Neuro-Symbolic Learning for Predictive Process Monitoring via Two-Stage Logic Tensor Networks with Rule Pruning

Fabrizio De Santis[1]✉, Gyunam Park[2], and Francesco Zanichelli[1]

[1] University of Parma, Parma, Italy
{fabrizio.desantis,francesco.zanichelli}@unipr.it
[2] Eindhoven University of Technology, Eindhoven, Netherlands
g.park@tue.nl

Abstract. Predictive modeling on sequential event data is critical for fraud detection and healthcare monitoring. Existing data-driven approaches learn correlations from historical data but fail to incorporate domain-specific sequential constraints and logical rules governing event relationships, limiting accuracy and regulatory compliance. For example, healthcare procedures must follow specific sequences, and financial transactions must adhere to compliance rules. We present a neuro-symbolic approach integrating domain knowledge as differentiable logical constraints using Logic Networks (LTNs). We formalize control-flow, temporal, and payload knowledge using Linear Temporal Logic and first-order logic. Our key contribution is a two-stage optimization strategy addressing LTNs' tendency to satisfy logical formulas at the expense of predictive accuracy. The approach uses weighted axiom loss during pretraining to prioritize data learning, followed by rule pruning that retains only consistent, contributive axioms based on satisfaction dynamics. Evaluation on four real-world event logs shows that domain knowledge injection significantly improves predictive performance, with the two-stage optimization proving essential knowledge (without it, knowledge can severely degrade performance). The approach excels particularly in compliance-constrained scenarios with limited compliant training examples, achieving superior performance compared to purely data-driven baselines while ensuring adherence to domain constraints.

Keywords: Predictive process monitoring, Neuro-symbolic AI, Logic Tensor Networks, Deep learning.

1 Introduction

Modern organizations across finance, healthcare, and manufacturing routinely collect vast amounts of event data that are timestamped, attribute-rich records that capture the execution of business processes [14]. Predictive process monitoring has emerged as a critical capability for these organizations, enabling them to anticipate process outcomes, detect anomalies, and optimize operational efficiency by analyzing historical event sequences [11,20].

Recent advances in machine learning have improved predictive process monitoring capabilities. LSTM networks [20,9] and Transformer architectures [4] have shown superior performance in capturing complex sequential patterns and long-range dependencies

in process data. Advanced approaches like bidirectional LSTM [6,22] and topological deep learning methods [14] have further pushed the boundaries by incorporating richer structural representations and higher-order relationships between process events.

However, purely data-driven approaches face a fundamental limitation: they struggle to incorporate explicit domain constraints and business rules that are critical for real-world deployment [23,10]. In many domains, these rules represent mandatory compliance requirements or safety constraints that cannot be violated, even if historical data suggests otherwise. For example, in healthcare, "surgery can only be scheduled if a patient was discharged more than seven days ago," or in banking, "no transfer to a new beneficiary can occur before identity verification is complete." Such constraints may be rarely violated in training data yet remain absolutely critical during live operation. This disconnect between learned patterns and domain requirements highlights a key gap in current predictive process monitoring approaches.

Neuro-symbolic artificial intelligence offers a promising solution by integrating neural networks' pattern recognition capabilities with symbolic reasoning's explicit knowledge representation [3,1]. This paradigm enables models to leverage both statistical patterns from data and formal logical constraints, ensuring predictions that are both accurate and compliant with domain requirements. Recent frameworks like Logic Tensor Networks [1], semantic loss functions [23], and constraint-aware training methods [10] have demonstrated the potential of neuro-symbolic approaches across various domains, yet their systematic application to predictive process monitoring remains largely unexplored.

This paper presents a comprehensive neuro-symbolic framework for predictive process monitoring that systematically integrates domain knowledge with neural learning. Our approach addresses three key challenges: (1) how to formalize diverse types of process constraints using logical representations, (2) how to handle conflicts between domain rules and historical data patterns, and (3) how to balance the influence of data and knowledge constraints while maintaining rule reliability during training. A critical difficulty is that not all logical rules are equally useful—some may be redundant or even harmful to the prediction task, leading to degraded performance if treated uniformly.

To address these challenges, we propose a two-stage neuro-symbolic learning strategy. First, a weighted axiom loss stabilizes the interplay between data-driven and logic-driven objectives during pretraining, preventing unreliable rules from dominating the optimization. Second, a rule pruning mechanism filters out non-contributive rules based on satisfaction dynamics, producing a refined knowledge base for fine-tuning. This approach leverages the Logic Tensor Network framework [1] to create a unified neuro-symbolic classifier that maintains high predictive accuracy while ensuring adherence to domain constraints.

Our key contributions are:

1. To introduce a principled approach for extracting and categorizing process knowledge into control-flow, temporal, and payload features, formalizing them using Linear Temporal Logic (LTL) and first-order logic for integration into neural predictive models.
2. To propose a novel training strategy combining weighted axiom loss and rule pruning that addresses a fundamental limitation of Logic Tensor Networks (i.e., their tendency to satisfy logical formulas at the expense of predictive accuracy), resulting in more stable training and superior performance even with limited compliant examples.

The paper is structured as follows. Sect. 2 reviews related work in predictive process monitoring and neuro-symbolic AI. Sect. 3 introduces preliminary concepts. Sect. 4 details our knowledge formalization and rule pruning mechanism. Sect. 5 presents comprehensive experimental results, and Sect. 6 concludes with future directions.

2 Related Work

Predictive process monitoring (PPM) aims to forecast process outcomes and future events from event logs. Deep learning approaches have significantly advanced this field, with LSTM networks [20,9], Transformer architectures [4], bidirectional models [6,22], and topological approaches [14] demonstrating strong performance in capturing complex sequential patterns.

However, most approaches remain purely data-driven, ignoring domain constraints crucial for compliance-critical settings. Few studies have integrated explicit process knowledge: Vazifehdoostirani et al. [21] encode control-flow patterns as static input features, while Di Francescomarino et al. [12] enforce LTL constraints as hard post-processing rules. Mezini et al. [18] incorporate LTL constraints in the training process using a differentiable loss function to address the problem of suffix prediction.

These approaches have three key limitations: (1) focusing primarily on control-flow constraints while neglecting temporal and payload knowledge, (2) injecting knowledge either at input or output but not throughout the learning process, and (3) lacking mechanisms to manage conflicting or noisy rules.

Neuro-symbolic AI addresses these limitations by integrating neural learning with symbolic reasoning [3]. Several frameworks have emerged for this purpose. Semantic loss functions [23] and constraint-aware training methods [10] translate logical formulas into differentiable loss terms. Gradient-based approaches like RL-Net [7] learn interpretable ordered rules by combining neural network optimization with symbolic rule representation. Systems like δILP [8] and DeepProbLog [17] combine logic programming with deep learning. In the temporal domain, methods like TempRule [2] demonstrate the effectiveness of learning symbolic rules on knowledge graphs. Knowledge-infusion techniques [25,24,15] fuse knowledge into neural architectures through attention mechanisms and learnable constraints.

Among these frameworks, Logic Tensor Networks (LTNs) [1] are particularly well-suited for predictive process monitoring. Unlike approaches that require manual translation of logical constraints into loss terms or architectural modifications, LTNs provide a unified framework that directly embeds first-order logic into differentiable computation graphs using fuzzy semantics. This enables three key advantages for our domain: (1) natural representation of diverse process constraints, such as control-flow patterns expressed in LTL, temporal constraints on durations and waiting times, and payload-based business rules, within a single formalism; (2) soft constraint satisfaction through fuzzy logic semantics, which gracefully handles partial compliance and conflicting rules common in real-world process data; and (3) seamless integration of reasoning and learning, where logical formulas actively guide neural optimization rather than serving as post-hoc corrections.

Building on the LTN framework, our work makes two key contributions. First, we demonstrate how to systematically formalize complex business rules and temporal constraints from process mining using FOL and LTL for integration into neural predictive models. Second, we introduce a two-stage optimization strategy that addresses a fundamental limitation of LTNs (i.e., their tendency to prioritize logical consistency over predictive accuracy) through a weighted axiom loss during pretraining and a rule pruning mechanism that filters out inconsistent or non-contributive rules, resulting in more stable and interpretable training dynamics.

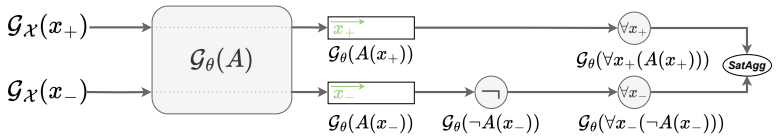

Fig. 1: Binary classification using LTN: loan applications (constants $\mathcal{C}$) are evaluated by predicate A (neural network) to determine approval. Variables x_+ and x_- represent positive and negative classes, aggregated by universal quantifiers $\forall$ to compute satisfaction.

3 Preliminaries

3.1 Logic Tensor Networks

Logic Tensor Networks (LTNs) [1] implement Real Logic, extending First-Order Logic (FOL) with fuzzy semantics where truth values are continuous in [0,1]. This enables differentiable integration of logical constraints with neural learning.

Real Logic Components. LTN comprises: (1) *constants* $\mathcal{C}$ representing individuals (e.g., loan applications), (2) *variables* $\mathcal{X}$ as typed placeholders (e.g., x_+ for approved applications), (3) *functions* $\mathcal{F}$ computing derived features, (4) *predicates* $\mathcal{P}$ expressing properties (e.g., A for "approved"), and (5) logical connectives $\{\neg,\wedge,\vee,\rightarrow,\leftrightarrow\}$ and quantifiers $\{\forall,\exists\}$.

Grounding and Differentiability. Each logic element is *grounded* to real tensors via function $\mathcal{G}$. Crucially, predicates are implemented as neural networks outputting truth values in [0,1], enabling gradient-based learning. Quantifiers are realized through differentiable aggregators:

$$\forall x\colon \text{pMeanError}(u_1,...,u_n) = 1 - \left(\frac{1}{n}\sum_{i=1}^{n}(1-u_i)^p\right)^{\frac{1}{p}},\ \exists x\colon \text{pMean}(u_1,...,u_n) = \left(\frac{1}{n}\sum_{i=1}^{n}u_i^p\right)^{\frac{1}{p}}$$

where $p \geq 1$ controls strictness. For binary classification (illustrated in Fig. 1), the knowledge base contains: $\forall x_+ A(x_+)$ (all positive examples satisfy A) and $\forall x_- \neg A(x_-)$ (all negative examples falsify A). The neural predicate A learns to output values near 1 for approved applications and near 0 for rejected ones.

Learning Objective. LTN minimizes: $L = 1 - \underset{\phi\in\mathcal{K}}{\text{SatAgg}}\mathcal{G}_\theta(\phi)$ where $\mathcal{K}$ is the knowledge base, SatAgg aggregates satisfaction degrees across axioms (using pMeanError), and θ denotes neural parameters. This objective ensures predictions satisfy both data labels and logical constraints through continuous optimization.

3.2 Predictive Process Monitoring

An *event* $e = (a,c,t,attr_1,...,attr_n)$ comprises activity a, case identifier c, timestamp t, and attributes. A *trace* $\sigma = \langle e_1,...,e_n\rangle$ is an ordered sequence with identical case identifiers and non-decreasing timestamps. A *trace prefix* $l = \langle e_1,...,e_k\rangle$ represents the first k events of a trace.

Let $\mathcal{L}$ denote all prefixes, with $\mathcal{L}_+$ and $\mathcal{L}_-$ representing positive and negative classes. A *predictive model* $\Delta\colon \mathcal{L} \rightarrow \{+,-\}$ classifies prefixes into outcome classes.

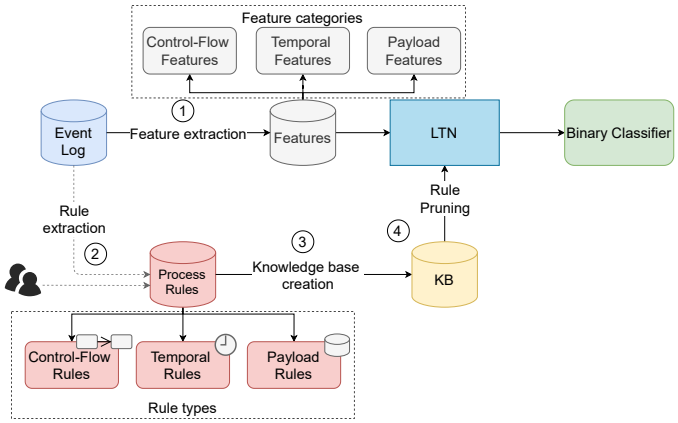

Fig. 2: The pipeline followed in the approach, which consists of feature extraction, rule extraction, knowledge base creation and injection, rule pruning, and then the creation of the neuro-symbolic model leveraging the LTN framework.

LTN Mapping. Process monitoring naturally maps to Real Logic: variables $\mathcal{X}$ correspond to trace prefixes, constants $\mathcal{C}$ to activities and attribute values, functions $\mathcal{F}$ to temporal computations (e.g., time elapsed between activities) and aggregations (e.g., patient age, avg. requested amount), and predicates $\mathcal{P}$ to predictive models and constraints. For example, the rule "antibiotics not administered within two hours after surgery for elderly patients leads to complications" formalizes as:

$$\forall l((WaitTime(l,\texttt{Surg},\texttt{ATB}) > 2 \wedge Age(l) > 60) \rightarrow P(l))$$

where $WaitTime, Age \in \mathcal{F}$ and $P \in \mathcal{P}$ models complication risk.

4 Neuro-Symbolic AI for Predictive Process Monitoring

Our approach integrates domain knowledge into predictive process monitoring through the pipeline in Fig. 2. We extract features and rules from event logs, construct a knowledge base in FOL, and inject it into an LTN-based binary classifier.

Consider predicting post-surgical complications in a healthcare process with activities: medical history review (`Rev`), physical examination (`Exam`), laboratory tests (`Lab`), antibiotic administration (`ATB`), surgery (`Surg`), pain management (`PAdm`), and post-operative check-up (`PostCU`). We use predicate $P(l)$ to model complication probability for prefix l, with l_+ and l_- denoting positive and negative classes.

4.1 Feature Extraction

We extract three feature categories that define the vocabulary for rule formalization:

- *Control-flow features*: Sequential structures (precedence, repetition, branching). Example: $IsNext(l,\texttt{Rev},\texttt{Exam})$.
- *Temporal features*: Time-based relationships (durations, waiting times, cycle times). Example: $WaitTime(l,\texttt{Surg},\texttt{ATB})$.

– *Payload features*: Case-level (static attributes) and event-level (activity-specific attributes) contextual data.

4.2 Rule Extraction

Rules are extracted automatically via process mining techniques or manually by domain experts. We extract three rule types aligned with the feature space:

Control-flow rules. Sequential execution constraints represented using Linear Temporal Logic (LTL). We employ Declare mining [16] to identify constraints. Examples:
– $\Box(\texttt{Rev} \Longrightarrow \Diamond\texttt{Exam})$: if `Rev` occurs, `Exam` must follow
– $\Box(\texttt{Surg} \Longrightarrow \bigcirc\texttt{PostCU})$: `PostCU` immediately follows `Surg`

Temporal rules. Timing constraints extracted from SLA compliance analysis [19], expressed as IF-THEN statements. Example: "if `ATB` occurs within two hours after `Surg`, complications decrease."

Payload rules. Statistical correlations [5] between attributes and outcomes, expressed as IF-THEN statements. Example: "if oxygen saturation $<90\%$ post-surgery, complication risk increases."

4.3 Knowledge Base Construction

FOL Translation. We convert extracted rules to FOL:
– *Control-flow*: LTL-to-FOL conversion [13] preserves temporal semantics. Example: $\Box(\texttt{Rev} \Longrightarrow \Diamond\texttt{Exam})$ translates to $\forall l(HasAct(l,\texttt{Rev}) \wedge Next(l,\texttt{Rev},\texttt{Exam}))$.
– *Temporal/Payload*: IF-THEN rules decompose into antecedent-consequent formulas. Example: $\forall l_+(WaitTime(l_+,\texttt{Surg},\texttt{ATB}) \leq 2 \rightarrow \neg P(l_+))$.

Axiom Types. In our framework, we distinguish between two types of axioms in the knowledge base: data axioms $\mathcal{K}_\mathcal{D}$ and knowledge axioms $\mathcal{K}_\mathcal{P}$. The knowledge base comprises:
– *Data axioms* $\mathcal{K}_\mathcal{D}$: $\{\forall x_+ P(x_+), \forall x_- \neg P(x_-)\}$: Classification-related constraints
– *Knowledge axioms* $\mathcal{K}_\mathcal{P}$: Domain-derived logical rules

4.4 Two-Stage Optimization with Rule Pruning

To address LTNs' tendency to satisfy logical constraints at the expense of predictive accuracy, we propose a two-stage optimization strategy (Alg. 1).

Phase 1: Weighted Pretraining and Rule Selection. We train using weighted loss:

$$L = 1 - \left(\alpha \cdot \underset{\phi \in \mathcal{K}_\mathcal{D}}{\text{SatAgg}} \mathcal{G}_\theta(\phi) + \beta \cdot \underset{\phi \in \mathcal{K}_\mathcal{K}}{\text{SatAgg}} \mathcal{G}_\theta(\phi) \right) \tag{1}$$

Setting $\alpha > \beta$ prioritizes data learning, preventing vacuous satisfaction of knowledge axioms before the classification predicate is properly learned. After E_p epochs, we construct a refined knowledge base $\mathcal{K}'_\mathcal{P}$ by selecting rules via gating scores obtained on the validation set:

$$g_i = \bar{s_i} \cdot e^{-\lambda \cdot \text{Var}(s_i)}$$

where $\bar{s_i}$ and $\text{Var}(s_i)$ denote mean and variance of rule r_i's satisfaction across samples. Rules with $g_i \geq \tau$ are added to $\mathcal{K}'_\mathcal{K}$, retaining only consistent, contributive axioms.

Algorithm 1 Two-stage optimization with rule pruning for compliance-aware scenarios

Require: $\alpha+\beta=1$, thresholds τ, λ
Initialize θ, $\mathcal{K}_\mathcal{D}$, $\mathcal{K}_\mathcal{P}=\{r_1,...,r_n\}$ {Phase 1: Weighted pretraining and pruning}
for epoch $=1$ to E_p **do**
  $L \leftarrow 1-(\alpha\cdot\text{SatAgg}(\mathcal{K}_\mathcal{D})+\beta\cdot\text{SatAgg}(\mathcal{K}_\mathcal{P}))$
  Update θ via backpropagation
end for
Initialize $\mathcal{K}'_\mathcal{P} \leftarrow \emptyset$
for each $r_i \in \mathcal{K}_\mathcal{P}$ **do**
  Compute $g_i \leftarrow \bar{s}_i \cdot e^{-\lambda\cdot\text{Var}(s_i)}$
  if $g_i \geq \tau$ **then**
    Add r_i to $\mathcal{K}'_\mathcal{P}$
  end if
end for{Phase 2: Fine-tuning on pruned knowledge base}
for epoch $=1$ to E_f **do**
  $L \leftarrow 1-\text{SatAgg}(\mathcal{K}_\mathcal{D}\cup\mathcal{K}'_\mathcal{P})$
  Update θ via backpropagation
end for

Phase 2: Fine-tuning. The model is fine-tuned on the refined knowledge base $\mathcal{K}_\mathcal{D}\cup\mathcal{K}'_\mathcal{P}$ for E_f epochs using Eq. 1 (with $\mathcal{K}'_\mathcal{P}$ instead of $\mathcal{K}_\mathcal{P}$) in the compliance-aware scenario, while for the temporal plit scenario we use the standard LTN loss $L=1-\text{SatAgg}(\mathcal{K}_\mathcal{D}\cup\mathcal{K}'_\mathcal{P})$.

Hyperparameters. We set $\alpha=0.8$, $\beta=0.2$, $E_p=5$, $E_f=50$, $\lambda\in\{1,2\}$ based on dataset characteristics, and select τ via validation. Gating scores are computed analytically, avoiding learnable parameters and preserving interpretability.

5 Evaluation

We evaluate our approach[3] by addressing three research questions:
- **RQ1:** Does injecting domain knowledge as logical constraints improve predictive performance compared to purely data-driven approaches?
- **RQ2:** What is the contribution of the two-stage optimization with rule pruning mechanism to model performance?
- **RQ3:** How effectively does the approach generalize to compliance-constrained scenarios with limited compliant training examples?

5.1 Experimental Setup

Model Variants. To address RQ1 and RQ2, we compare:
- **LSTM [20], TFR [4]**: Purely data-driven baselines with binary cross-entropy loss (no logical components)
- **LTN-Data-L, LTN-Data-T**: LTN [1] with LSTM/Transformer backbones trained only on data axioms $\mathcal{K}_\mathcal{D}$ (no domain knowledge)

[3] The code to reproduce our experiments is available at `https://github.com/FabrizioDeSantis/NeSy-PPM`.

Table 1: RQ1: Performance on temporal split comparing purely data-driven baselines with knowledge-enhanced variants. Results show mean and standard deviation over 5 seeds.

Method	Sepsis		BPIC2012		BPIC2017		Traffic fines	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
LSTM	88.69±0.92	70.59±1.31	52.47±0.77	52.44±0.74	70.16±0.41	69.47±0.44	67.34±0.42	66.93±0.28
TRF	90.11±0.47	73.80±0.36	52.29±0.58	51.86±0.92	71.15±0.25	70.74±0.22	67.81±0.11	67.34±0.14
Two-Stage-L	89.39±0.47	**75.82±0.92**	60.33±0.86	57.25±0.38	70.69±0.89	70.67±0.89	**68.44±0.43**	**68.20±0.38**
Two-Stage-T	**91.30±0.11**	75.63±0.99	**65.27±0.27**	**62.12±0.39**	**72.30±0.11**	**72.16±0.16**	67.94±0.11	67.72±0.08

- **LTN-NoP-L, LTN-NoP-T**: LTN with data and knowledge axioms $\mathcal{K}_{\mathcal{D}} \cup \mathcal{K}_{\mathcal{P}}$ but without weighted loss or pruning
- **Two-Stage-L, Two-Stage-T**: Our complete approach with weighted axiom loss and rule pruning

Architecture and Training. LSTM: 2 layers, hidden size 128. Transformer: 2 attention heads, hidden size 128. Both use Adam optimizer (lr=0.001, batch size=32). Categorical features use embeddings with a dimension of 32; numerical features are standardized and passed through linear layers. We use simplified architectures to isolate knowledge integration effects, though our approach is architecture-agnostic. All results report mean and standard deviation over 5 random seeds.

Evaluation Protocols. We employ two evaluation settings:
- *Temporal split*: Standard 80-20 train-test split based on completion timestamps, evaluating predictive performance on future cases (addresses RQ1 and RQ2)
- *Compliance-aware split*: Test set enriched with rule-compliant traces (satisfying logical rules and labels) plus random samples, assessing generalization to compliance-constrained scenarios (addresses RQ3)

5.2 RQ1: Impact of Domain Knowledge Injection

To answer RQ1, we compare purely data-driven baselines (LSTM [20], TFR [4]) against knowledge-enhanced variants on the temporal split (Tab. 1).

Key Findings. Two-Stage-L and Two-Stage-T consistently outperform their purely neural counterparts across all datasets. Improvements are most pronounced on smaller datasets: *Sepsis* shows F1 improvements of 5.23% (LSTM→Two-Stage-L) and 1.83% (TFR→Two-Stage-T), while *BPIC2012* exhibits gains of 4.81% and 10.26% respectively. For larger datasets like *Traffic fines*, improvements are more modest (1.1% and 0.38%), as abundant training data reduces the relative contribution of logical rules.

Architecture comparison reveals no clear overall winner: LSTMs excel on short sequences (*Sepsis*, *Traffic fines*: max length 13 and 10), achieving 75.82% and 68.20% F1 with Two-Stage-L, while Transformers perform better on longer traces (*BPIC2012*, *BPIC2017*: max length 40 and 20), reaching 62.12% and 72.16% F1 with Two-Stage-T.

5.3 RQ2: Ablation Study on Two-Stage Optimization

To answer RQ2, we conduct an ablation study comparing: (1) LTN-Data (data axioms only), (2) LTN-NoP (knowledge axioms without pruning), and (3) Two-Stage (our complete method) on the temporal split (Tab. 2).

Table 2: RQ2: Ablation study comparing LTN variants on temporal split: Data-only (no knowledge), NoP (knowledge without pruning), and Two-Stage (complete method).

Method	Sepsis		BPIC2012		BPIC2017		Traffic fines	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
LTN-Data-L	82.56±4.52	69.10±4.22	54.24±0.63	53.18±0.63	69.73±0.07	69.71±0.07	67.59±0.60	67.31±0.56
LTN-Data-T	81.09±6.25	66.79±4.65	57.17±1.13	55.40±0.77	71.15±0.25	70.74±0.22	67.53±0.19	67.27±0.1
LTN-NoP-L	21.36±5.96	18.79±2.95	54.50±4.35	45.00±5.54	69.16±0.24	66.27±0.36	59.33±0.27	46.35±0.41
LTN-NoP-T	21.69±7.58	19.18±4.76	47.47±6.68	33.82±2.70	68.54±0.52	66.09±0.69	57.57±0.86	41.78±1.94
Two-Stage-L	89.39±0.47	**75.82±0.92**	60.33±0.86	57.25 ±0.38	70.69±0.89	70.67±0.89	**68.44±0.43**	**68.20±0.38**
Two-Stage-T	**91.30±0.11**	75.63±0.99	**65.27±0.27**	**62.12±0.39**	**72.30±0.11**	**72.16±0.16**	67.94±0.11	67.72±0.08

Table 3: RQ3: Performance on compliance-aware test set. Models must generalize to scenarios emphasizing rule-compliant behavior with limited compliant training examples.

Method	Sepsis		BPIC2012		BPIC2017		Traffic fines	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
LSTM	81.50±2.41	73.26±3.28	53.89±1.59	53.71±1.49	65.71±1.04	65.41±1.19	77.47±0.21	76.92±0.33
TRF	82.74±1.39	74.66±1.77	56.16±0.79	55.61±0.68	68.77±0.83	68.54±0.96	77.39±0.48	76.85±0.45
LTN-NoP-L	39.91±4.83	37.52±6.51	47.58±5.82	34.61±4.53	65.28±0.05	64.48±0.02	64.93±0.05	42.39±0.11
LTN-NoP-T	34.55±4.10	30.38±5.47	63.61±0.74	53.39±1.00	62.02±6.69	60.68±7.77	65.15±0.09	42.74±0.22
Two-Stage-L	**91.47±2.82**	**89.96±2.96**	62.88±0.80	60.07±0.49	75.53±0.35	74.66±0.22	78.36±0.52	77.76±0.59
Two-Stage-T	89.21±0.88	87.36±0.89	**68.17±0.14**	**64.23±0.06**	**77.16±0.24**	**75.47±0.03**	**78.86±0.26**	**78.44±0.13**

Key Findings. The results reveal a critical insight: adding knowledge axioms without proper control (LTN-NoP) severely degrades performance, particularly on small datasets. On *Sepsis*, LTN-NoP-L and LTN-NoP-T achieve only 18.79% and 19.18% F1, substantially worse than even purely data-driven baselines. This occurs because LTNs optimize overall knowledge base satisfaction, allowing models to exploit logical shortcuts (vacuous satisfaction) rather than learning meaningful predictive patterns.

Our two-stage approach addresses this fundamental limitation. On *Sepsis*, Two-Stage-L recovers to 70.59% F1 (51.80% improvement over LTN-NoP-L). Similar patterns emerge across all datasets: *BPIC2012* shows 28.12% improvement, *BPIC2017* shows 5.78% improvement, and *Traffic fines* shows 25.94% improvement.

Comparing LTN-Data to Two-Stage isolates the net benefit of domain knowledge when properly integrated. Two-Stage consistently outperforms LTN-Data: improvements range from 10.26% on *BPIC2012* to 8.84% on *Sepsis*, demonstrating that the two-stage mechanism successfully harnesses knowledge benefits while avoiding pitfalls.

5.4 RQ3: Generalization to Compliance Constraints

To answer RQ3, we evaluate on the compliance-aware test set that combines rule-compliant traces with random samples (Tab. 3). This setting is particularly challenging because: (1) compliant examples in training are scarce (4% in *Sepsis*, 18.25% in *BPIC2012*), and (2) the test set is enriched with compliant cases (32.3% and 61.9% respectively), requiring models to generalize beyond the training distribution.

Key Findings. Two-Stage variants substantially outperform baselines on this evaluation. On *Sepsis* with only 4% compliant training examples, Two-Stage-L achieves 89.96% F1 compared to 73.26% for LSTM (16.7% improvement). The gap widens further compared to LTN-NoP-L (52.44%), demonstrating that naive knowledge injection fails catastrophically in compliance-constrained settings.

Sepsis (4% compliant) shows the largest improvement (16.7%), followed by *BPIC2012* (18.25%, 10.52%), *BPIC2017* (62.7%, 10.06%), and *Traffic fines* (39.67%, 1.52%). This

trend confirms that logical rules provide crucial inductive bias when compliant examples are scarce. Notably, all purely data-driven approaches (LSTM, TFR, LTN-Data) exhibit similar performance, unable to extrapolate compliance patterns from limited examples.

5.5 Discussion

Our evaluation provides several insights. For RQ1, domain knowledge injection consistently improves performance, with gains varying across datasets. The improvements are most pronounced under limited training data, indicating that knowledge can constrain model learning when statistical evidence is insufficient. In data-rich settings, where patterns are already well captured, the marginal benefit naturally diminishes. The results for RQ2 emphasize that knowledge integration is not inherently beneficial. Simple incorporation can compromise the optimization process and significantly degrade performance. The proposed two-stage optimization is therefore essential, as it stabilizes training by balancing statistical objectives and logical constraints, highlighting the non-trivial interaction between symbolic knowledge and data-driven learning. Finally, for RQ3, the approach is particularly effective in compliance-constrained scenarios with scarce compliant examples. While purely data-driven models tend to overlook rare but semantically critical behaviors, explicit encoding preserves relevant patterns, addressing a key limitation of conventional predictive process monitoring.

6 Conclusion

We presented a neuro-symbolic approach for predictive process monitoring that systematically integrates domain knowledge as differentiable logical constraints using Logic Tensor Networks. Our key contribution is a two-stage optimization strategy that addresses a fundamental limitation of LTNs: their tendency to satisfy logical formulas at the expense of predictive accuracy. The approach combines weighted axiom loss during pretraining to prioritize data learning, followed by rule pruning that retains only consistent, contributive axioms based on satisfaction dynamics.

Our evaluation on four real-world event logs answered three research questions. First, domain knowledge injection improves predictive performance, especially with limited training data (RQ1). Second, the two-stage optimization is essential—without it, knowledge injection can severely degrade performance through vacuous satisfaction of logical constraints (RQ2). Third, the approach effectively generalizes to compliance-constrained scenarios with scarce compliant training examples (RQ3), addressing a critical gap in current predictive process monitoring methods.

In future work, we plan to formalize an explicit representation defining specific templates to automatically connect features with rules. Moreover, we aim to evaluate our neuro-symbolic approach on a case study where domain knowledge plays a critical role, such as in healthcare. Finally, we plan to extend the approach to more process-aware tasks, such as predictions related to next events and time-related aspects.

Acknowledgements

F. De Santis is supported by the Italian Ministry of University and Research (MUR) under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 1, Investment 4.1, CUP D91I23000080006, funded by the European Union - NextGenerationEU.

References

1. Badreddine, S., d'Avila Garcez, A.S., Serafini, L., Spranger, M.: Logic tensor networks. Artif. Intell. **303**, 103649 (2022)
2. Bao, X., Wang, K., Wang, Z., Wu, H., Zuo, J., Zhang, X., Feng, Z., Wu, H.: Explainable temporal knowledge graph reasoning via expressive logic rules. In: Wu, X., Spiliopoulou, M., Wang, C., Kumar, V., Cao, L., Wu, Y., Yao, Y., Wu, Z. (eds.) Advances in Knowledge Discovery and Data Mining - 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2025, Sydney, NSW, Australia, June 10-13, 2025, Proceedings, Part I. Lecture Notes in Computer Science, vol. 15870, pp. 175–186. Springer (2025)
3. Bhuyan, B.P., Ramdane-Cherif, A., Tomar, R., Singh, T.P.: Neuro-symbolic artificial intelligence: a survey. Neural Comput. Appl. **36**(21), 12809–12844 (2024)
4. Bukhsh, Z.A., Saeed, A., Dijkman, R.M.: Processtransformer: Predictive business process monitoring with transformer network. CoRR **abs/2104.00721** (2021)
5. Cafaro, M., Epicoco, I., Pulimeno, M.: Data mining: Mining frequent patterns, associations rules, and correlations. In: Ranganathan, S., Gribskov, M., Nakai, K., Schönbach, C. (eds.) Encyclopedia of Bioinformatics and Computational Biology - Volume 1, pp. 358–366. Elsevier (2019)
6. Camargo, M., Dumas, M., Rojas, O.G.: Learning accurate LSTM models of business processes. In: Hildebrandt, T.T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) Business Process Management - 17th International Conference, BPM 2019, Vienna, Austria, September 1-6, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11675, pp. 286–302. Springer (2019)
7. Dierckx, L., Veroneze, R., Nijssen, S.: Rl-net: Interpretable rule learning with neural networks. In: Kashima, H., Idé, T., Peng, W. (eds.) Advances in Knowledge Discovery and Data Mining - 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25-28, 2023, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13935, pp. 95–107. Springer (2023)
8. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. J. Artif. Intell. Res. **61**, 1–64 (2018)
9. Evermann, J., Rehse, J., Fettke, P.: Predicting process behaviour using deep learning. Decis. Support Syst. **100**, 129–140 (2017)
10. Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., Vechev, M.T.: DL2: training and querying neural networks with logic. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 1931–1941. PMLR (2019)
11. Francescomarino, C.D., Ghidini, C.: Predictive process monitoring. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook, Lecture Notes in Business Information Processing, vol. 448, pp. 320–346. Springer (2022)
12. Francescomarino, C.D., Ghidini, C., Maggi, F.M., Petrucci, G., Yeshchenko, A.: An eye into the future: Leveraging a-priori knowledge in predictive business process monitoring. In: Carmona, J., Engels, G., Kumar, A. (eds.) Business Process Management - 15th International Conference, BPM 2017, Barcelona, Spain, September 10-15, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10445, pp. 252–268. Springer (2017)
13. Giacomo, G.D., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Rossi, F. (ed.) IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. pp. 854–860. IJCAI/AAAI (2013)
14. Huang, L., Xia, K., Hu, C.: Path complex neural networks for sequential process activities classification. In: Sun, Y., Chierichetti, F., Lauw, H.W., Perlich, C., Tok, W.H., Tomkins, A. (eds.) Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025. pp. 544–554. ACM (2025)

15. Li, B., Wang, B., Li, L., Li, W., Mo, T.: Equitynet: Unveiling corporate equity relationships in business conglomerates using graph neural networks and GDV features. In: Wu, X., Spiliopoulou, M., Wang, C., Kumar, V., Cao, L., Wu, Y., Yao, Y., Wu, Z. (eds.) Advances in Knowledge Discovery and Data Mining - 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2025, Sydney, NSW, Australia, June 10-13, 2025, Proceedings, Part I. Lecture Notes in Computer Science, vol. 15870, pp. 137–148. Springer (2025)
16. Maggi, F.M., Mooij, A.J., van der Aalst, W.M.P.: User-guided discovery of declarative process models. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence 2011, April 11-15, 2011, Paris, France. pp. 192–199. IEEE (2011)
17. Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., Raedt, L.D.: Neural probabilistic logic programming in deepproblog. Artif. Intell. **298**, 103504 (2021)
18. Mezini, A., Umili, E., Donadello, I., Maggi, F.M., Mancanelli, M., Patrizi, F.: Neuro-symbolic predictive process monitoring. CoRR **abs/2509.00834** (2025)
19. Taghiabadi, E.R., Fahland, D., van Dongen, B.F., van der Aalst, W.M.P.: Diagnostic information for compliance checking of temporal compliance requirements. In: Salinesi, C., Norrie, M.C., Pastor, O. (eds.) Advanced Information Systems Engineering - 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7908, pp. 304–320. Springer (2013)
20. Tax, N., Verenich, I., Rosa, M.L., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10253, pp. 477–492. Springer (2017)
21. Vazifehdoostirani, M., Genga, L., Dijkman, R.M.: Encoding high-level control-flow construct information for process outcome prediction. In: Burattin, A., Polyvyanyy, A., Weber, B. (eds.) 4th International Conference on Process Mining, ICPM 2022, Bolzano, Italy, October 23-28, 2022. pp. 48–55. IEEE (2022)
22. Wang, J., Yu, D., Liu, C., Sun, X.: Outcome-oriented predictive process monitoring with attention-based bidirectional LSTM neural networks. In: Bertino, E., Chang, C.K., Chen, P., Damiani, E., Goul, M., Oyama, K. (eds.) 2019 IEEE International Conference on Web Services, ICWS 2019, Milan, Italy, July 8-13, 2019. pp. 360–367. IEEE (2019)
23. Xu, J., Zhang, Z., Friedman, T., Liang, Y., den Broeck, G.V.: A semantic loss function for deep learning with symbolic knowledge. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 5498–5507. PMLR (2018)
24. Ying, X., Luo, S., Yu, M., Zhao, M., Yu, J., Guo, J., Li, X.: Two-stage knowledge graph completion based on semantic features and high-order structural features. In: Yang, D., Xie, X., Tseng, V.S., Pei, J., Huang, J., Lin, J.C. (eds.) Advances in Knowledge Discovery and Data Mining - 28th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2024, Taipei, Taiwan, May 7-10, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14645, pp. 143–155. Springer (2024)
25. Zhao, M., Jia, W., Huang, Y.: Attention-based aggregation graph networks for knowledge graph information transfer. In: Lauw, H.W., Wong, R.C., Ntoulas, A., Lim, E., Ng, S., Pan, S.J. (eds.) Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12085, pp. 542–554. Springer (2020)