EVENFLOW

Robust Learning and Reasoning for Complex Event Forecasting

EVENFLOW White Paper

Neurosymbolic Learning, Forecasting & Verification in EVENFLOW

www.evenflow-project.eu

Robust Learning and Reasoning for Complex Event Forecasting

White Paper

Neurosymbolic Learning, Forecasting & Verification in EVENFLOW

Nikos Katzouris[1], Nikos Manginas[1,3], Vasilis Manginas[1], Elina Syrri[1], Elias Alevizos[1], Alessio Lomuscio[2], Georgios Paliouras[1]

[1]Institute of Informatics & Telecommunications, NCSR "Demokritos", Athens, Greece
[2]Department of Computing, Imperial College London, UK
[3]Department of Computer Science and Leuven.AI, KU Leuven, Belgium

Executive Summary

The Horizon Europe project EVENFLOW develops hybrid learning and reasoning techniques for complex event forecasting, which combine deep learning with logic-based learning and reasoning into neuro-symbolic forecasting models. These methods combine neural representation learning techniques, capable of constructing event-based features from streams of perception-level data with symbolic learning and reasoning tools, that utilize such features to synthesize high-level, interpretable patterns of critical situations to be forecast. To deal with the brittleness of neural predictors and the high volume/velocity of temporal data flows, the EVENFLOW techniques rely on novel, formal verification techniques for machine learning, in addition to a suite of scalability algorithms for federated training and incremental model construction. The forecasters developed in EVENFLOW are interpretable and scalable, allowing for explainable insights, delivered in a timely fashion, thus enabling proactive decision making. The EVENFLOW techniques are evaluated on three use cases related to (i) oncological forecasting in precision medicine, (ii) safe and efficient behaviour of autonomous transportation robots in smart factories and (iii) reliable life cycle assessment of critical infrastructure.

In this white paper we summarize the work that has been done in the project regarding neuro-symbolic temporal learning, reasoning, forecasting and neuro-symbolic verification. In particular, we present a brief overview of EVENFLOW techniques for:

- Jointly training neural networks alongside temporal knowledge, in order to align neural perceptive modules with temporal reasoning tasks and explaining the predictions of the hybrid, neuro-symbolic models.
- Forecasting the occurrence of imminent critical events from perceptual data streams, in order to allow for proactive decision-making, in a robust and transparent fashion.
- Learning interpretable temporal complex event models from sub-symbolic input and utilizing such models for detecting and forecasting such events.
- Obtaining formal guarantees on the robustness of hybrid, neuro-symbolic systems, i.e. formally verifying the property that small perturbations on the subs-symbolic input do not affect the high-level reasoning output.

1 From Perceptual Streams to Actionable Insights

The data driving decision-making in many critical sectors increasingly arrive as perceptual streams: sequences of video frames from cameras, multi-variate trajectories of moving objects (e.g. robots, vehicles), or high-dimensional time series from infrastructure monitoring in predictive maintenance, patient monitoring in medical applications and so on. On their own, these streams are opaque to human decision-makers. What operators, engineers, or clinicians actually care about are semantically meaningful events, e.g. a dangerous driving pattern in a traffic camera video feed, a safety-threatening near-collision collision incident between robots in a smart factory, indicators regarding how the condition of a patient evolves over time and so on. Even more than that, stakeholders are primarily interested in early warnings about how and when such events may unfold in the future, in order to respond proactively.

Complex Event Recognition and Forecasting (CER/F) offers a principled way to move from raw streams to such higher-level situational awareness. In CER/F, low-level observations are abstracted into simple events, which are then combined into more complex event hierarchies. However, traditional CER/F solutions assume that simple events are already available as clean symbolic inputs and that complex event patterns are hand-crafted by experts. More often than not, however, neither assumption holds: symbols must be extracted from noisy perception signals using sophisticated machine learning models, patterns evolve as processes and environments change, and labels for intermediate events across several levels of an event hierarchy are scarce or entirely missing.

In the remainder of this section we review the foundations of CER/F and highlight how they extend to settings characterized by sub-symbolic, perceptual input, setting the stage for the temporal neuro-symbolic methods introduced in the rest of this white paper.

1.1 Complex Event Recognition & Forecasting

Complex Event Recognition [6] and **Forecasting** [1] **(CER/F)** systems seek to detect, or even forecast ahead of time, occurrences of special events of interest, across a set of input data streams. The input streams consist of *simple events*, i.e. time-stamped pieces of information, and the output are the detected/forecast instances of the target situations, which are called *complex events* and are usually defined as spatio-temporal combinations of the simple events.

CER/F systems typically rely on a set of complex event patterns, which are declarative specifications of the interesting situations to be monitored across the input datastreams. Such situations usually involve sets of correlated events that are expected to occur in a sequential fashion. Due to the sequential nature of such complex event patterns, the computational objects that correspond to such patterns are some type of automata (finite state machines), typically, *symbolic automata*, where the transitions are guarded by predicates, rather than by mere symbols from a finite alphabet. The recognition process then amounts to matching such automata-based patterns against the simple event input, i.e. reaching an accepting state in the automaton during processing the input stream. The forecasting task amounts to deriving probabilistic estimates of future full pattern matches from partial matches that have been observed so far. Figure 1 presents an example of such a pattern for an overtake complex event specification in autonomous driving/traffic monitoring applications.

The symbolic nature of traditional CER/F systems restricts their applicability to symbolic input. However, numerous applications deal with sub-symbolic, perceptual level input, such as video, audio or high-dimensional time series. A typical baseline approach in such cases is to train a neural predictor to

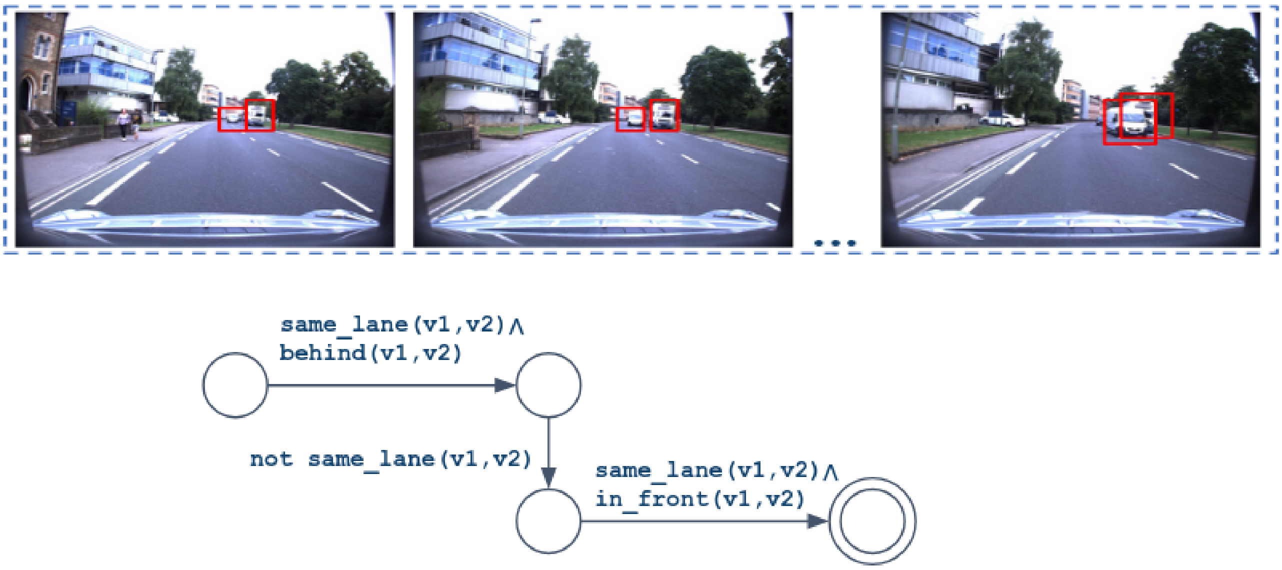


Figure 1: A simple pattern specifying a situation where vehicle v_1 overtakes vehicle v_1 via a set of sequential conditions. Initially the two vehicles are in the same lane with v_1 being behind v_2. This is followed by a condition where the two vehicles are not in the same lane, capturing the part of the episode where v_1 changes lanes to overtake v_2. The episode is completed once the two vehicles are again in the same lane, this time with v_1 leading. A partial pattern match is a path in the automaton that has not yet reached the accepting state (the double-circled one). Partial matches correspond to early stages in evolving complex event episodes and can be used to estimate the likelihood of episode completion (full pattern matches) from data, thus *forecasting* complex event occurrences. Matching such patterns on sub-symbolic input, such as video feeds from traffic monitoring cameras, requires Neuro-symbolic techniques where perception neural networks extract simple events from the video frames - e.g. $incoming\_lane(v_1)$, $outgoing\_lane(v_2)$ etc - and symbolic modules that reason over these predictions logically - e.g. for inferring spatial relations between the vehicles - and temporally, to ensure that the logical conditions in the pattern are indeed observed in the required order.

map the sub-symbolic input to a set of symbols, corresponding to the simple events in our case, which are then passed to the symbolic model that handles the downstream CER/F task. Such approaches are of **neuro-symbolic (NeSy)** nature, since they combine neural and symbolic components, albeit in a loosely coupled fashion, and are thus often sub-optimal: the neural predictors are trained in isolation, ignoring the downstream task and the symbolic components ignore the stochastic, error-prone nature of the neural grounding process that produces its input symbols. They are also often infeasible, since they require large amounts of simple event-labeled data, which are usually difficult to obtain.

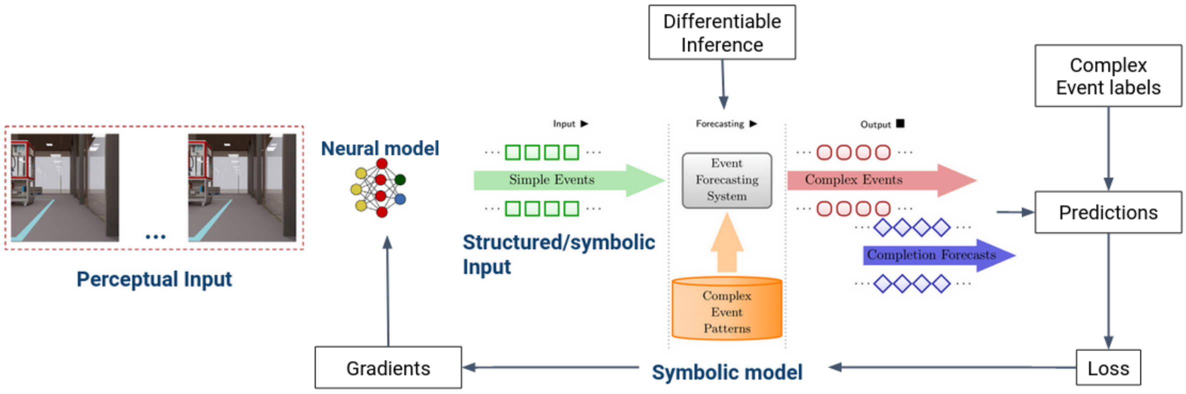


Figure 2: Neuro-symbolic Complex Event Recognition & Forecasting.

In contrast, **tightly integrated NeSy AI** [10] approaches treat perception and symbolic reasoning as a single, coupled learning problem, allowing the CER/F model to shape how symbols are grounded and, conversely, allowing uncertainty in the grounding process to propagate through the temporal reasoning layer. Neural components can be trained under losses derived from temporal logics and automata that define complex events, so that they learn to align their predictions with the requirements of temporal reasoning. At the same time, symbolic components can be extended with probabilistic semantics to account for the graded, error-prone outputs of neural predictors, enabling robust probabilistic forecasts over streams of noisy simple events. This tighter NeSy integration promises CER/F systems that require far fewer simple-event labels, generalize better to out-of-distribution temporal patterns, and provide explanations in terms of human-understandable patterns (rules, automata) that remain anchored to the underlying perceptual data.

2 Temporal Neurosymbolic Learning & Reasoning

A central challenge in temporal NeSy AI is to couple perception models with rich temporal knowledge for reasoning over long, noisy data streams (video, multivariate time series) and answer queries that are naturally probabilistic and counterfactual – e.g., "what is the probability that this pattern will occur within the next ten steps?".

To address this problem, in EVENFLOW we developed **NeSyA (Neuro-Symbolic Automata)**[1] [8], a NeSy technique for seamlessly integrating perception neural networks with symbolic temporal knowledge. NeSyA is a formal probabilistic framework for joint NeSy training of neural and symbolic temporal models. In particular, NeSyA enables the following:

- Declaratively specify existing domain knowledge, either as a symbolic automaton, or as a formula in Linear Temporal Logic, which is compiled into such an automaton at runtime. Such knowledge may dictate patterns specifying events whose sequential occurrence across the input streams needs to be monitored, or temporal constraints that need to be satisfied by neural networks processing such streams.

- Train perception networks alongside that knowledge, so that **perception is aligned with the temporal reasoning tasks.** This means that these networks are not trained in isolation on dense latent concept (simple event) supervision, but instead, trained with an objective directly related to the downstream task of complex event detection, and in an indirect fashion, from complex event labels only. Of course, simple event ground truth can be utilized as well, if available.

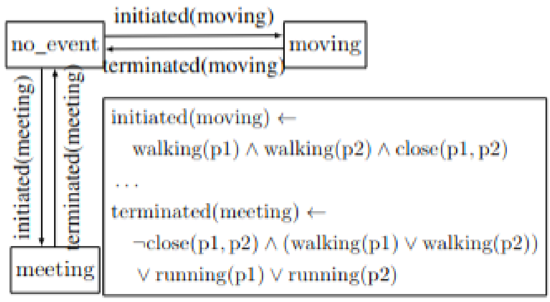
- **Explain the behavior** of the hybrid (neural + symbolic) model, thanks to the formal probabilistic semantics of the NeSyA framework. Probabilistic reasoning is at the core of NeSyA: Knowledge Compilation techniques [4, 3] are used to compile the logical formulas that guard the transitions in the event pattern automata into tractable probabilistic circuits. These circuits are used to calculate the probabilities of logical expressions over sets of symbols from the probabilities of the symbols, which are predicted by the neural component from the input streams. Using the probabilities of the automaton's transitions and Markov Chain techniques, NeSyA can then compute the probability that the automaton is in each one of its states at each point in time, as well as the most likely paths through the automaton's transitions that are responsible for that. Since some states are semantically linked to the target events, such inferences can be used to explain how and why certain events of interest were detected.

Figure 3 presents an illustration of NeSyA on interpretable activity recognition from video. NeSyA supports probabilistic queries that are directly related to explainability, such as marginals and most likely explanations for observed events. Despite the fact that such queries are intractable, we get them "for free" in NeSyA, thanks to its backbone of knowledge compilation and probabilistic circuits, which allow for answering such queries in time linear in the circuit size.

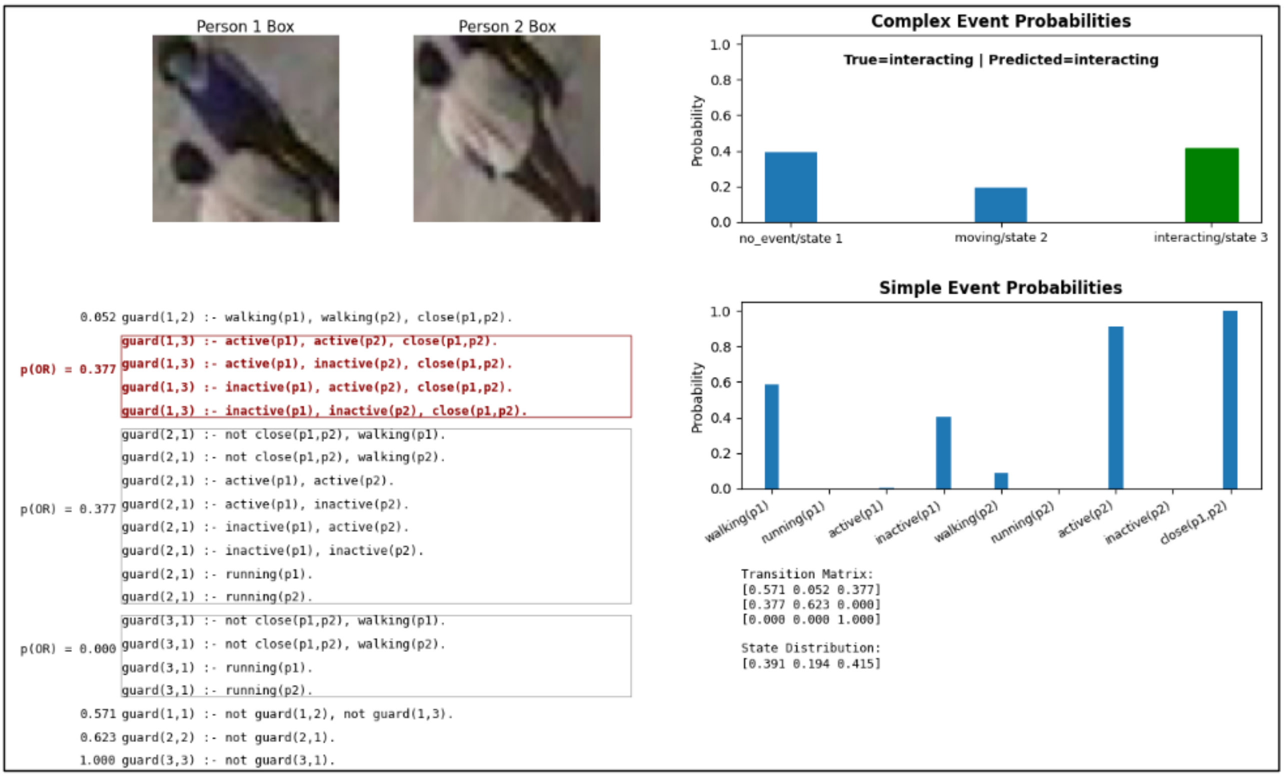
Note that the neural network that is used in the scenario of Figure 3 is trained from a small number of training sequences with **scarce event-based supervision** and **significantly outperforms SoA purely neural baselines, such CNN + LSTM/Transformer stacks, especially in out-of-Distribution (OOD) settings** [8]. Similar results have been obtained in additional real-world applications, such as **autonomous driving** [2] – see Figure 4, as well as challenging synthetic benchmarks.

[1] https://github.com/nmanginas/nesya

(a) Example input image sequence.



(b) Symbolic automaton used by NeSyA.



(c) Graphical explanation of NeSyA's predictions.

Figure 3: NeSyA on intrepretable human activity recognition from video. **(a)** Images from the input video frames are first processed by a CNN. The bounding boxes corresponding to the two tracked persons in this scenario (p_1, $p2$) are mapped to *simple events* – e.g. $walking(p_1)$ at time t) – corresponding to individual activities; **(b)** A deterministic symbolic automaton encoding *complex events* as spatio-temporal combinations of simple events. The complex events are *two people moving together*, or *meeting and interacting*. Only a subset of the transition logic is shown for brevity. In the case that no outgoing transition from a state is satisfied the SFA loops in its current state.; **(c)** Graphical explanation of NeSyA's predictions for a particular frame in the input video. The entire transition logic is illustrated as a logic progrm. Simple event probabilities, predicted by the CNN are linked to the logical rules, which in turn, cause the automaton to transition across states. For instance, the disjunctive rules higlighted in red are responsible for a transition to state 3 in this frame (corresponding to the *interacting* complex event), causing this event to be recognized.

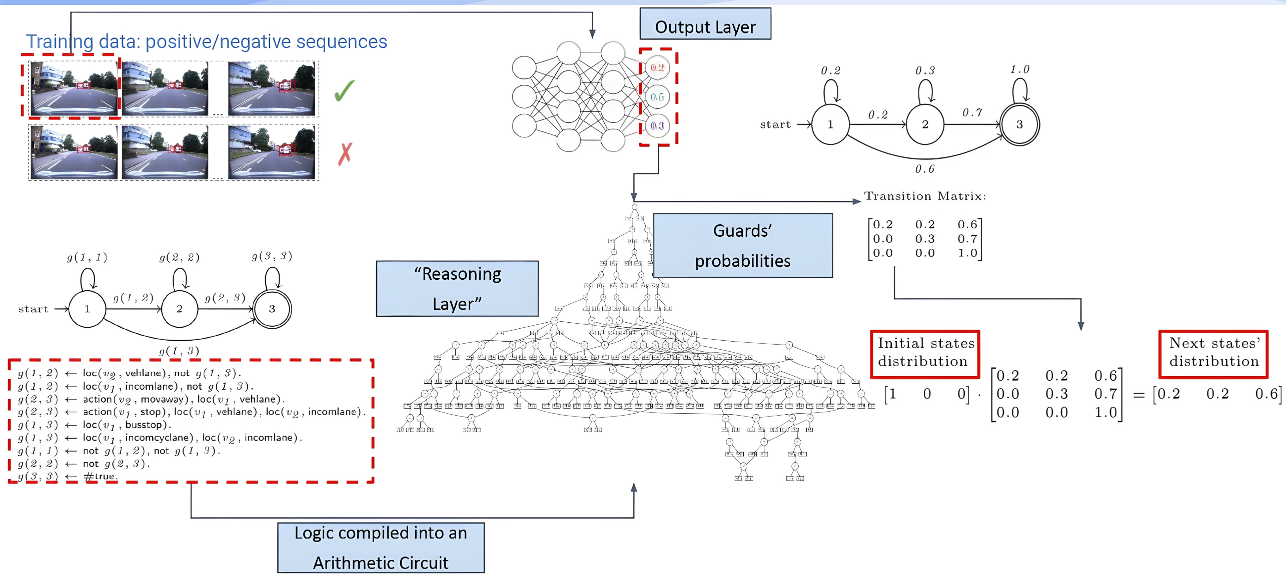


Figure 4: Illustration of the inference and training procedure in NeSyA in an autonomous driving domain, where the goal is to recognize overtake incidents from camera footage, in an interpretable fashion. A symbolic automaton, learned from fully labelled symbolic sequences, captures overtake definitions and simulates symbolic background knowlegde that is combined with the perception neural module. First, videos are processed through this module, which outputs simple event probability distributions (*Output Layer*). These probabilities help answer the probabilistic query of whether the sequence is in a certain state at a given frame (*Guards' Probabilities*), utilizing a compiled Boolean circuit (*Logic compiled into an Arithmetic Circuit*). Over time, each state accumulates probabilities, by multiplying the probability distribution with the *Transition Matrix*, resulting in a final state probability distribution at the end of the sequence. We use this distribution to compute the loss for the 'overtake' incident prediction and backpropagate the loss to train the network, repeating the process until we achieve the minimum loss value.

3 Neurosymbolic Forecasting

Central to EVENFLOW is taking the event detection task a step forward, towards forecasting critical events before they actually occur, in order to facilitate proactive measures in decision making, instead of merely detecting them in a post-hoc fashion. The backbone for event forecasting in EVENFLOW is the **Wayeb**[2] system [1].

Wayeb is an online, probabilistic system designed for Complex Event Forecasting, addressing the challenge of predicting the potential occurrence of a declaratively defined Complex Event pattern (often formulated as a Symbolic Regular Expression (SRE)) within an event stream before it is actively detected by a Complex Event Recognition (CER) engine. Wayeb converts an SRE into a Deterministic Symbolic Finite Automaton (DSFA), which, when consuming the input stream, is functionally analogous (via isomorphism) to a classical deterministic automaton operating over the minterms of the DSFA predicates. To model the statistical properties of the stream, Wayeb employs Variable-order Markov Models (VMMs), specifically, Prediction Suffix Trees (PST), which capture long-term dependencies, while avoiding the computational explosion associated with exhaustive enumeration in fixed-order models. The probabilistic model is constructed by learning the PST from the minterms derived from the DSFA, using an approach that either involves creating an embedding of a probabilistic automaton within the DSFA by taking their Cartesian product, or, for superior memory

[2] https://github.com/ElAlev/Wayeb

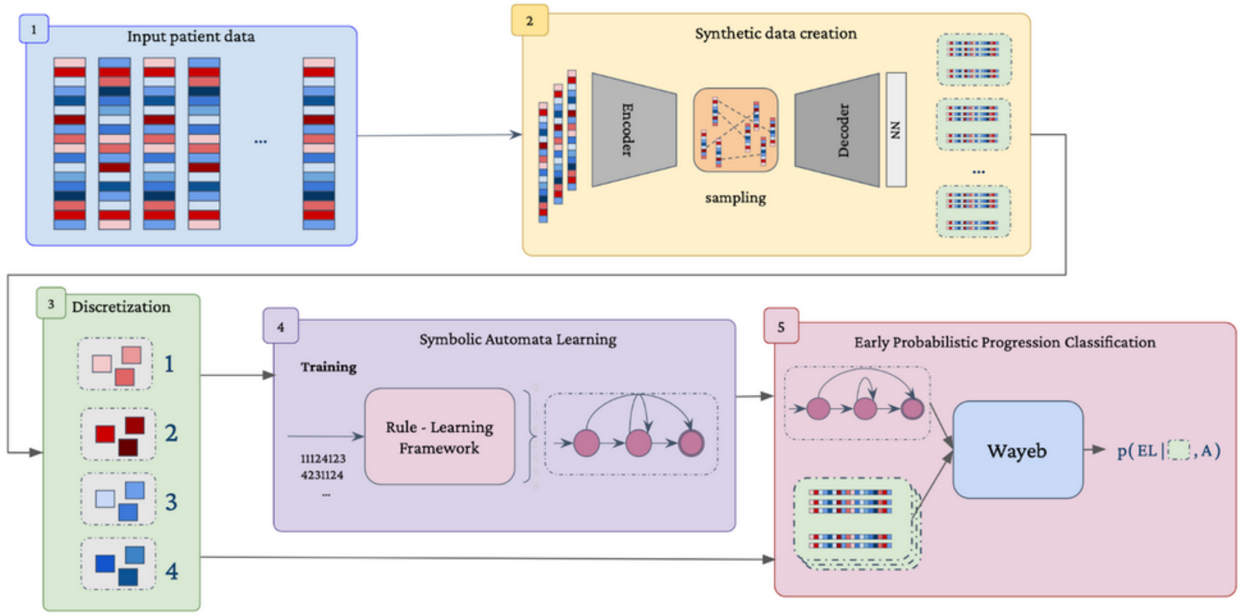


Figure 5: Illustration of the event forecasting process in EVENFLOW's Personilized Medicine use case on cancer progression. The purpose in this case is to make timely forecasts for synthetic patients transitioning from early cancer to late cancer states from high-dimentional gene expression time series. To that end, the time series are first preprocessed to reduce their dimentionality, thus focusing only on the most relevant genes based on gene panels from the literature, and are discretized into a set of symbols. Next, a symbolic automaton that captures cancer stage transition patterns is learned from these sequences (see also Section 4), and finally the induced automaton is used alongside Wayeb, in order to forecast transition incidents from early signs.

efficiency, by directly estimating waiting-time distributions through recursive traversal of the PST, thereby bypassing the construction of the probabilistic automaton. These calculated waiting-time distributions, based on the theory of absorbing Markov chains, allow Wayeb to output forecasts, typically in the form of intervals [start, end], representing the predicted number of future events until pattern completion with a user-defined confidence threshold θ. Wayeb has been demonstrated to achieve high throughput and competitive accuracy compared to state-of-the-art solutions, often leveraging its ability to accommodate higher-order models for enhanced performance.

Figure 5 presents an overview of how Wayeb was used in the project, using here one of EVENFLOW's use cases on Personalized Medicine as an example. Along with the Barcelona Supercomputing Center (BSC) that lead this use case, we were able to show with this approach that we can achieve timely forecasts on cancer progression on synthetic patient data. These forecasts were comparable in predictive quality and earliness to those of purely neural baselines, with the additional advantage of being transparent and interpretable. The latter is thanks to the symbolic nature of the event patterns that specify the conditions that we wish to monitor and the transparent fashion in which Wayeb operates.

We attained similar results in EVENFLOW's Industry 4.0 use case – see Figure 6, where the goal was to forecast deadlock incidents between Automated Guided Vehicle (AGV) robots in a smart factory, in order for the robots to timely re-plan their trajectories. Along with our partners from DFKI who lead this use case, we were able to show that forecasting deadlock incidents and using the forecasts for path re-planning in real-time can significantly reduce deadlock-induced down-times.

In addition to the aforementioned loosely coupled NeSy forecasting pipelines, we have also developed more tightly integrated NeSy forecasting techniques. The most recent advancement, which

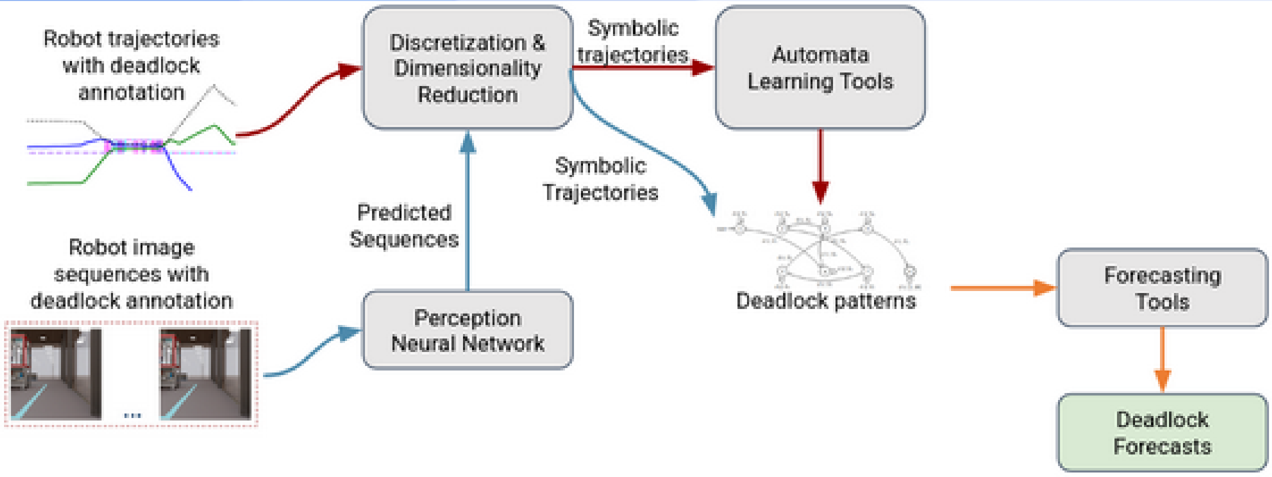


Figure 6: An overview of deadlock forecasting in EVENFLOW's Industry 4.0 use case.

we call *Mutual Information Markov Models (MiMM)*, attempts to model temporal phenomena as Markov processes over abstract states. Learning interpetable Markov abstractions directly from high-dimensional streams is an important open problem with clear links to forecasting: good abstractions should encapsulate the temporal information needed to accurately predict future events. Existing techniques, such as Hidden Markov Models (HMMs), and their extensions (including neural extensions) attempt to generatively model the data, which assume a process of reconstructing the observations given the states. However, this reconstruction process is at odds with interpretability, since it requires to encapsulate into the states a large amount of otherwise irrelevant data characteristics.

MiMM is a NeSy method for discovering discrete latent states and transition dynamics directly from high-dimensional Markov data (e.g., image streams), without reconstructing the observations. A neural network maps each observation to a probabilistic assignment over a finite set of latent states, and its parameters are trained to maximize the mutual information between successive latent states, yielding an abstract Markov chain that preserves the essential temporal structure of the original process, while remaining low-dimensional and interpretable. Prior knowledge about the system's dynamics (e.g., a PRISM model) can be injected by regularizing the learned transition matrix towards a symbolic prior, ensuring that the discovered states and transitions respect known behaviour. MiMM provides a way to derive such latent Markov models from raw data and then use probabilistic model checking to answer forecasting queries, such as the probability of reaching a critical or goal state within a given time horizon, thereby directly supporting the EVENLOW's objectives on neuro-symbolic complex event forecasting and robust, explainable decision support. Figure 7 illustrates some indicative results on using this method on water pipe leakage forecasting in EVENFLOW.

4 Event Pattern Learning

In the previous sections we have mostly assumed that the symbolic knowledge that is input to a NeSy system is given beforehand. Under this assumption we presented a technique, NeSyA, tailored specifically for temporal event-based domains, for training the neural part of the NeSy system alongside the knowledge, so that the two components (neural/symbolic) are aligned. However, the assumption of existing knowledge does not always hold. In many event-based applications the patterns that we are interested to monitor are not known beforehand, or they evolve over time as the underlying processes or data characteristics change. This motivates the need for learning such patterns directly

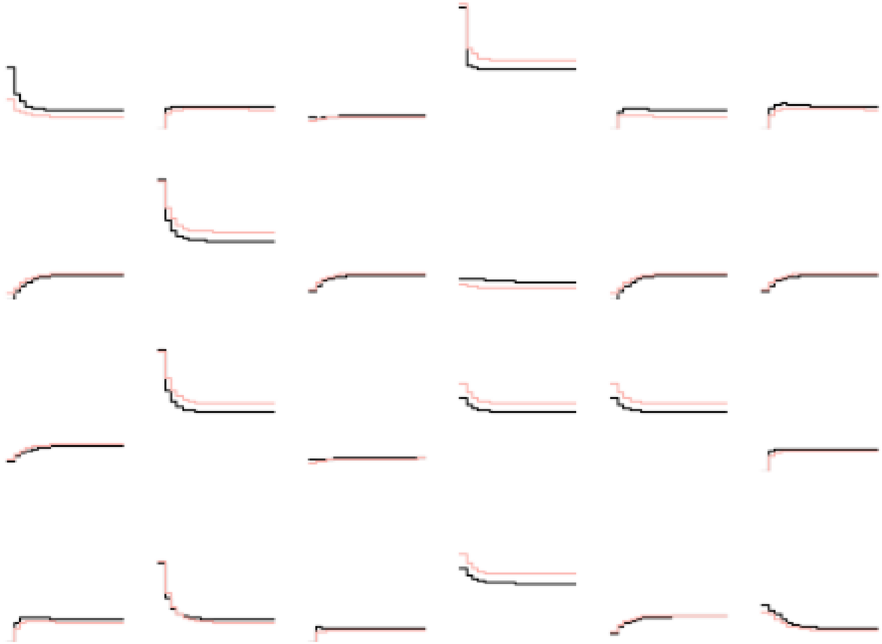


Figure 7: Indicative NeSy forecasting results with MiMM on EVENFLOW's water pipe leakage use case for various leakage types (classes) and a forecast horizon of 20 seconds. Black lines show the actual signal and red lines correspond to the forecast signal.

from data, rather than relying on hand-crafted specifications, allowing us to automate part of the pattern engineering process and to seamlessly adapt to new situations. However, although knowledge learning in a purely symbolic setting (e.g., inductive logic programming, automata and grammar induction) is a well-studied problem, it is much less explored in the case where the input data are of perceptual nature, such as videos (image sequences) or high-dimensional time-series data.

A straightforward approach to symbolic knowledge acquisition in such cases is to train a neural network to map percepts to symbols and then use an off-the-shelf symbolic learner to induce temporal structure from the network's symbolic inferences. This approach has several shortcomings, outlined earlier in Section 1.1.

To address this knowledge acquisition problem, in EVENFLOW we developed event patter learning techniques capable of learning symbolic temporal patterns of interesting situations, either from symbolic sequences, multi-variate time series, or even more complex perceptual input, such as image sequences. The backbone for event structure learning in EVENFLOW has been **ASAL (Answer Set Automata Learning)**[3] [7].

ASAL is a framework for learning and revising complex event patterns represented as symbolic finite automata (SFA) from labeled streams of multivariate event-based data. In ASAL, a symbolic temporal model that accepts or rejects the input event traces is encoded as an answer set automaton (ASA), i.e., an answer set program that combines a generic automata interpreter, a specification of the automaton's structure (states and transitions), background predicates that operate over event tuples (e.g., trends, thresholds, attribute comparisons etc), and transition guard rules, defined as boolean combinations of the background predicates. This formulation allows ASAL to express the core Complex Event Recognition (CER) operators of sequence, iteration, and filtering, while making both the automaton's structure and the guard conditions learnable, thanks to the strong connections of

[3]https://github.com/nkatzz/asal

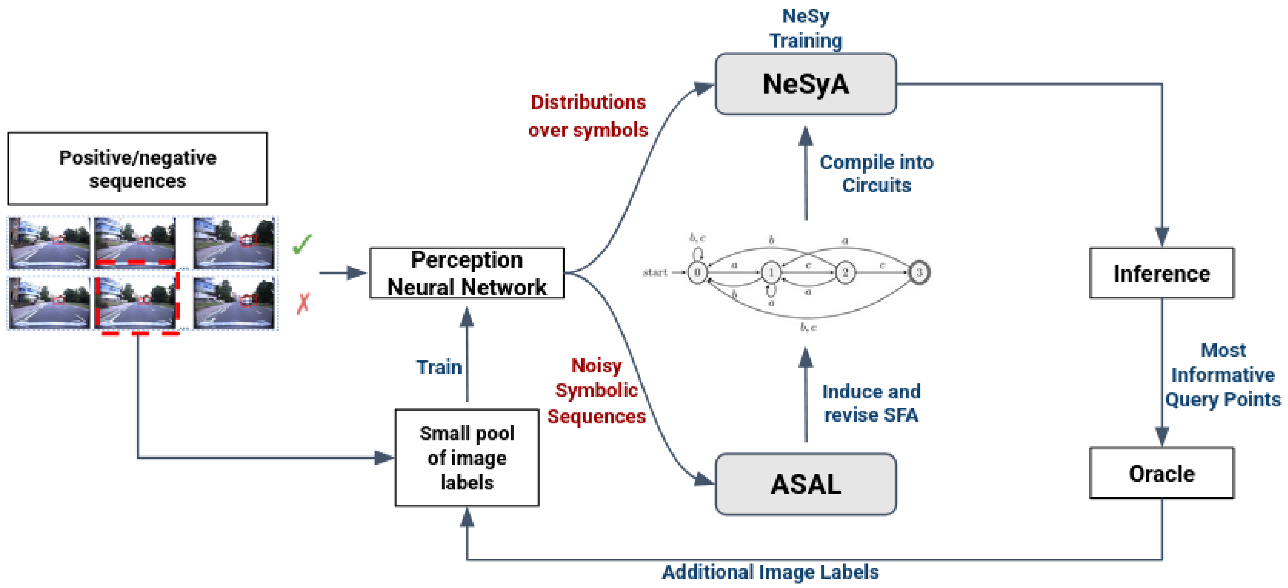


Figure 8: NeurASAL overview.

Answer Set Programming (ASP) to symbolic learning. In particular, ASAL casts SFA induction as an abductive, constraint-driven learning process: temporal structure and guard definitions are generated and tested against constraints related to predictive accuracy/minimality tradeoffs, in an effort to approximate a global optimum in the training data. To alllow for scaling to larger domains (long sequences of higher dimensionality), ASAL comes with an incremental variant, which trades optimality for efficiency and is based on iterative SFA revision from data batches in a Monte Carlo Tree Search framework.

ASAL was developed in EVENFLOW and was used extensively in the project for learning the event patterns to use alongside the Wayeb system in the project's use cases, in order to allow for interpretable Nesy forecasting – see also Figures 5 and 6. Moreover, it has recently been extended within the project in several ways, as follows:

Joint Neural-Symbolic Learning. ASAL has been combined with NeSyA (see Section 2), towards a framework that can learn symbolic patterns, while training a perception neural network to map percepts to the symbols that these patterns use. An overview of the approach is presented in Figure 8.

The input to this neural/symbolic learner consists of training image sequences with downstream (complex event) labels only, together with a small subset of images from these sequences for which simple event labels are also available. ASAL and NeSyA are combined in a co-training framework in which a perceptual neural network is initially partially trained on the small pool of labeled images. The partially trained network is then used to predict simple event labels for all images, thereby inducing noisy symbolic sequences from the raw training sequences. ASAL is subsequently used to learn an initial symbolic automaton (SFA) from these sequences. To account for the noise introduced by the poorly trained neural network, each sequence is weighted by the average entropy of the neural predictions across time, and ASAL incorporates these weights during induction, alongside a Minimum Description Length (MDL) heuristic, to bias its search towards models that explain low-entropy sequences while discounting high-entropy ones when they incur a large MDL penalty. NeSyA is then used to further train the neural network for a few epochs, using the SFA induced by ASAL as a "teacher". In this step, two losses are combined: the standard NeSyA loss from sequence misclassification and the image-level misclassification loss for the labeled images. The system then employs active learning principles to identify the most informative images to label and requests a batch of new labels under a query budget. ASAL induces an improved SFA using the newly acquired labels (rather than the network's pseudo-labels), and the joint training loop continues until convergence. This approach ensures that the neural and symbolic components are trained jointly and are mutually informed. Experiments on synthetic NeSy benchmarks indicate that convergence is achieved with only a fraction of the labeled images that would be required without NeSy training or without an active learning heuristic.

Differentiable Learning of Symbolic Automata patterns. ASAL relies on combinatorial search and, as a result, its running times and memory requirements scale exponentially with the dimensionality and length of the training sequences and with the size of the symbolic vocabulary (number of symbols or predicates) produced by the perceptual module. To alleviate this complexity we have extended ASAL to a version that uses differentiable pattern induction techniques. Given a fixed automaton topology (number of states and candidate transitions), we parameterize each transition guard as a neural logical expression over learned base predicates/symbols, implemented as differentiable conjunction and disjunction operators on top of a perceptual network. Given a sequence of inputs, the corresponding guard truth values are converted into a probability distribution over the SFA states' outgoing transition and a classical forward algorithm is used to compute the sequences' acceptance probabilities at the end of each sequence. The fuzzy relaxation and the fully differentiable forward pass through the SFA graph yield an objective that can be optimized end-to-end from sequence-level labels using standard gradient-based methods, without performing explicit symbolic search during training. Post-training, an extraction pipeline prunes small or redundant weights, sweeps thresholds to discretize the neural DNFs into candidate Boolean guards, and uses a lightweight ASP-based meta-encoding to select which guards and literals to retain, subject to structural constraints such as determinism and sparsity. In this way, ∂SFA recovers a compact, human-readable SFA that closely matches the behavior of the trained differentiable model, while training and extraction are significantly faster and more scalable than running ASAL from scratch on the same data.

Experiments on synthetic bechmarks, real-world datasets and EVENFLOW use case data showed that in this way we can learn high-quality, interpretable event patterns in a fraction of the time and memory required by symbolic search over noisy, weighted sequences prediction from perceptual sequences via perception networks.

5 Neurosymbolic Verification

In order to deploy NeSy systems in mission-critical applications, it is often necessary to have formal guarantees of their reliable performance. In EVENFLOW, we addressed the challenge of verifying the robustness of probabilistic NeSy systems [9], i.e., verifying the property that input perturbations do not affect the reasoning output. Techniques for NN verification are valuable to that end, since they can derive robustness guarantees for purely neural systems. In particular, relaxation-based techniques [5, 11] can scalably compute bounds for the NN outputs, with respect to input perturbations, which can then be used to assess robustness. Our work is focused on extending such techniques to the NeSy case by propagating these bounds through the probabilistic reasoning layer of a NeSy architecture. As such, we can then provide robustness guarantees for the entire system.

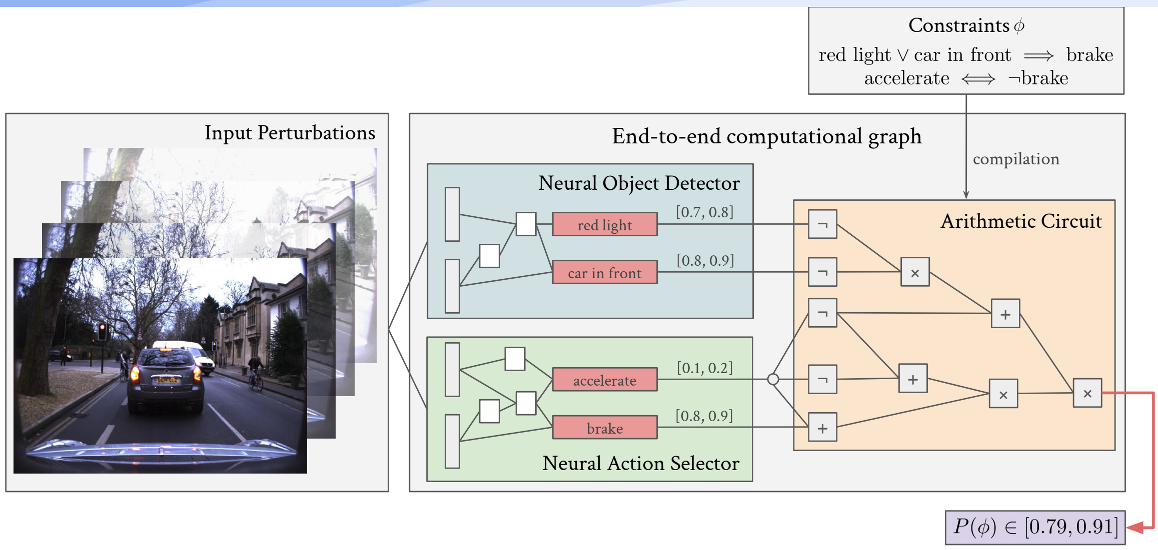


Figure 9: A motivating example for probabilistic NeSy verification. In this autonomous driving example we want to verify two logical constraints ϕ on top of two neural networks accepting the same dashcam image as input. The symbolic constraints are compiled into a tractable representation containing only addition, subtraction, and multiplication. During inference, this is used to reason over the NN outputs and calculate the probability that the constraints are satisfied. For verification, we exploit this structure to scalably compute how perturbations in the input affect the probabilistic output of the whole (NNs + reasoning) NeSy system.

In [9] we studied the complexity of solving the probabilistic NeSy verification task exactly and proposed an approximate solution, extending relaxation-based NN verification techniques to the NeSy setting. We showed how to compile the entire NeSy system into a single computational graph, which encapsulates both the neural and the symbolic components and is amenable to verification by off-the-shelf, state-of-the-art formal NN verifiers. Figure 9 provides an overview of the approach.

6 Conclusion

Starting from the limitations of purely symbolic CER/F systems and purely neural sequence models, EVENFLOW develops a coherent neuro-symbolic stack that spans perception, temporal abstraction, learning of interpretable structure, and formal verification. In this white paper we presented a brief overview of these techniques, focusing on neurosymbolic learning and reasoning, forecasting, event pattern learning and verification. Together, these techniques and the corresponding tools that were developed in the project allow for scalable training of robust and interpretable models in temporal domains. A unifying theme across these methods is **trustworthiness**, and in particular two of its main technical pillar, transparency and robustness. Transparency/interpretability is achieved thanks to the symbolic temporal knowledge and the formal probabilistic semantics of the EVENFLOW tools, that link low level neural predictions to higher-level reasoning inferences in a clear and transparent fashion. Robustness is pursued both via the regularization of neural components by symbolic knowledge, and more explicitly, via verification-based formal guarantees. In summary, EVENFLOW offers a suite of state-of-the-art neuro-symbolic tools for making sense of perceptual datastreams.

References

[1] Elias Alevizos, Alexander Artikis, and Georgios Paliouras. Complex event forecasting with prediction suffix trees. *The VLDB Journal*, 31(1):157–180, 2022.

[2] Tatiana Boura and Nikos Katzouris. Neurosymbolic complex event recognition in autonomous driving. In *Proceedings of the 1st International Workshop on Advanced Neurosymbolic Applications (ANSyA)*, 2025.

[3] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[4] Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[5] Rüdiger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In Deepak D'Souza and K. Narayan Kumar, editors, *Automated Technology for Verification and Analysis - 15th International Symposium, ATVA 2017, Pune, India, October 3-6, 2017, Proceedings*, volume 10482 of *Lecture Notes in Computer Science*, pages 269–286. Springer, 2017.

[6] Nikos Giatrakos, Elias Alevizos, Alexander Artikis, Antonios Deligiannakis, and Minos N. Garofalakis. Complex event recognition in the big data era: a survey. *VLDB J.*, 29(1):313–352, 2020.

[7] Nikos Katzouris and Georgios Paliouras. Answer set automata: A learnable pattern specification framework for complex event recognition. In *30th International Symposium on Temporal Representation and Reasoning (TIME 2023)*, pages 17–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023.

[8] Nikolaos Manginas, George Paliouras, and Luc De Raedt. NeSyA: Neurosymbolic automata. to appeat at IJCAI, 2025. Pre-print doi: 10.48550/arXiv.2412.07331.

[9] Vasileios Manginas, Nikolaos Manginas, Edward Stevinson, Sherwin Varghese, Nikos Katzouris, Georgios Paliouras, and Alessio Lomuscio. A scalable approach to probabilistic neuro-symbolic robustness verification. In *Proceedings of the 19th International Conference on Neurosymbolic Learning and reasoning*, 2025.

[10] Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, page 104062, 2024.

[11] Kaidi Xu, Zhouxing Shi, Huan Zhang, Minlie Huang, Kai-Wei Chang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis on general computational graphs. *CoRR*, abs/2002.12920, 2020.

This project is funded by the EU under Horizon Europe Grant Agreement No 101070430.
Views and opinions expressed are those of the EVENFLOW consortium only and
do not necessarily reflect those of the EU or the European Commission, neither the EU
nor the granting authority can be held responsible for them.