Confidence is not Timeless: Modeling Temporal Validity for Rule-based Temporal Knowledge Graph Forecasting

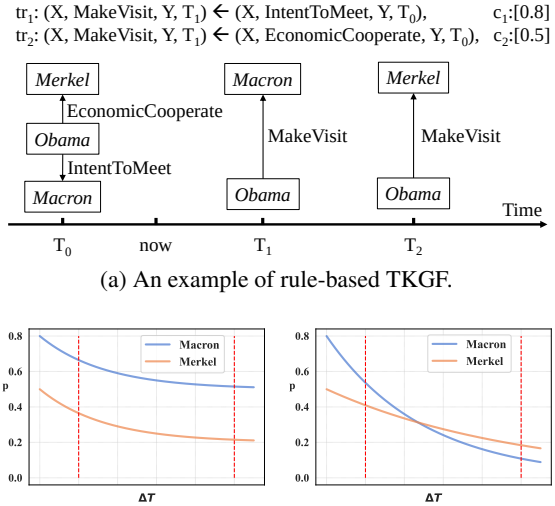
Rikui Huang[1,2,3], Wei Wei[*1], Xiaoye Qu[1,5], Shengzhe Zhang[1] , Dangyang Chen[4] , Yu Cheng[6]
[1]School of Computer Science & Technology, Huazhong University of Science and Technology
[2]School of Artificial Intelligence & Automation, Huazhong University of Science and Technology
[3]Institute of Artificial Intelligence, Huazhong University of Science and Technology
[4]Ping An Property & Casualty Insurance company of China
[5]Shanghai AI Laboratory, [6]The Chinese University of Hong Kong
{huangrk, weiw, xiaoye, zshengz}@hust.edu.cn
chengyu@cse.cuhk.edu.hk, chendangyang273@pingan.com.cn

Abstract

Recently, Temporal Knowledge Graph Forecasting (TKGF) has emerged as a pivotal domain for forecasting future events. Unlike black-box neural network methods, rule-based approaches are lauded for their efficiency and interpretability. For this line of work, it is crucial to correctly estimate the predictive effectiveness of the rules, i.e., the confidence. However, the existing literature lacks in-depth investigation into how confidence evolves with time. Moreover, inaccurate and heuristic confidence estimation limits the performance of rule-based methods. To alleviate such issues, we propose a framework named **TempValid** to explicitly model the temporal validity of rules for TKGF. Specifically, we design a time function to model the interaction between temporal information with confidence. TempValid conceptualizes confidence and other coefficients as learnable parameters to avoid inaccurate estimation and combinatorial explosion. Furthermore, we introduce a *rule-adversarial negative sampling* and a *time-aware negative sampling* strategies to facilitate TempValid learning. Extensive experiments show that TempValid significantly outperforms previous state-of-the-art (SOTA) rule-based methods on six TKGF datasets. Moreover, it exhibits substantial advancements in cross-domain and resource-constrained rule learning scenarios.

1 Introduction

Representing, acquiring, and applying knowledge have always been hot topics in the field of artificial intelligence research (Cai et al., 2020; Ji et al., 2021; Mialon et al., 2023). Knowledge reasoning (Huang et al., 2023, 2024) is an invaluable pathway for humanity to understand and acquire new knowledge. Previous knowledge reasoning focused on static reasoning, however, temporal reasoning holds more profound implications for daily life. For example, if dark clouds are observed, one could take an umbrella to avoid catching a cold caused by the probable heavy rain. Temporal Knowledge Graph Forecasting (TKGF) is a natural scenario for predicting future events based on historical data, thus drawing substantial interest from researchers in recent years (Trivedi et al., 2017; Jin et al., 2020; Wang et al., 2023).



(a) An example of rule-based TKGF.

(b) Uniform temporal pattern. (c) Rule-independent pattern.

Figure 1: An example of rule-based TKGF with different temporal patterns. Employing uniform temporal patterns will yield a temporally insensitive outcome.

Compared to the neural methods obsessed with non-transparent latent vector representations, rule-based approaches are capable of discerning and implementing subtle rules for efficient and interpretable reasoning (Galárraga et al., 2013; Meilicke et al., 2019; Liu et al., 2022). However, previous efforts focused on estimating the static pre-

[*] Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10783–10794

dictive capability of rules (Galárraga et al., 2015; Pellissier Tanon et al., 2017; Ortona et al., 2018), i.e., confidence, overlooking their interaction with temporal information, which may result in temporally insensitive outcomes. As shown in Figure 1a, there are two temporal rules with different confidence, if we employ static or uniform temporal patterns in Figure 1b, then it will yield the outcome *Macron* at all future points in time. To predict the example in Figure 1a, it is necessary to model rule-independent temporal characteristics shown in Figure 1c to get richer temporal patterns.

In addition to the interaction of temporal information with rules, the interaction between rules is also important for reasoning. Existing methods perform confidence estimation and aggregating reasoning based on rule-independence assumptions (Meilicke et al., 2019; Liu et al., 2022; Li et al., 2023), which may lead to an overestimation of incorrect outcomes (Ott et al., 2021). However, delving into the interactions between rules may encounter potential combinatorial explosion of confidence (Betz et al., 2023). While parameterized confidence may alleviate this challenge (Ott et al., 2023), it remains an unsolved problem to model both the interaction of temporal information with rules and the interaction between temporal rules.

To address the aforementioned challenges, we propose a framework, named **TempValid**, which models the temporal validity of rules for TKGF. Specifically, TempValid assumes that confidence decays over time rather than being timeless. We design a time function with a learnable decay coefficient to regulate the decay rate impacting rule confidence. The confidence and decay coefficients are conceptualized as learnable parameters to avoid inaccurate heuristic estimation and potential combinatorial explosion. To effectively optimizing TempValid, we design a *rule-adversarial negative sampling* and a *time-aware negative sampling* strategies. Extensive experiments show that TempValid significantly outperforms existing rule-based TKGF methods and achieves competitive performance with the state-of-the-art baselines on six TKGF datasets. Moreover, TempValid significantly outperforms existing approaches in cross-dataset and low-resource scenarios.

In summary, our contributions are as follows: (1) We propose a framework named TempValid for modeling the temporal validity of temporal rules. To our best knowledge, this is the first study of investigating the temporal validity of temporal rules in the context of temporal knowledge graphs. (2) We design a time function with controlled decay rate, conceptualize confidence and decay coefficients as learnable parameters, and propose two elaborate negative sampling strategies to facilitate TempValid training. (3) Our proposed TempValid significantly surpasses the existing rule-based TKGF methods and achieves competitive performance with the state-of-the-art baselines on six representative TKGF datasets, including ICEWS14, ICEWS18, ICEWS05-15, YAGO, WIKI and GDELT.

2 Related Work

2.1 Temporal Knowledge Graph Forecasting

Temporal knowledge graph reasoning can be classified into two settings: interpolation and extrapolation settings. Interpolation setting aims to predict missing historical facts within a known time frame (Leblay and Chekol, 2018; Garcia-Duran et al., 2018; Goel et al., 2020). However, these methods cannot predict future facts effectively.

To solve the deficiency of interpolation setting, extrapolation setting is proposed, i.e, TKGF. RENET (Jin et al., 2020) and REGCN (Li et al., 2021b) treat TKGs as sequences of subgraphs, employing GNNs to model graph structures and RNNs to capture sequential patterns, respectively. Inspired by the repetition of historical events (Trompf, 1979; Schlesinger, 1999), CyGNet (Zhu et al., 2021) and CENET (Xu et al., 2023) utilize copy mechanism to predict future facts. TiRGN (Li et al., 2022) considers the sequential, repetitive and cyclical patterns of historical facts simultaneously, achieving a remarkable performance.

However, due to the nature of black-box models, the above methods lack transparency and interpretability. To address this issue, xERTE (Han et al., 2020) generates interpretable reasoning paths through subgraph expansion and pruning. CluSTeR (Li et al., 2021a) and TITer (Sun et al., 2021) use reinforcement learning strategies to travel relational paths for reasoning. TECHS (Lin et al., 2023) attempt to encode the rules mined by TLogic and perform predictions using GNNs. The most relevant to our work is TLogic (Liu et al., 2022), which is a extension of AnyBURL (Meilicke et al., 2019), a rule-based reasoning method for static knowledge graphs. It designs the temporal random walk to mine temporal rules and aggregates temporal rules for reasoning. Compared to TLogic,

our proposed TempValid focuses on modeling the temporal validity of rules.

2.2 Estimation of Confidence

Although various metrics (Galárraga et al., 2013; Zhang et al., 2021) and assumptions (Galárraga et al., 2015; Pellissier Tanon et al., 2017; Ortona et al., 2018) have been proposed to model the quality of rules, confidence is considered the appropriate for knowledge graph reasoning. However, the above methods assume rule independence both in confidence estimation and rule aggregation reasoning. Such the strong assumption may lead to redundant reasoning, further overestimating incorrect answers (Ott et al., 2021; Betz et al., 2023).

Parameterizing confidence bridges the gap between confidence estimation and aggregated reasoning (Ott et al., 2023), avoiding the risks associated with strong assumptions and the potential explosion of parameter combinations. Nonetheless, these approaches ignore the specificity and complexity of confidence estimation in temporal scenarios. Currently, TR-Rules (Li et al., 2023) investigated the misestimation of the confidence due to temporal redundancy. Some works of differentiable rule learning also set confidence as a learnable parameter, however, they mostly focus on interpolation reasoning or interval reasoning and (Xiong et al., 2022; Singh et al., 2023).

3 PROBLEM STATEMENT

3.1 Temporal Rules

A TKG can be denoted as $\mathcal{G} = \{(e_s, r, e_o, t)\}$, where $e_s, e_o \in \mathcal{E}$ are subject and object entity (Gu et al., 2022; Qu et al., 2023), $r \in \mathcal{R}$ represents the semantic relations and $t \in \mathcal{T}$ is timestamp. Based on timestamps, a TKG can be divided into sequential subgraphs $\{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_t\}$. Given a query $(e_q, r_q, ?, t_q)$, the objective of the temporal knowledge graph forecasting task is to predict the missing entity based history fact $\{\mathcal{G}_t | t < t_q\}$.

Rule-based TKGF aims to predict future events by learning and applying temporal rules. In this paper, we focus on modeling the confidence and temporal validity of rules rather than the methodology for mining them, so we use TLogic's rule learning approach to mine cyclic temporal logical rules (Liu et al., 2022), which is defined as follows:

$$(E_1, r_h, E_{l+1}, T_{l+1}) \leftarrow \wedge_{i=1}^{l}(E_i, r_i, E_{i+1}, T_i), \quad with \quad T_1 \leq T_2 \leq ... \leq T_l < T_{l+1} \quad (1)$$

where the left of the arrow is termed the rule head H, while the right is the rule body B. E_i and T_i are replaceable variables that represent entities and timestamps, respectively. r_h is the head relation and r_i is a body relation. $t_b(tr) = T_1$ denotes the earliest time of the grounded rule instance.

3.2 Rule-Based TKGF

Let's index quadruples and use $y_i \in \{0, 1\}$ denotes the truth value of quadruple $q_i = (e_s, r, e_o, t)$. For a target relation r, given the learned temporal rule set TR, where the size of the rule set is $K = |TR|$, $c(tr_j)$ is the confidence of the j_{th} temporal rule $tr_j \in TR$. A feature vector $\mathbf{\Delta t}_i \in \mathbb{N}_+^K$ for every example $\mathbf{\Delta t}_{ij} = t - t_b(tr_j)$ denotes the time interval between the rule body and fact. Specially, $\mathbf{\Delta t}_{ij} = +\infty$ if none of the relational path instance could be grounded by tr_j.

Usually, a heuristic confidence denotes the likelihood that the temporal rule tr_j predicted quadruple q_i, i.e. $p(y_i | c(tr_j))$. We are motivated by the belief that this predictive effectiveness should be subject to time rather than timelessness. Therefore, we need to design a time function $tv(\Delta t)$ interacting with the confidence, i.e. the likelihood should be the form of $p(y_i | \Phi(c(tr_j), tv(\Delta t_{ij})))$, abbreviated as $\Phi(tr_j, \Delta t)$, where $\Phi(\cdot)$ is a function that couples the confidence with time function.

For prediction, a typical approach is aggregating rules with Noisy-OR (Meilicke et al., 2019; Liu et al., 2022) to score q_i:

$$s(q_i) = 1 - \prod_{j=1}^{K}(1 - \Phi(tr_j, \Delta t_{ij})) \quad (2)$$

4 METHODOLOGY

4.1 Temporal Validity of Temporal Rules

Our core motivation is to model the temporal validity of temporal rules. Our approach is underpinned by two key intuitions: 1) The confidence of a temporal rule would diminish as the time interval increases; 2) The temporal sensitivity of the rules is different. Some rules experience an abrupt decline in predictive power, while others maintain efficacy over extended timeframes.

To model our intuition, there are two key points that must be satisfied: 1) A monotonically non-increasing time function $tv(\Delta t) \in [0, 1]$ needs to be designed for temporal decay on confidence. 2) The time function should contain a parameter indicating the decay rate of the temporal rule. To

satisfy the above two requirements, we design an exponential form of time decay function:

$$tv(\Delta t_{ij}) = e^{-\beta_j \Delta t_{ij}} \tag{3}$$

where β_j is the decay coefficient of tr_j that controls the decay rate of confidence $c(tr_j)$. A large β indicates that the confidence is decaying fast, i.e., the temporal rule is sensitive to temporal information and vice versa. While $\beta = 0$, the confidence of the rule is considered efficient forever.

Subsequently, we applied the time function to the confidence so that it could decay over time. The coupling function that integrates confidence with time function is formulated as follows:

$$\Phi(tr_j, \Delta t) = c(tr_j) * tv(\Delta t_{ij}) \tag{4}$$

In this paper, we only discuss the effect of temporal decay of confidence on the TKGF. More powerful temporal functions and coupling methods may be potential research interests in the future.

4.2 Learning Decay Coefficients and Confidence of Rules for TKGF

To accurately estimate the confidence and temporal validity of temporal rules, we use a machine learning model to learn the confidence and decay coefficients. Straightforwardly, we can compute the scores of the quadruples using the Noisy-OR function in Equation 2:

$$s(q_i) = 1 - \prod_{j=1}^{K} (1 - c(tr_j) * e^{-\beta_j \Delta t_{ij}}) \tag{5}$$

Noisy-OR is an aggregation function based on the assumption that the effectiveness of rules is independent, which implies the probability that at least one rule works. The cumulative product form of the score function may be difficult to be optimised due to potential gradient vanishing. Therefore, inspired by Relational Logistic Regression (Kazemi et al., 2014; Ott et al., 2023), we transform the Noisy-OR into a linear model through the $g(z) = log(1-z)$, the two ends of the equation are positively correlated while $z \in [0, 1)$. Rewriting $\tilde{\Phi}(tr_j, \Delta t_{ij}) = log(1 - \Phi(tr_j, \Delta t_{ij}))$ and we will obtain a new scoring function:

$$s(q_i) = \sum_{j=1}^{K} c(tr_j) * e^{-\beta_j \Delta t_{ij}} \tag{6}$$

From a feature engineering perspective, for a query (e_q, r_q, e_a, t_q), TempValid generates a K-dimensional feature vector, K denotes mined temporal rules for r_q. The feature value is the time interval between the t_q and the earliest timestamp of the grounded instance of the rule. In Eq. 6, if we regard the confidence and decay coefficients as feature weights for training, then essentially we are performing an exponential decay transformation on the temporal information of all the rules and then linearly aggregating them.

4.3 Negative Sampling and Optimization

Whichever score function is adopted, we need to make the score for the positive sample higher than the score for the negative sample. Conventionally, for a quadruple $q_i = (e_q, r_q, e_a, t_q)$, negative samples can be generated by simply replacing entity $q_i' = (e_q, r_q, e_c', t_q)$ or timestamps $q_i' = (e_q, r_q, e_a, t')$ (Sun et al., 2018; Leblay and Chekol, 2018). This approach, however, is not applicable to rule-based methods, as numerous entities simply have no rules to ground, leading to the generation of nonsense negative samples. Therefore, we propose two adapted negative sampling strategies based on the idea of replacing entities and replacing timestamps, respectively.

Rule-Adversarial Negative Sampling. For a query $(e_q, r_q, ?, t_q)$ and a set of temporal rules for r_q, when rules are applied to a query, it is possible to reach other incorrect candidate entities besides the correct one. A high-quality rule that not only provides accurate predictions but also avoids incorrect predictions will be assigned a greater weight. As a result, we select negative samples from quadruples that can be covered by the premined rules for training purposes. In pursuit of high-quality negative samples, we give precedence to those instances with high scores through TLogic (Liu et al., 2022). Nonetheless, there are still a great number of queries that are only able to generate a few negative samples. To alleviate this problem, negative samples generated by replacing timestamps are used as a supplement.

Time-Aware Negative Sampling. For TKGF, a positive sample requires not only the correct entities and relation, but also the correct timestamp. Guided by this intuition, we derive time-aware negative samples by replacing the timestamp of a positive sample. Employing a sliding window mechanism, we can readily generate a substantial amount of time-aware negative samples. For ex-

	#Nodes	#Rels	#Rules	#Train	#Used	#Valid	#Test	Interval
ICEWS14	6,869	230	23,814	74,845	61,670	8,514	7,371	24 hours
ICEWS18	23,033	256	32,394	373,018	172,923	45,995	49,995	24 hours
ICEWS0515	10,094	251	50,921	368,868	198,138	46,302	46,159	24 hours
YAGO	10,623	10	74	161,540	1,110	19,523	20,026	1 year
WIKI	12,554	24	83	539,286	1,245	67,538	63,110	1 year
GDELT	7,691	240	49,614	1,734,399	405,273	238,765	305,241	15 minutes

Table 1: Dataset statistics. #Rules denotes the number of pre-learned rules, and #Used denotes the number of quadruples used to generate the feature vectors.

ample, for a positive sample (e_q, r_q, e_a, t_q) and its feature vector $\mathbf{\Delta t}$ introduced in Section 3.2, if we set a sliding window of 200 with an offset time t_o, then we can get 200 negative samples with feature vector $\mathbf{\Delta t} - t_o$. We set all non-positive elements in the generated feature vector to $+\infty$ to indicate the disabling of the corresponding temporal rule.

Ultimately, in this paper, we adopt a loss function similar to the negative sampling loss (Mikolov et al., 2013; Sun et al., 2018):

$$L = -\log(s(q_i)) + \frac{1}{N}\sum_{n=1}^{N}\log(s(q_i^n)) \quad (7)$$

where q_i^n denotes n_{th} negative sample generated based on q_i, N is the number of negative samples.

5 EXPERIMENTS

5.1 Experiment Setup

Datasets. We evaluate TempValid on the entity prediction task of TKGF using six TKG datasets: ICEWS14, ICEWS05-15 (Garcia-Duran et al., 2018), ICEWS18, GDELT (Jin et al., 2020), WIKI (Leblay and Chekol, 2018), and YAGO (Mahdisoltani et al., 2013). We follow the previous works (Li et al., 2021b; Gastinger et al., 2023) to divide the dataset into training set and testing set, as well as validation set, with 8:1:1 according to the order of timestamps. The statistics are shown in Table 1.

Evaluation Metrics. We report two extensively adopted metrics for evaluating the performance of our model in temporal knowledge graph reasoning: Mean Reciprocal Rank (MRR) and Hits@k (H@k) in percentage (%). For each query, there is a rank for the true entity among all candidates. MRR is the average reciprocal values of the ranks, and Hits@k denotes the proportion where the ranks fall within the top k. A query may exist several correct answers, filtering out additional correct entities provides a more robust assessment of model performance. We adopt the widely-used time-aware filtered setting in TKGF to report the results (Han et al., 2020; Gastinger et al., 2023).

Baseline. We compare several renowned TKGF baselines: RE-NET (Jin et al., 2020), REGCN (Li et al., 2021b) and TiRGN (Li et al., 2022) which are evolutionary representation learning based methods; TANGO (Han et al., 2021), which uses neural ODEs; CyGNet (Zhu et al., 2021) featuring copy mechanisms; TITer (Sun et al., 2021) based on reinforce learning; xERTE (Han et al., 2020), known for its interpretability; the purely rule-based reasoning approach, TLogic (Liu et al., 2022) and TECHS (Lin et al., 2023) which performs reasoning with encoding temporal rules.

Hyperparameter Setting. Learning rate lr, batch size b and number of negative samples N are obtained by grid search based on the MRR of the validation set. Ultimately, 0.01 lr and 128 b are applied to all datasets, except ICEWS05-15 which uses 1024 batch size. For N, 50, 50, 50, 90, 70, and 60 are set for YAGO, WIKI, GDELT ICEWS14, ICEWS18 and ICEWS05-15, separately.

More details of the dataset and implementation are in the Appendix A and B.

5.2 Main Results

The results of the TKGF task are shown in Table 2 and 3. Notably, our approach outperforms TLogic, the method most closely related to ours, across all datasets. This suggests that the confidence and temporal validity modeled by our method surpass those derived from statistical or manually designed methods, as they can better capture complex semantic and temporal patterns in the data. Constrained by the size of the rule set and computational costs, both TLogic and TempValid focus exclusively on cyclic temporal rules which limits their ability to predict relations that require acyclic temporal rules for description. This is why TempValid and TLogic are worse than the representation-based models on YAGO, with some relations not being able to be modeled using only cyclic rules (the relevant data

Model	ICEWS14				ICEWS18				ICEWS0515			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RE-NET	38.48	28.52	42.85	58.10	28.02	18.62	31.59	46.44	44.56	34.16	50.06	64.51
CyGNet	39.86	30.11	44.02	58.21	29.78	19.73	32.55	48.46	40.42	29.44	46.06	61.60
TANGO	36.48	26.90	41.03	54.82	28.97	19.51	32.61	47.51	42.86	32.72	48.14	62.34
xERTE	40.79	32.70	45.67	57.30	29.31	21.03	33.51	46.48	46.62	37.84	52.31	63.92
REGCN	42.48	31.90	47.73	62.85	32.84	22.65	37.02	52.87	48.10	37.48	53.92	68.56
TITer	41.54	32.61	46.15	58.00	29.61	21.39	33.26	44.98	47.85	38.35	53.05	65.42
TiRGN	44.04	33.83	48.95	63.84	**33.66**	23.19	**37.99**	**54.22**	50.04	39.25	56.13	**70.71**
TLogic	42.53	33.20	47.61	60.29	29.59	20.42	33.60	48.05	46.94	36.16	53.24	67.21
TECHS	43.88	34.59	49.36	61.95	30.85	21.81	35.39	49.82	48.38	38.34	54.69	68.92
TempValid	**45.78**	**35.50**	**51.34**	**65.06**	33.50	**23.91**	37.89	52.33	**50.31**	**39.46**	**56.71**	70.55

Table 2: Performance for entity prediction task on ICEWS18, ICEWS14 and ICEWS0515.

Model	YAGO				WIKI				GDELT			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RE-NET	66.93	58.59	71.48	86.84	58.32	50.01	61.23	73.57	19.55	12.38	20.80	34.00
CyGNet	68.98	58.97	76.80	86.98	58.78	47.89	66.44	78.70	19.55	12.38	20.80	34.00
TANGO	63.34	60.04	65.19	68.79	53.04	51.52	53.84	55.46	19.66	12.50	20.93	33.55
xERTE	84.19	80.09	88.02	89.78	73.60	69.05	78.03	79.73	19.45	11.92	20.84	34.18
REGCN	82.30	78.83	84.27	88.58	78.53	74.50	81.59	84.70	19.69	12.46	20.93	33.81
TITer	87.47	80.09	89.96	90.27	73.91	71.70	75.41	76.96	18.19	11.52	19.20	31.00
TiRGN	87.95	**84.34**	**91.37**	**92.92**	81.65	**77.77**	85.12	87.08	21.67	13.63	23.27	**37.60**
TLogic	78.76	74.31	83.38	83.72	78.93	73.05	84.97	86.91	19.83	12.27	21.74	35.72
TECHS	**89.24**	-	-	92.39	75.98	-	-	82.39	-	-	-	-
TempValid	79.72	74.64	84.78	85.73	**83.19**	74.67	**90.12**	**97.54**	**21.88**	**14.37**	**24.40**	37.00

Table 3: Performance for entity prediction task on YAGO, WIKI and GDELT.

makes up more than 10% of the test set).

For non-rule-based methods, despite not modeling entity representations, TempValid demonstrates superiority over methods, such as RE-NET and REGCN, which are centered around entity evolving representations. Even though TiRGN draws on various strengths from different approaches, TempValid still achieves competitive results against it, which suggests that rule-based methods still hold promising competitive prospects.

5.3 Heuristic vs. Learned Confidence and Decay Coefficients

Previous efforts (Ott et al., 2023) have shown that confidence learned by a canonical model outperforms statistically derived confidence in static knowledge graph reasoning tasks. A question to be explored is whether TempValid's efficacy is primarily due to its learned confidence, temporal validity, or a synergy of both aspects. To answer the question, we design three variant models of TempValid on ICEWS14, ICEWS18 and ICEWS05-15: (1) learning confidence without temporal information (LCWT) which makes $\mathbf{\Delta t} \in \{0, 1\}^K$ which denotes whether the corresponding rule is applicable, (2) learning confidence with a uniform, fixed temporal validity (LCFT), and (3) employing the statistically derived confidence from TLogic while only learning temporal validity (LTV). We include TLogic which using heuristic confidence and uniform decay coefficients joins as well.

Model	ICEWS14	ICEWS18	ICEWS05-15
TLogic	42.53	29.59	46.94
LCWT	37.45	26.01	42.18
LCFT	44.72	32.60	48.61
LTV	44.93	32.38	48.86
TempValid	**45.78**	**33.50**	**50.31**

Table 4: Analysis of variant models of TempValid.

In Table 4, it can be observed that LCWT performs worse than TLogic, indicating the significance of temporal information in TKGF tasks. While learning confidence or temporal validity individually leads to performance surpassing that of TLogic, TempValid achieves a superior outcome by jointly optimizing both aspects. This suggests that our proposed TempValid find an effective way to couple the semantic confidence and temporal validity of temporal rules. While there may be room for

more sophisticated temporal validity modeling and coupling with confidence scores, we leave these explorations to future work.

In Figure 2, we further investigate the relative performance of LCFT against TempValid across different time decay coefficient β. It is evident that the model exhibits sensitivity to the β parameter. The intricate process of hyperparameter searching accentuates the necessity to learn rule-independent time decay coefficients.

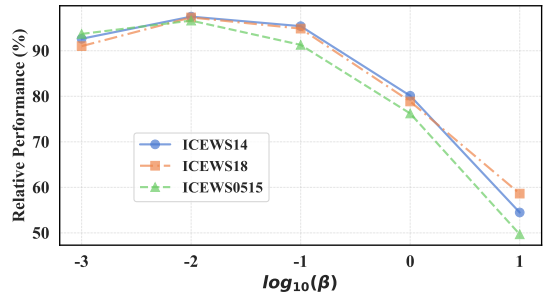


Figure 2: Sensitivity analysis of β on learning confidence with fixed temporal decay rate (LCFT).

5.4 Impact of Score Function and Negative Sampling Strategies

In this section, we discuss the contributions of optimization objectives and negative sampling strategies to the performance of TempValid.

Firstly, we experiment with replacing the scoring function with the Noisy-OR, which has been commonly used in previous works. Subsequently, we delve into the extent to which different negative sampling strategies contribute to the performance of TempValid. The strategy that optimizes only using positive samples in Equation 7 is abbreviated as OP. Removing the Rule-Adversarial Negative Sampling or Time-Aware Negative Sampling is abbreviated as w/o. TANS or w/o. RANS. The results are shown in Table 5.

Model	ICEWS14	ICEWS18	ICEWS05-15
Noisy-OR	28.28	23.32	38.29
Only Positive	15.29	23.78	40.50
w/o. RANS	39.17	28.62	44.73
w/o. TANS	45.41	33.29	49.69
TempValid	**45.78**	**33.50**	**50.31**

Table 5: Performance of different score functions and negative sampling strategies.

It can be observed that the performance using the Noisy-OR model significantly lags behind that of the canonical model. There are two potential reasons: 1) The Noisy-OR scoring function is computed under an assumption of rule independence, which may not be suitable when considering interactions among rules. 2) Under the assumption of rule independence and the Noisy-OR computation, even poor-quality rules can make a non-negligible positive contribution. Moreover, as we did not employ sophisticated techniques to filter rules such as clustering (Ott et al., 2021) or top-k rule selection (Betz et al., 2023; Liu et al., 2022), this might have allowed some detrimental rules to interfere with the prediction process (Ortona et al., 2018).

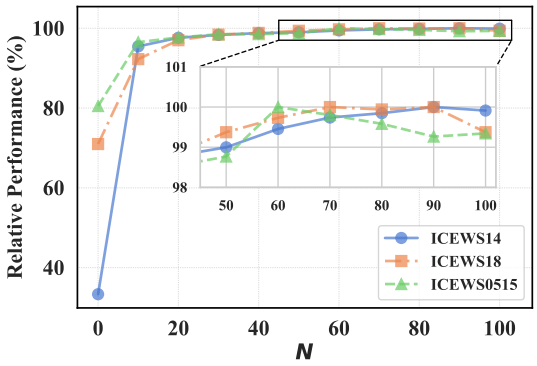


Figure 3: Relative performance with different number of negative samples N on TempValid.

We observed that both negative sampling strategies can bring some gains, but the performance drops significantly after removing RANS. The reason is that the TANS essentially generates negative samples through a simple linear transformation of the feature vectors, which limits its performance gain. Nevertheless, due to its simple and cost-effective, it still serves as a valuable alternative negative sampling approach.

In Figure 3, we illustrate the performance of TempValid under varying numbers of negative samples N. When N is zero, the model degenerates to the Only Positive variant as seen in Table 5. While $N \leq 50$, the model's performance improves as the number of negative samples increases apparently. It suggests that an appropriate amount of negative samples is beneficial in enabling the model to learn confidences and temporal decay coefficients effectively. As $N > 50$, the model's performance fluctuates slightly within a range that is close to the best performance which indicates that TempValid is not burdened by the overhead typically associated with extensive negative sampling and is not sensitive to the choice of the number of negative samples.

5.5 Cross-dataset Generalization

Generally, representation learning-based methods struggle with cross-dataset generalization, training a model on a dataset and performing well on another dataset, due to many entities are not shared across different datasets. Rule-based methods, by contrast, inherently possess an advantage in this regard as they model relations rather than specific entities. In this paper, we follow the settings of (Liu et al., 2022) for conducting cross-dataset generalization experiments.

$\mathcal{G}_{train}$	$\mathcal{G}_{test}$	Model	MRR	H@10
ICEWS0515	ICEWS14	AnyBURL	26.64	44.77
		TLogic	42.53	61.22
		TempValid	**45.72**	**65.08**
ICEWS14	ICEWS18	AnyBURL	15.46	29.58
		TLogic	29.15	47.95
		TempValid	**32.53**	**51.94**

Table 6: Reasoning across different datasets.

Specifically, we mine rules and train our model on ICEWS05-15, then perform predictions on ICEWS14. Similarly, we apply this procedure for training on ICEWS14 and predicting on ICEWS18. In Table 6, it can be seen that TempValid not only inherits TLogic's ability to reason across datasets, but it also improves upon it. It shows that TempValid's estimation of the confidence and decay rate of the rules is consistent across datasets.

5.6 Complexity Analysis

The time complexities of rule mining and feature vector generation in the TempValid framework, respectively, correspond to the training and inference time complexities within TLogic. For rule mining, the worst-case time complexity is $\mathcal{O}(|\mathcal{R}|nlDb)$, where l is the length of the rule, n is the number of walk, D is the maximum node degree, and b is the number of body samples for pre-estimating the confidence. For feature vector generation, the worst-case time complexity is $\mathcal{O}(|\mathcal{G}| + |TR|D^L|\mathcal{E}|\log(k))$, where L is the maximum rule length in temporal rule set $|TR|$ and k is the minimum number of candidates. Regarding model parameters, TempValid is a lightweight model that requires learning only two parameters per rule: the confidence and the decay coefficient. The proposed TempValid model trains with fewer than 50,000 parameters for all datasets.

Generating feature vectors from the training set can often result in a significant overhead, especially considering that the training set is typically larger than the test set. Fortunately, the number of parameters that TempValid needs to learn is relatively modest, making it possible to obtain near-optimal parameters even without extensive training data.

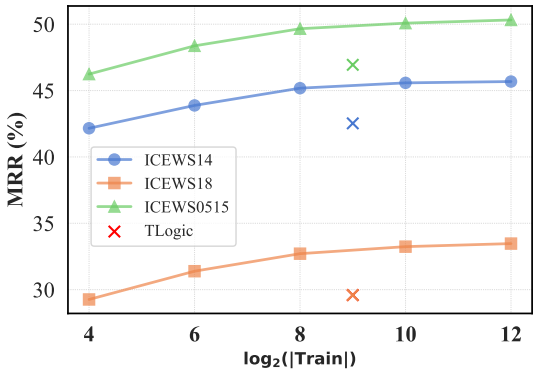


Figure 4: Performance with different size of train data $|Train|$ on TempValid.

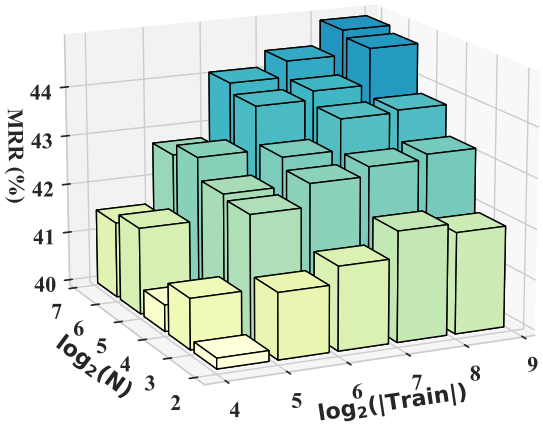


Figure 5: MRR with different size of train data $|Train|$ and number of negative samples N on ICEWS14.

In Figure 4, we show the relationship between model performance and the size of training data on ICEWS14, ICEWS18 and ICEWS05-15, where $|Train|$ denotes the maximum number of train data for a target relation. Evidently, TempValid can achieve baseline performance with as few as 16 data instances to learn the semantic confidences and decay coefficients. With just 32 data, TempValid attains results comparable to TLogic, which utilizes 500 samples for the estimation of confidence under standard settings. With just 32 data instances, TempValid attains results comparable to TLogic, which utilizes 500 samples for confidence estimation under standard settings.

The number of negative samples should also be considered as part of the training data. To this end,

id	Rules	C_{TL}	C_{TV}	DC
1	$(X, Engage\ in\ negotiations, Y, T_1) \leftarrow (X, Express\ intent\ to\ ease\ sanction, Y, T_0)$	0.96	1	0.001
2	$(X, Make\ a\ visit, Y, T_1) \leftarrow (X, Express\ intent\ to\ meet, Y, T_0)$	0.32	0.76	1.413
3	$(X, Engage\ in\ negotiations, Y, T_1) \leftarrow (X, Engage\ in\ cooperation, Y, T_0)$	0.26	0.17	0.121

Table 7: Case study. C_{TL} means confidence learned by TLogic, C_{TV} and DC denote confidence and decay coefficient learned by TempValid.

we show TempValid's performance on ICEWS14 as a function of training data size and number of negative samples in Figure 5. With the same amount of data, increasing the number of negative samples strengthen TempValid's capacity to learn semantic confidences and decay coefficients effectively. Referring to Figure 3, when the $N > 50$, the model can achieve near-optimal results. It implies that for training TempValid, we only need to generate feature vectors from a small subset of historical quadruples rather than generating them for the entire vast collection of training quadruples.

5.7 Case Study

To facilitate the understanding of TempValid's modeling mechanism, we provide a case study with three example rules in Table 7. It is observed that both TLogic and TempValid assigns a high confidence to the Rule 1, and TempValid learns a small decay coefficient. It indicates that Rule 1 is not sensitive to temporal information but are more focused on the semantic information. These rules are mainly composed of 3 step length rules. These types of rules typically have high confidence but low exposure (i.e., support (Galárraga et al., 2013)) and are usually composed of three-step length rules. For example, although Rule 1 showed high confidence, it only occurred 23 times in 500 samples.

Upon further observation of Rule 2 and Rule 3, TempValid assigned confidences that diverged from those generated by TLogic. For Rule 2, TempValid allocated a higher confidence and a decay coefficient, whereas for Rule 3, TempValid assigned a lower confidence and decay coefficient. This aligns with human cognition: If a person expresses an intention to visit someone, they are likely to meet within a short period. However, this probability decays very quickly because if they do not meet for a long time, the plan might be forgotten or canceled. As a comparison, cooperation is a long-term intention. Although the probability of it happening at any given moment is smaller than that of a meeting, in the long run, once this intention is formed, it will consistently drive the visit over a period of time.

6 Conclusion

In this paper, we propose the TempValid to model the temporal validity of temporal rules for TKGF. We believe that the predictive effectiveness (confidence) of the rules would decay over time. TempValid use a machine learning model to learn confidence and decay coefficients for TKGF. In addition, we design a rule-adversarial and a time-aware negative sampling strategies to train TempValid more efficiently, and obtain competitive results with the baselines on six classical benchmarks. We conclude that learning confidence and decay coefficients as well as bridging the gap between rule quality estimation and aggregate reasoning are necessary for rule-based TKGF. In addition, TempValid significantly outperforms existing TKGF method in the across datasets and low-resource scenarios.

7 Limitations

Our proposed has the following limitations: First, there are many patterns of rules in the time dimension, e.g., periodicity, randomness, etc., and our proposed TempValid models only temporal validity, i.e., the decay of the predictive effectiveness of rules. Second, more expressive time functions may exist. Without loss of generality, TempValid uses the common exponential form of the decay function inspired by (Yèche et al., 2023). Last but not least, the generation of feature vectors, i.e., rule grounding, is still time-consuming although it is a one-time process.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No .62276110, No .62172039 and in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank the anonymous reviewers and meta review for their comments on improving the quality of this paper.

References

Patrick Betz, Stefan Lüdtke, Christian Meilicke, and Heiner Stuckenschmidt. 2023. On the aggregation of rules for knowledge graph completion. In *ICML 2023 Workshop on Knowledge and Logical Reasoning in the Era of Data-driven Learning*.

Zixing Cai, Liyu Liu, Jingfeng Cai, and Baifan Chen. 2020. *Artificial Intelligence: Principles & Applications*. Beijing: Tsinghua university press.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 413–422.

Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821.

Julia Gastinger, Timo Sztyler, Lokesh Sharma, Anett Schuelke, and Heiner Stuckenschmidt. 2023. Comparing apples and oranges? on the evaluation of methods for temporal knowledge graph forecasting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 533–549.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3988–3995.

Yingjie Gu, Xiaoye Qu, Zhefeng Wang, Yi Zheng, Baoxing Huai, and Nicholas Jing Yuan. 2022. Delving deep into regularity: A simple but effective method for chinese named entity recognition. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1863–1873.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364.

Rikui Huang, Wei Wei, Xiaoye Qu, Wenfeng Xie, Xianling Mao, and Dangyang Chen. 2024. Joint multifacts reasoning network for complex temporal question answering over knowledge graph. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 10331–10335.

Yongfeng Huang, Yanyang Li, Yichong Xu, Lin Zhang, Ruyi Gan, Jiaxing Zhang, and Liwei Wang. 2023. Mvp-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 13417–13432.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6669–6683.

Seyed Mehran Kazemi, David Buchman, Kristian Kersting, Sriraam Natarajan, and David Poole. 2014. Relational logistic regression. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776.

Ningyuan Li, E Haihong, Shi Li, Mingzhi Sun, Tianyu Yao, Meina Song, Yong Wang, and Haoran Luo. 2023. Tr-rules: Rule-based model for link forecasting on temporal knowledge graph considering temporal redundancy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7885–7894.

Yujia Li, Shiliang Sun, and Jing Zhao. 2022. Tirgn: time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2152–2158.

Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2021a. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4732–4743.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021b. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417.

Qika Lin, Jun Liu, Rui Mao, Fangzhi Xu, and Erik Cambria. 2023. Techs: Temporal logical graph networks

for explainable extrapolation reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 1281–1293.

Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4120–4127.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eight International Joint Conference on Artificial Intelligence*, pages 3137–3143.

Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *Transactions on Machine Learning Research*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*.

Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. 2018. Robust discovery of positive and negative rules in knowledge bases. In *2018 IEEE 34th International Conference on Data Engineering*, pages 1168–1179. IEEE.

Simon Ott, Patrick Betz, Daria Stepanova, Mohamed H Gad-Elrab, Christian Meilicke, and Heiner Stuckenschmidt. 2023. Rule-based knowledge graph completion with canonical models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1971–1981.

Simon Ott, Christian Meilicke, and Matthias Samwald. 2021. Safran: An interpretable, rule-based link prediction method outperforming embedding models. In *3rd Conference on Automated Knowledge Base Construction*.

Thomas Pellissier Tanon, Daria Stepanova, Simon Razniewski, Paramita Mirza, and Gerhard Weikum. 2017. Completeness-aware rule learning from knowledge graphs. In *16th International Semantic Web Conference*, pages 507–525.

Xiaoye Qu, Jun Zeng, Daizong Liu, Zhefeng Wang, Baoxing Huai, and Pan Zhou. 2023. Distantly-supervised named entity recognition with adaptive teacher learning and fine-grained student ensemble. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13501–13509.

Arthur M Schlesinger. 1999. *The cycles of American history*. HMH.

Ishaan Singh, Navdeep Kaur, Garima Gaur, et al. 2023. Neustip: A neuro-symbolic model for link and time prediction in temporal knowledge graphs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4497–4516.

Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *International Conference on Machine Learning*, pages 3462–3471.

Garry Winston Trompf. 1979. *The idea of historical recurrence in Western thought: from antiquity to the Reformation*, volume 1. Univ of California Press.

Jiapu Wang, Boyue Wang, Meikang Qiu, Shirui Pan, Bo Xiong, Heng Liu, Linhao Luo, Tengfei Liu, Yongli Hu, Baocai Yin, et al. 2023. A survey on temporal knowledge graph completion: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2308.02457*.

Siheng Xiong, Yuan Yang, Faramarz Fekri, and James Clayton Kerce. 2022. Tilp: Differentiable learning of temporal logical rules on knowledge graphs. In *The Eleventh International Conference on Learning Representations*.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023. Temporal knowledge graph reasoning with historical contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4765–4773.

Hugo Yèche, Alizée Pace, Gunnar Ratsch, and Rita Kuznetsova. 2023. Temporal label smoothing for early event prediction. In *International Conference on Machine Learning*, pages 39913–39938.

Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. 2021. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open*, 2:14–35.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4732–4740.

A Details of the dataset

For datasets, on the one hand, Considering the existence of multiple variant datasets with identical names, we adhere to the dataset selection and processing strategies outlined by (Gastinger et al., 2023). On the other hand, TempValid needs to ground the rules from the raw training data to generate feature vectors. Subject to the temporal constraints of the temporal rules, the earlier the timestamp, the fewer rules the quadruple can be grounded, the sparser the generated feature vectors will be.

B Implementation Details

B.1 Hyperparameters search

The search range for hyperparameters is as follows: learning rate $lr \in \{0.1, 0.01, 0.001, 0.0001\}$, batch size $b \in \{64, 128, 256, 512, 1024\}$ and number of negative samples $N \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. The maximum epochs for training is set as 3000. The optimal model parameters are selected based on the MRR of validation set and implement early stopping to prevent overfitting.

B.2 Feature vectors generation

For a query $(e_s, r, ?, t)$, we endeavor to ground the learned temporal rules TR. If there is a candidate could be connected by temporal rule tr_j, then Δt_j will be assigned as the time interval between the rule and t, otherwise $\Delta t_j = +\infty$.

For rule-adversarial negative samples, we generate feature vectors for the candidate entities reachable by TLogic for training. For ICEWS14, YAGO and WIKI, we generate the feature vectors of top 100 candidate entities. For ICEWS18, ICEWS05-15 and GDELT, we generate the feature vectors of top 50 candidate entities. If the number of negative samples is still insufficient, we supplement it with the time-aware negative sampling strategy.