs, the baseline was implemented in our framework to allow it to benefit from our Rao-Blackwellised inference and learning strategy.

We consider two metrics for the evaluation. First, the reconstruction error (RE), measured by the mean absolute difference in pixel values, which is first averaged over the images, then separately averaged over all images of the sequence. Second, the reconstruction accuracy (RA), which uses a pre-trained classifier for the location of the agent and measures how much the reconstructed trajectory aligns with the ground truth. This is crucial to understand whether the agent is moving according to the actual rules of the game.

5.2 Discriminative

The next setting consists of a discriminative task, where the goal is to classify trajectories of symbolic states. The main challenge is that the transition function is now partially unknown and needs to be learned from examples. That is, we do not use neural networks for perception as is usually done in NeSy, but have a transition that is both neural and logical. More concretely, the dataset for the discriminative task consists of trajectories similar to the generative dataset. However, there are now also enemies present that are trying to kill the agent (Figure 3b). The input to the model in this case is fully symbolic, meaning we do not input images, but rather the precise starting coordinate of the agent and the list of actions performed by the agent. On top of that, we observe if one of the enemies hits the agent, *i.e.* the observations Z_t are binary random variables. The discriminative task is binary classification, where a trajectory has label 1 if the agent dies somewhere in the trajectory and 0 otherwise. While we know the basic rules of NetHack, such as permitted movements and damage mechanics, we do not know the transition function of the enemy. That is, we don't know the behaviour of the enemies and fill this gap in knowledge with a neural network that should respect the known rules of NetHack. We assume that all the enemies share the same behaviour.

Similar to the generative experiment, we use a transformer and a Deep-HMM as baselines, this time in a discriminative configuration. To specifically gauge the out-of-distribution (OOD) generalisation capabilities of all methods, we train only using simple sequences of length 10 containing just one enemy moving on a 10×10 grid and we test on more complex sequences. The OOD cases consider different combinations of sequences on grids of size 10×10 or 15×15 , length 10 or 20, and with 1 or 2 enemies. When more enemies are present or the sequences are longer, it is naturally easier for the agent to be killed. Conversely, the enemies might need

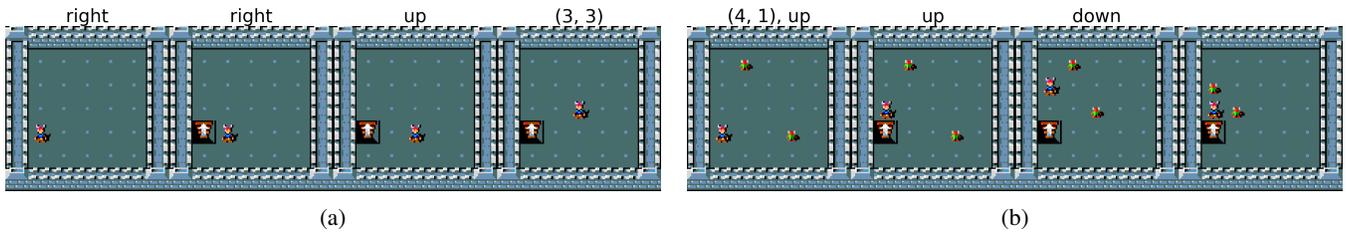


Figure 3: Example trajectories of length 4 in a 5×5 grid for the generative (a) and discriminative (b) datasets, with the corresponding labels above the images. Note that for the discriminative task, the models do not take images as input but rather the symbolic state. The images are provided for visualization purposes.

N	T	E	Death (%)	OOD
10	10	1	17.2	–
		2	60.9	✓
	20	1	89.0	✓
		2	99.0	✓
15	10	1	9.9	✓
		2	32.1	✓
	20	1	75.1	✓
		2	96.7	✓

Table 1: Percentage of trajectories leading to the death of the agent for the discriminative experiment, based on grid size (N), trajectory length (T) and number of enemies (E).

Metric	N	Methods		
		VT	Deep-HMM	NeSy-MM
RE (\downarrow)	5	3.30 ± 0.04	4.97 ± 0.37	3.32 ± 1.80
	10	2.23 ± 0.01	4.66 ± 0.08	3.78 ± 0.07
RA (\uparrow)	5	91.39 ± 0.54	44.55 ± 5.75	97.17 ± 1.00
	10	30.62 ± 1.24	1.54 ± 0.00	89.63 ± 3.59

Table 2: Reconstruction error (RE) and accuracy (RA) for the generative experiment on different grid sizes ($N \times N$). RE is multiplied by 1000, and RA is in percentage.

more steps to reach the agent when the grid is bigger. These differences lead to a difference in class balance from one configuration to the other (Table 1), posing an additional significant learning challenge. Ideally, we want a model that is able to counterbalance the bias inherent to its training data. To test such abilities, we evaluate all methods in terms of both balanced accuracy and F1-score.

5.3 Results

Results for the generative experiment are reported in Table 2 while the results for the discriminative task can be found in Table 3 and Table 4. We report the mean and standard error for all the metrics. We discuss the main findings of both experiments by highlighting the advantages of NeSy-MMs.

Better generative consistency. Integrating knowledge about the environment is clearly advantageous to the generation process in terms of logical consistency, as can be

seen from the reconstruction accuracies. NeSy-MMs significantly outperform both baselines in this regard, especially on larger grids. In terms of reconstruction error, the variational transformer performs on-par or even better than NeSy-MMs. While this shows how transformers are very capable of optimising their losses, the sub-par reconstruction accuracy questions the degree of transfer from optimised solutions to desired solutions. The Deep-HMM generally underperforms compared to both the VT and the NeSy-MM, illustrating the challenge of tackling sequential tasks without logic (NeSy-MM) or a longer time dependency (VT).

Logical interpretability and intervenability. One of the biggest advantages of neurosymbolic generation is its ability to induce interpretable and intervenable logical consistency into subsymbolic generation. As an example, consider the generation in Figure 4 where the generative model was asked to generate a trajectory for the agent, following a given sequence of actions, while adhering to the movement rules of the game. Because the symbolic rules of the game are an inherent part of the generative model, NeSy-MMs generate sequences that perfectly adhere to the mechanics of the game and the provided actions. Other methods lack the necessary semantics or symbolic knowledge to fully guarantee this sort of logical consistency. Moreover, NeSy-MMs allow imposing constraints at test time in addition to the ones used during learning, which corresponds to zero-shot adherence to new queries (Figure 5).

Scaling NeSy to non-trivial time horizons. NeSy methods are known for their scalability issues. When sequential generative settings are considered, the situation is even more dramatic. VAE fails to perform inference on a single sequence of length 10 even on a smaller grid of size 3×3 , with $6h$ timeout. On the contrary, we manage to perform inference and generative learning that does not deteriorate over time, even compared to the neural baselines. In fact, NeSy-MMs perform a forward and backward pass over a batch of sequences in $\approx 0.25s$, for both grid sizes. In the discriminative setting, the presence of a neural network as transition prevented us from applying any existing NeSy system, as they do not provide effective strategies to integrate neural networks except as perception.

Better out-of-distribution generalisation. Pivoting the attention to the discriminative experiment, we can see that NeSy-MMs exploit their relational expressivity to perform



Figure 4: Generated trajectory for actions: right, down, left, up, left, up, right, down.



Figure 5: Generated trajectory for actions: *right*, down, left, up, right, *right*, up, *right*; but with the test-time constraint that the area to the right of the start position should not be entered. When the agent is asked to move in the unsafe area (*i.e.* actions in italics) it, instead, stays in the safe zone, and then it continues following the rest of the instructions.

		Methods			
N	T	E	Transformer	Deep-HMM	NeSy-MM
10	1	1	75.72 ± 1.35	59.88 ± 0.28	64.45 ± 1.10
	2	2	68.61 ± 0.78	38.43 ± 0.21	78.13 ± 0.67
10	1	1	50.40 ± 0.19	49.99 ± 0.39	67.66 ± 0.92
	2	2	16.09 ± 1.33	49.33 ± 0.17	57.47 ± 0.56
15	1	1	78.53 ± 3.09	—	57.22 ± 1.78
	2	2	67.85 ± 1.79	—	75.57 ± 0.42
15	1	1	50.47 ± 0.18	—	71.85 ± 0.58
	2	2	41.54 ± 2.39	—	77.13 ± 2.14

Table 3: Balanced accuracy (%) for the discriminative experiment for grid sizes $N \times N$, with trajectory length T and E enemies. The first line is in-distribution performance, the rest is OOD. Deep-HMM cannot be applied to bigger grids.

well in all the out-of-distribution settings. Both the accuracy (Table 3) and F1-score (Table 4) paint a similar picture: the transformer is able to achieve better performance when staying in distribution ($N = 10$, $H = 10$ and $E = 1$). However, the out-of-distribution settings deteriorate the transformer’s performance. Only when the balance between positive and negative classes is closest to the balance of the training data, *i.e.* when only N is increased to 15 (Table 1), the transformer is able to keep a good accuracy. In contrast, NeSy-MMs show that their relational representations are much more robust to distribution shifts. Deep-HMMs land somewhere in the middle between transformers and NeSy-MMs as their performance is always lower than NeSy-MMs, but depending on the case they can be more robust than the transformers. Finally, notice that Deep-HMMs cannot be applied to larger grid sizes, limiting their OOD capabilities.

6 Conclusion

We introduced relational neurosymbolic Markov models (NeSy-MMs), a powerful new class of relational proba-

		Methods			
N	T	E	Transformer	Deep-HMM	NeSy-MM
10	1	1	0.61 ± 0.02	0.25 ± 0.01	0.41 ± 0.03
	2	2	0.59 ± 0.02	0.56 ± 0.00	0.79 ± 0.01
10	1	1	0.02 ± 0.01	0.79 ± 0.01	0.92 ± 0.01
	2	2	0.02 ± 0.01	0.98 ± 0.00	0.99 ± 0.00
15	1	1	0.57 ± 0.03	—	0.15 ± 0.02
	2	2	0.52 ± 0.03	—	0.65 ± 0.01
15	1	1	0.02 ± 0.01	—	0.78 ± 0.02
	2	2	0.02 ± 0.01	—	0.98 ± 0.01

Table 4: F1-Score for the discriminative experiment. This follows the same notation as Table 3.

bilistic models that can incorporate neural networks beyond just perception modules. These models are provided with a novel scalable and differentiable particle filtering technique for inference and learning, facilitating the bidirectional flow of information necessary for a proper neurosymbolic model (**D.III**). Our empirical results show that the integration of relational symbolic knowledge into deep Markov models leads to significant improvements in generative and discriminative tasks (**D.IV**), while also providing guarantees that neural models alone cannot achieve. Importantly, we stressed the relational aspect of NeSy-MMs by showing that purely neural models and even deep probabilistic models struggle to learn representations that generalise to unseen data and settings (**D.I**). While such generalisation behaviour is inherent to many neurosymbolic approaches, our experiments showed that NeSy-MMs scale to sequential settings beyond the reach of existing NeSy systems (**D.II**).

Future work will focus on further applying NeSy-MMs to new settings, *e.g.* reinforcement learning, and applications where continuous random variables are used differently from image generation, *e.g.* physical systems.

Acknowledgments

This research has also received funding from the KU Leuven Research Funds (C14/24/092, STG/22/021, iBOF/21/075), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement n°101142702).

References

- Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34: 17981–17993.
- Badreddine, S.; Garcez, A. d.; Serafini, L.; and Spranger, M. 2022. Logic tensor networks. *Artificial Intelligence*.
- Baum, L. E.; and Petrie, T. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6): 1554–1563.
- Bishop, C. M. 2006. Pattern recognition and machine learning. *Springer google schola*, 2: 645–678.
- Ciatto, G.; Sabbatini, F.; Agiollo, A.; Magnini, M.; and Omicini, A. 2024. Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review. *ACM Computing Surveys*, 56(6): 1–35.
- Corenflos, A.; Thornton, J.; Deligiannidis, G.; and Doucet, A. 2021. Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, 2100–2111. PMLR.
- Darwiche, A. 2020. An Advance on Variable Elimination with Applications to Tensor-Based Computation. In *ECAI 2020*, 2559–2568. IOS Press.
- de Penning, L.; Garcez, A.; Lamb, L. C.; and Meyer, J. 2011. A neural-symbolic cognitive agent for online learning and reasoning. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, volume 2, 1653–1658. International Joint Conferences on Artificial Intelligence.
- De Raedt, L.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis lectures on artificial intelligence and machine learning*.
- De Raedt, L.; Kimmig, A.; and Toivonen, H. 2007. ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In *IJCAI*. Hyderabad.
- De Smet, L.; Sansone, E.; and Zuidberg Dos Martires, P. 2023. Differentiable Sampling of Categorical Distributions Using the CatLog-Derivative Trick. In *NeurIPS*.
- De Smet, L.; Zuidberg Dos Martires, P.; Manhaeve, R.; Marra, G.; Kimmig, A.; and De Raedt, L. 2023. Neural Probabilistic Logic Programming in Discrete-Continuous Domains. *UAI*.
- Dean, T.; and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2): 142–150.
- Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.
- Holtzen, S.; Van den Broeck, G.; and Millstein, T. 2020. Scaling exact inference for discrete probabilistic programs. *Proceedings of the ACM on Programming Languages*, 4(OOPSLA): 1–31.
- Huang, J.; Li, Z.; Chen, B.; Samel, K.; Naik, M.; Song, L.; and Si, X. 2021. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. *NeurIPS*.
- Hummel, J. E.; and Holyoak, K. J. 2003. A symbolic-connectionist theory of relational inference and generalization. *Psychological review*, 110(2): 220.
- Juang, B. H.; and Rabiner, L. R. 1991. Hidden Markov models for speech recognition. *Technometrics*, 33(3): 251–272.
- Khiatani, D.; and Ghose, U. 2017. Weather forecasting using hidden Markov model. In *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 220–225. IEEE.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kisa, D.; Van den Broeck, G.; Choi, A.; and Darwiche, A. 2014. Probabilistic sentential decision diagrams. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*.
- Koller, D.; and Friedman, N. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.
- Kool, W.; van Hoof, H.; and Welling, M. 2019. Buy 4 reinforce samples, get a baseline for free! *ICLR Deep RL Meets Structured Prediction Workshop*.
- Krishnan, R.; Shalit, U.; and Sontag, D. 2017. Structured inference networks for nonlinear state space models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).
- Küttler, H.; Nardelli, N.; Miller, A.; Raileanu, R.; Selvatici, M.; Grefenstette, E.; and Rocktäschel, T. 2020. The nethack learning environment. *Advances in Neural Information Processing Systems*, 33: 7671–7684.
- Liu, R.; Regier, J.; Tripuraneni, N.; Jordan, M.; and McAuliffe, J. 2019. Rao-Blackwellized stochastic gradients for discrete distributions. *ICML*.
- Manhaeve, R.; Dumančić, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2021. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence*.
- Marra, G.; Dumančić, S.; Manhaeve, R.; and De Raedt, L. 2024. From Statistical Relational to Neurosymbolic Artificial Intelligence: a Survey. *Artificial Intelligence*, 104062.

- Misino, E.; Marra, G.; and Sansone, E. 2022. VAE: Bridging Variational Autoencoders and Probabilistic Logic Programming. In *NeurIPS*.
- Mor, B.; Garhwal, S.; and Kumar, A. 2020. A Systematic Review of Hidden Markov Models and Their Applications. *Archives of Computational Methods in Engineering*, 28: 1429 – 1448.
- Morettin, P.; Zuidberg Dos Martires, P.; Kolb, S.; and Passerini, A. 2021. Hybrid Probabilistic Inference with Logical and Algebraic Constraints: a Survey. In *IJCAI*.
- Murphy, K.; and Russell, S. 2001. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Sequential Monte Carlo methods in practice*, 499–515. Springer.
- Nitti, D.; De Laet, T.; and De Raedt, L. 2016. Probabilistic logic programming for hybrid relational domains. *Machine Learning*.
- Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.
- Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N.; and Prabhat, F. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743): 195–204.
- Russell, S.; and Norvig, P. 2020. *Artificial Intelligence: A Modern Approach*. Hoboken: Pearson, 4th edition edition. ISBN 978-0-13-461099-3.
- Samvelyan, M.; Kirk, R.; Kurin, V.; Parker-Holder, J.; Jiang, M.; Hambro, E.; Petroni, F.; Kuttler, H.; Grefenstette, E.; and Rocktäschel, T. 2021. MiniHack the Planet: A Sandbox for Open-Ended Reinforcement Learning Research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
- 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.
- Ścibior, A.; Masrani, V.; and Wood, F. 2021. Differentiable Particle Filtering without Modifying the Forward Pass. In *International Conference on Probabilistic Programming (PROBPROG)*.
- Towell, G. G.; and Shavlik, J. W. 1994. Knowledge-based artificial neural networks. *Artificial intelligence*, 70(1-2): 119–165.
- Tran, S. N. 2021. Compositional Neural Logic Programming. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 3059–3066. International Joint Conferences on Artificial Intelligence.
- Tran, S. N.; and Garcez, A. d. 2023. Neurosymbolic reasoning and learning with restricted boltzmann machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 6558–6565.
- Tsamoura, E.; Carral, D.; Malizia, E.; and Urbani, J. 2021. Materializing knowledge bases via trigger graphs. *Proceedings of the VLDB Endowment*, 14(6): 943–956.
- van Krieken, E.; Thanapalasingam, T.; Tomczak, J.; Van Harmelen, F.; and Ten Teije, A. 2024. A-nesi: A scalable approximate method for probabilistic neurosymbolic inference. *Advances in Neural Information Processing Systems*, 36.
- Van Roy, M.; Robberechts, P.; Yang, W.-C.; De Raedt, L.; and Davis, J. 2023. A Markov framework for learning and reasoning about strategies in professional soccer. *Journal of Artificial Intelligence Research*, 77: 517–562.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.
- Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30.
- Yang, W.-C.; Marra, G.; Rens, G.; and De Raedt, L. 2023. Safe Reinforcement Learning via Probabilistic Logic Shields. In Elkind, E., ed., *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 5739–5749. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Yang, Z.; Ishay, A.; and Lee, J. 2020. Neurasp: Embracing neural networks into answer set programming. In *IJCAI*.
- Younis, A.; and Sudderth, E. B. 2023. Differentiable and Stable Long-Range Tracking of Multiple Posterior Modes. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Zhang, H.; Dang, M.; Peng, N.; and Van Den Broeck, G. 2023. Tractable Control for Autoregressive Language Generation. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 40932–40945. PMLR.