APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 380–390
https://doi.org/10.14743/apem2025.3.547

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Manufacturing process quality prediction via temporal knowledge graph reasoning with adaptive multi-scale temporal path fusion and self-attention mechanism

Zong, H.[a,b,c,d,*], **Shuai, B.**[a,b,c,d,*]

[a]School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, P.R. China
[b]Institute of System Science and Engineering, Southwest Jiaotong University, Chengdu, P.R. China
[c]National United Engineering Laboratory of Intergrated and Intelligent Transportation, Southwest Jiaotong University, Chengdu, P.R. China
[d]National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Chengdu, P.R. China

ABSTRACT

To tackle the pronounced temporal dynamics and intricate interdependencies within process manufacturing knowledge, this paper introduces an innovative framework: the Adaptive Multi-Scale Temporal Path Fusion Network (AMTPFNet). The method constructs short-term (high-frequency) and long-term (low-frequency) historical subgraphs to generate multi-scale temporal representations. It also employs a self-attention mechanism for query-aware temporal path modeling, enabling adaptive weight allocation based on varying time spans. Extensive experiments are conducted on benchmark datasets, including ICEWS18, GDELT, WIKI, and YAGO. Additionally, an application analysis is presented using electromechanical fault data. The results demonstrate that AMTPFNet exhibits remarkable effectiveness and robustness in temporal knowledge graph reasoning tasks, achieving MRR scores of 0.914 on YAGO and 0.838 on WIKI. It achieves high efficiency in predicting future production facts and assessing process quality in industrial workflows. Root causes of failures (e.g., insulation, friction) for motor components (stators, rotors) are accurately predicted, demonstrating the framework's transferability to real-world manufacturing scenarios. Although electromechanical fault data are used as a case study, the framework generalizes to manufacturing quality prediction and is readily transferable to finance, healthcare, and social media analytics.

ARTICLE INFO

Keywords:
Temporal knowledge graph reasoning (TKGR);
Multi-scale temporal modeling;
Temporal path fusion;
Self-attention mechanism;
Knowledge graph embedding;
Manufacturing process analytics;
Process quality prediction

**Corresponding author:*
zonghan@foxmail.com
(Zong, H.)
bsh67@126.com
(Shuai, B.)

Article history:
Received 25 August 2025
Revised 1 October 2025
Accepted 13 October 2025

1. Introduction

With the rapid advancement of Industrial Internet technologies, process manufacturing enterprises have amassed substantial volumes of historical data through intelligent monitoring terminals and data acquisition systems. These data capture equipment operating conditions, failure events, and associated process operations, encompassing dynamic information such as the failure types of critical equipment, temporal patterns of occurrence, and their scope of impact. Moreover, these data exhibit complex temporal coupling with process parameters and product quality out-

comes. Over time, enterprises have also accumulated extensive domain knowledge and experiential rules, including operational guidelines, fault response strategies, and maintenance protocols. The deep integration of equipment failure data with domain knowledge provides a robust foundation for intelligent diagnostics. This enables quality prediction in process manufacturing based on anomalous equipment behaviors.

Currently, most studies have focused on quality prediction using mathematical modeling or machine learning techniques. While these approaches have yielded performance improvements to some extent, they are hindered by two significant limitations. First, they lack the capability to structurally represent and integrate domain knowledge, such as operational standards and experiential insights, particularly in situations of time constraints or crises [1]. Second, they fail to effectively capture the pronounced temporal characteristics and interdependent coupling among variables inherent in process data [2]. Although machine learning techniques have been applied for fault prediction in manufacturing, they may not adequately capture the temporal dynamics and interdependencies within process data [3, 4]. Consequently, a critical challenge in intelligent process manufacturing is to develop a quality prediction model that effectively integrates process data with domain knowledge. This model must also be both temporally aware and capable of causal reasoning.

Knowledge Graph (KG), as a structured representation method for entities and their relationships, has been widely applied in various fields such as Natural Language Processing, Recommender Systems, and Intelligent Question Answering. Xie *et al*. [5] demonstrates the potential of knowledge graphs in managing the complex relationships and dependencies in supply chains. However, knowledge in the real world is not static; it exhibits significant temporal dynamics. For instance, the evolution of international events, the changes in social network relationships, and the fluctuations in financial transactions all evolve over time. To more accurately model such dynamic knowledge, Temporal Knowledge Graphs (TKG) have emerged. By incorporating temporal dimension information into traditional knowledge graphs, TKGs empower reasoning models with the ability to understand and predict the evolution of relationships between entities over time, thereby demonstrating substantial value in tasks such as event prediction, anomaly detection, and knowledge completion.

Knowledge Graph Embedding (KGE) aims to map entities and relations into continuous vector spaces, thereby enhancing computational efficiency and model generalization while preserving the structural information of the graph. KGE has been widely applied in knowledge graph reasoning tasks. Early embedding methods primarily include translation-based models and semantic matching models [6]. Translation-based models, such as that proposed by Bordes *et al*. [7], assume that a relation is represented as a translation vector from the head entity to the tail entity. While Wang *et al.* [8] and Lin *et al.* [9] capture more complex relationships by projecting into a specific relationship or entity space. Semantic matching models, including those by Trouillon *et al*. [10], model the interactions between entities and relations through tensor decomposition and other related techniques. Furthermore, deep neural network-based models, such as those proposed by Dettmers *et al*. [11] and Schlichtkrull *et al*. [12], leverage convolutional neural networks and graph neural networks to further enhance the expressiveness of embeddings.

However, these methods are primarily designed for static knowledge graphs and are unable to effectively handle dynamic information that evolves over time. Knowledge graphs in the real world are subject to dynamic changes; for example, the attributes of entities may change over time, and relationships between entities can emerge, disappear, or evolve [13]. Therefore, effectively handling temporal information has become a key focus in the research of knowledge graph embedding. To capture the temporal characteristics of facts in knowledge graphs, researchers have introduced the concept of Temporal Knowledge Graphs (TKG) and developed various temporal reasoning methods. Dasgupta *et al*. [14] capture temporal information by projecting entities and relations onto hyperplanes associated with specific timestamps. Goel *et al*. [15] define a function that takes entities and timestamps as inputs to generate time-specific representations. In addition, Liu *et al*. [16] defined the joint probability distribution of all events using an autoregressive approach to perform temporal reasoning. Zhu *et al*. [17] propose a copy-generation mechanism to predict future events based on historical vocabulary. Li *et al*. [18] introduce a model that incorporates a Historical Information Completion Strategy (HICS) and a Pretrained Language Model

(PLM) to perform interpretable inductive reasoning over temporal knowledge graphs. Furthermore, Liu *et al*. [19] enhance the accuracy and interpretability of temporal prediction by proposing an adaptive framework for temporal knowledge graph reasoning with dynamic event-driven dependencies. Some studies have also begun to explore the integration of knowledge graphs with Large Language Models (LLMs) to enhance the capability of LLMs in complex reasoning and knowledge-intensive tasks [20]. In addition, Transformer models have been revisited and applied to knowledge graph reasoning to address the limitations of message passing neural networks [21].

However, these methods typically rely on fixed-length time windows. This makes it difficult to simultaneously capture both short-term high-frequency dynamics and long-term low-frequency trends. In addition, many methods use simple weighting or multiplication operations in edge weight modeling, failing to adequately consider the varying importance of different historical facts for querying relationships, which affects the reasoning capability of the model.

To address the aforementioned issues, this paper proposes an Adaptive Multi-scale Temporal Path Fusion Network (AMTPFNet). The key innovations of this method are as follows:

Multi-scale Historical Information Modeling: By constructing short-term (high-frequency) and long-term (low-frequency) historical subgraphs, the recent dynamic changes and long-term evolution trends are captured respectively, enabling the model to leverage more comprehensive temporal information for reasoning.

Query-aware Temporal Path Modeling: Utilizing a self-attention mechanism, query-based path representation is constructed in the multi-scale historical graph, so that the path information of different time spans can adaptively adjust the weight, thereby enhancing the reasoning accuracy.

Adaptive Edge Weight Assignment: During the temporal information passing process, the weight of the edge is dynamically allocated through the attention mechanism, enabling the model to more accurately model the impact of different historical facts on the current query.

Through these innovations, experimental results on multiple public datasets (such as ICEWS18, GDELT, WIKI, and YAGO) demonstrate that AMTPFNet excels in key metrics such as MRR, Hits@1, Hits@3, and Hits@10, validating its effectiveness and robustness in temporal knowledge graph reasoning tasks.

2. Methodology

2.1 Problem statement

Temporal Knowledge Graph is an extended form of a Knowledge Graph, where each triple (s, r, o) (representing the relationship between subject s, relation r, and object o) is associated with a specific timestamp t, forming a quadruple (s, r, o, t). Since TKGs aim to capture the dynamic characteristics of entities and relationships evolving over time, their reasoning tasks are primarily divided into two categories [22]:

Interpolation reasoning: Predicting missing triples in the case of missing facts at certain timestamps [15, 23]. For example, completing events that are missing at some time points.

Extrapolation reasoning: Predicting future possible facts based on historical events [17, 24, 25], such as forecasting the event $(s, r, ?, t + 1)$ at a future time $t + 1$.

This study investigates extrapolative reasoning through a temporal knowledge graph constructed from fault information and causal relationships in electromechanical equipment, specifically motors and their components (e.g., stators, rotors, drive-end bearings, non-drive-end bearings), with associated timestamps of fault occurrences. Formally, the knowledge graph is represented as a set of temporal quadruples $G = (s, r, o, t)$. Given a historical time window G_τ and a query $(s, r, ?, t + 1)$, the task is to infer the most likely object entity o, representing the root cause of the failure. This approach aims to enhance the timeliness and effectiveness of fault prediction, thereby improving the accuracy and reliability of intelligent diagnostic systems in industrial environments.

2.2 Model formulation

To address the limitations of existing methods in capturing multi-scale temporal information, this paper proposes an Adaptive Multi-Scale Temporal Path Fusion Network (AMTPFNet). By modeling both short-term (high-frequency) and long-term (low-frequency) historical information, and incorporating a self-attention mechanism for query-aware path modeling, the proposed method enhances the accuracy and robustness of TKG reasoning tasks.

Basic mathematical notations

Detailed information on the basic mathematical notations and the corresponding description of AMTPFNet are shown in Table 1.

Table 1 Applicable mathematical notation list

Notation	Description
s	subject entity of the query
r	relation of the query
o	candidate object entities； o^+ denotes the positive sample, and o_j^- denotes the j-th negative sample
t	current timestamp; with the query timestamp set to $t+1$
τ	timestamp
G_τ	historical subgraph at timestamp τ
r_τ	composite relation representation, which combines the relation r with the timestamp τ
k_{short}	the number of continuous timestamps at a short-term scale
m_{total}	the temporal duration within a long-term historical interval
Δ	sampling interval at the long-term scale
G_{short}	short-term historical graph, containing facts for $\tau \in [t-k_{\text{short}}+1, t]$
G_{long}	long-term historical graph, containing facts for $\tau \in \{t-m_{\text{total}}+1, t-m_{\text{total}}+1+\Delta, \ldots, t-k_{\text{short}}\}$
$p_{s\to o}^{(i)}$	the set of all paths from s to o at scale i
p	a path consisting of several edges
$f(p)$	mapping function of path p
$\phi(e_i)$	encoded representation of edge e_i
$\oplus$	aggregation operation on path representations (e.g., summation, averaging)
$\otimes$	continuous fusion operation along the path (element-wise product)
$h_k^{(i)}(s,r,o)$	node representation at layer k at scale i
$\Psi(r) \in R^d$	static initial representation of query relation r
$N^{(i)}(o)$	the set of neighboring nodes of node o at scale i
$\Delta\tau(o',o)$	the relative temporal difference of the edge (o',r,o), defined as $(t+1)-\tau$
$w(r,\Delta\tau)$	edge weight vector computed based on relation r and the relative temporal difference Δτ
$f_{\text{edge}}(\cdot)$	feedforward network for computing edge weights
TimeEmb($\Delta\tau$	temporal embedding based on Δτ
a	self-attention vector
$\odot$	Hadamard product (element-wise product)
|	vector concatenation operation
K	number of message passing layers
$z_{s\to o}^{(i)}$	final query-aware representation after K layers of updates at scale i
$z_{s\to o}$	final representation after multi-scale fusion
$\Phi(\cdot)$	multi-scale information fusion function implemented using MLP
$\chi(r)$	projected representation of query relation r
F(·)	scoring function (feedforward network)
$\sigma(\cdot)$	Sigmoid activation function
L_{cls}	binary cross-entropy loss
R	relation embedding matrix
I	identity matrix
$\lvert\cdot\rvert_F$	Frobenius norm
L_{reg}	orthogonal regularization term for relation embeddings
L_{attn}	attention regularization term, encourage attention distribution is close to uniform.
λ_1, λ_2	regularization hyperparameter
L_{total}	total loss

Multi-scale historical temporal graph construction

For a given query $(s, r, ?, t+1)$, where s is the query subject, r represents the query relation, and 1 is a time step. We construct historical subgraphs at different scales i. For scale i (e.g., short-term, long-term), the time interval t in the four-dimensional sequence (s, r, o, t) is used to determine the scale, and the short-term and long-term ranges are specified either heuristically or through grid search.

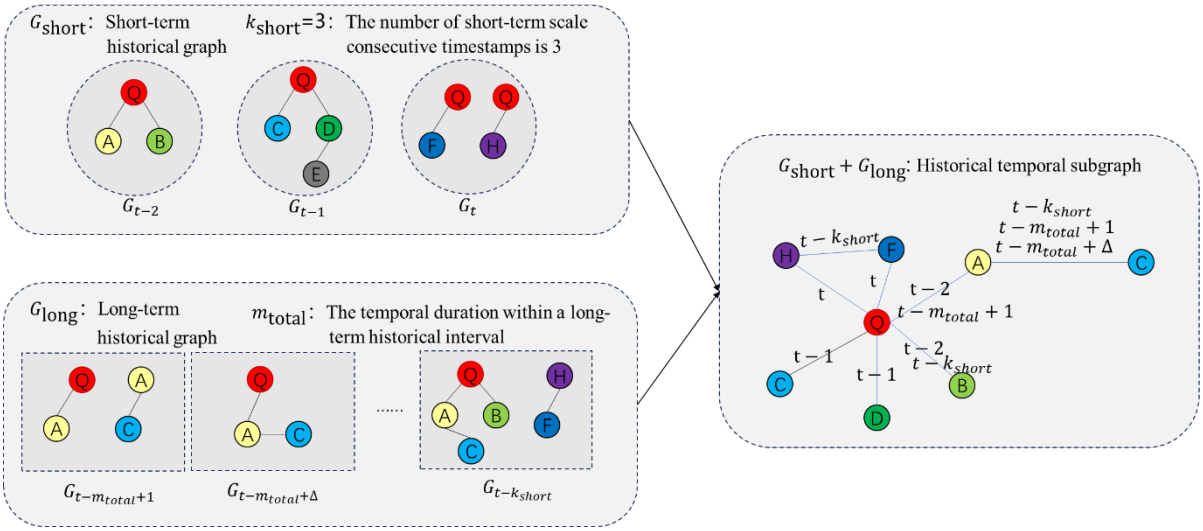

Fig. 1 Example of historical temporal subgraph construction

Let the corresponding historical length be m_i, and define the multi-scale historical temporal graph as $G_{(i)} = (\mathrm{s}, r_\tau, o)|(\mathrm{s}, \mathrm{r}, \mathrm{o}) \in G_\tau, \tau \in [\mathrm{t} - m_i + 1, \mathrm{t}]$, where r_τ represents the composite representation of relation r and the time label τ. $G_{(i)}$ denotes the historical temporal subgraph at scale i. Therefore, we construct short-term and long-term historical graphs, as shown in Fig. 1.

As shown in Fig. 1, let the most recent continuous k_{short} timestamps be $\tau \in [t - k_{short} + 1, t]$, and the corresponding short-term historical graph is denoted as $G_{\text{short}} = (\mathrm{s}, r_\tau, \mathrm{o})|(\mathrm{s}, \mathrm{r}, \mathrm{o}) \in G_\tau, \tau \in [\mathrm{t} - k_{short} + 1, \mathrm{t}]$. Let the long-term historical interval be $\tau \in [t - m_{total} + 1, t - k_{short}]$, and sampled at intervals of Δ, then the long-term historical graph is denoted as $G_{\text{long}} = (\mathrm{s}, r_\tau, \mathrm{o})|(\mathrm{s}, \mathrm{r}, \mathrm{o}) \in G_\tau, \tau \in \{\mathrm{t} - m_{total} + 1, \mathrm{t} - m_{total} + 1 + \Delta, \ldots, \mathrm{t} - k_{short}\}$. This design ensures that the short-term scale uses continuous time information, while the long-term scale captures historical trends at different frequencies through sparse sampling.

$$\tau \in \{t - m_{total} + 1,\ t - m_{total} + 1 + \Delta,\ \ldots,\ t - k_{short}\}$$

Temporal path definition and abstraction

For a given query $(s, r, ?, t+1)$ and candidate entity $\mathrm{o} \in \mathcal{V}$, within any historical graph G (whether G_{short} or G_{long}), the set of all paths from the query s to candidate o at scale i is denoted as $p_{\mathrm{s}\to\mathrm{o}}^{(\mathrm{i})}$. The representation of a single path p (composed of k edges) is defined as $f(p) = \otimes_{i=1}^{k} \phi(e_j)$, where $\phi(e_i)$ represents the encoding of each edge e_i, which can be obtained through a subsequent edge encoding module, and $\otimes$ denotes the continuous fusion operation along the path (element-wise multiplication). The aggregated path representation is then obtained as:

$$\mathbf{z}_{s\to o}^{(i)} = \bigoplus_{p \in p_{\mathrm{s}\to\mathrm{o}}^{(\mathrm{i})}} f(p), i \in \{short, long\}, \tag{1}$$

where $f(p)$ is the mapping function of path p (for example, after applying linear or nonlinear transformations to the edges along the path, followed by aggregation), and $\oplus$ denotes the aggregation operation.

Query-aware temporal path processing

At each scale i, k layers of message passing are used to update the path representations. The representation at the k-th layer is denoted as $h_{k^{(i)}}(s,r,o)$, with the initial layer defined as:

$$\boldsymbol{h}_0^{(i)}(s,r,o) = \begin{cases} \Psi(r), & o \Leftrightarrow s, \\ \mathbf{0}, & otherwise, \end{cases} \tag{2}$$

where $\Psi(r) \in R^d$ is the initial static representation of the query relation r, with d being the embedding dimension.

For the k-th layer, the update of the target node o is based on its neighbors o' (satisfying $(o',r,o) \in G_i$):

$$\mathbf{h}_{k^{(i)}}(\mathrm{s},\mathrm{r},\mathrm{o}) = \sum_{o' \in \mathcal{N}^{(i)}(o)} \alpha_{o',o}^{(i)} \left(\mathbf{h}_{k-1}^{(i)}(\mathrm{s},\mathrm{r},o') \odot w\big(\mathrm{r}, \Delta\tau(o',\mathrm{o})\big) \right) \tag{3}$$

where $\mathcal{N}^{(i)}(o)$ denotes the set of neighbors connected to node o at scale i; $\odot$ represents the Hadamard product (element-wise multiplication) operator; $\Delta\tau(o',o)$ denotes the relative temporal distance of the edge $e = (o',r,o)$, i.e., $\Delta\tau(o',o) = (t+1) - \tau$, where τ is the timestamp of the edge; $w\big(r, \Delta\tau(o',o)\big)$ represents the edge weight vector computed based on relation r and the relative temporal difference. The weight vector is calculated as: $w(r,\Delta\tau) = f_{edge}([\Psi_{(r)}; \mathrm{TimeEmb}\ (\Delta\tau(o',o)])$, where $f_{\mathrm{edge}}(\cdot)$ is a feed-forward network, $\Psi_{(r)}$ is the static representation of the query relation r, and $\mathrm{TimeEmb}\big(\Delta\tau(o',\mathrm{o})\big)$ is the time embedding based on the relative time $\Delta\tau$, allowing the edge representation to capture both relational and temporal information. $\alpha_{o',o}^{(i)}$ is the self-attention weight, and its computation is as follows:

$$\alpha_{o',o}^{(i)} = \frac{\exp\left(\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{h}_{k-1}^{(i)}(s,r,o') \| w(r,\Delta\tau(o',o))])\right)}{\sum_{o'' \in \mathcal{N}_i(o)} \exp\left(\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{h}_{k-1}^{(i)}(s,r,o'') \| w(r,\Delta\tau(o'',o))])\right)} \tag{4}$$

where a is the learnable attention vector, and || denotes the vector concatenation operator. The symbol || represents concatenating two (or more) vectors along their dimensions, that is, arranging them sequentially in an "end-to-end" manner to form a new vector, whose new dimension is the sum of the individual vector dimensions.

By introducing the self-attention mechanism, the information of each edge can receive an adaptive weight based on its relevance to the query during message passing, thereby enhancing the effectiveness of multi-resolution information aggregation. After K layers, the final query-aware representation for each scale is obtained as:

$$z_{s\to o}^{(i)} = h^{(i)}K(s,r,o), \quad i \in \{short, long\} \tag{5}$$

Multi-scale information fusion and score function formulation

The representations of the two scales are fused to obtain the final representation:

$$\mathbf{z}_{s\to o} = \Phi(\mathbf{z}_{s\to o}^{(short)}, \mathbf{z}_{s\to o}^{(long)}) = \mathrm{MLP}(\mathbf{z}_{s\to o}^{(short)} \| \mathbf{z}_{s\to o}^{(long)}) \tag{6}$$

where $\Phi(\cdot)$ is the fusion function, implemented here using an MLP, and || denotes the concatenation operation, which connects the vectors of the two scales into a new vector. By fusing the representations of the short-term and long-term scales through the MLP, both recent high-frequency changes and long-term low-frequency trends can be captured simultaneously.

Next, the fused representation is concatenated with the projection $\chi(r)$ of the query relation and passed through a feed-forward network $\mathcal{F}(\cdot)$ to obtain the score:

$$\mathrm{Score}(s,r,o) = \mathcal{F}(\mathbf{z}_{s\to o} \| \chi(r)) \tag{7}$$

The final predicted probability formula is as follows:

$$p(o \mid s,r) = \sigma(\mathrm{Score}(s,r,o)) \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function, which maps the output of the $\mathrm{Score}(s,r,o)$ to the interval (0,1), serving as the predicted probability.

Loss function and parameter learning

For each query, positive samples (s, r, o^+) and a set of negative samples $\{(s, r, o_j^-)\}_{j=1}^N$ are constructed, and the binary cross-entropy loss is employed:

$$L_{\text{cls}} = -\log \sigma\left(\text{Score}(s, r, o^+)\right) - \sum_{j=1}^{N} \log\left[1 - \sigma\left(\text{Score}(s, r, o_j^-)\right)\right]. \tag{9}$$

Additionally, an orthogonal regularization term for the relation embeddings is added:

$$L_{\text{reg}} = \lambda_1 \|R^\top R - I\|_F^2, \tag{10}$$

Furthermore, an attention regularization term is introduced to encourage each layer's attention distribution to be close to uniform:

$$L_{\text{attn}} = \lambda_2 \sum_{i \in short, long} \sum_{o' \in \mathcal{N}_i(o)} \left\| \alpha_{o',o}^{(i)} - \frac{1}{|\mathcal{N}_i(o)|} \right\|^2 \tag{11}$$

where $|\mathcal{N}_i(o)|$ denotes the number of neighbors of o at scale i. The final total loss is:

$$L_{\text{total}} = L_{\text{cls}} + L_{\text{reg}} + L_{\text{attn}}. \tag{12}$$

3. Algorithmic time complexity analysis

Time complexity: AMTPFNet divides historical information into short-term and long-term scales, handling k_{short} consecutive timestamps (with a history length of m_{short}) and a long-term history sampled at an interval Δ (with a history length of m_{long}). At the short-term scale, the algorithm performs ω layers of query-aware temporal path aggregation on the historical subgraph edge set $|\mathcal{E}_{\text{short}}|$, with a complexity of $O(\omega(m_{\text{short}} \cdot |\mathcal{E}_{short}| + |\mathcal{V}|))$; at the long-term scale, the processing complexity of the edge set $|\mathcal{E}_{\text{long}}|$ is $O(\omega(m_{\text{long}} \cdot |\mathcal{E}_{\text{long}}| + |\mathcal{V}|))$. Subsequently, the two-scale representations are fused through an MLP, with a complexity of $O(|\mathcal{V}|)$. Therefore, the overall time complexity is $O(\omega(m_{\text{short}} \cdot |\mathcal{E}_{\text{short}}| + m_{\text{long}} \cdot |\mathcal{E}_{\text{long}}| + 2|\mathcal{V}|))$. This multi-scale processing improves the reasoning capability by separating short-term dynamics and long-term trends while maintaining computational efficiency.

Space complexity: The space complexity of AMTPFNet mainly comes from the storage of relation embeddings and temporal path representations. The learnable embeddings of the relation types set $\boldsymbol{R} \in R^{|\mathbb{R}| \times d}$ occupy $O(|\mathcal{R}| \cdot d)$ space. The temporal path representations for the short-term and long-term scales, $\mathbf{H}_{\text{short}}$ and $\mathbf{H}_{\text{long}} \in \mathbb{R}^{|\mathcal{V}| \times d}$, collectively occupy $O(2|\mathcal{V}| \cdot d)$ space. In addition, model parameters (such as the weights and biases of linear transformations) occupy $O(d^2)$ space. Therefore, the total space complexity is $O(|\mathcal{R}| \cdot d + 2|\mathcal{V}| \cdot d + d^2)$. Compared to single-scale methods, the multi-scale design only adds an additional $|\mathcal{V}| \cdot d$ space overhead, which still remains linear in nature, making it suitable for large-scale datasets.

4. Experiments

The experimental datasets, including ICEWS18, GDELT, WIKI, YAGO, are introduced, and the performance of AMTPFNet on these datasets is evaluated. A comparison is made between AMTPFNet and existing methods, such as TiPNN, DaeMon, TransE, and TTransE.

4.1 Datasets

This study uses four temporal knowledge graph datasets: ICEWS18, GDELT, WIKI, and YAGO. ICEWS18 is derived from the Integrated Crisis Early Warning System (ICEWS), which records global political events in 2018. GDELT (Global Database of Events, Language, and Tone) covers global news events. WIKI and YAGO are subsets of the historical data from Wikipedia and YAGO3, respectively, containing temporally annotated knowledge. In the data preprocessing step, each dataset is divided into training (80 %), validation (10 %), and test (10 %) sets in chronological order, ensuring that the timestamps of the training set are earlier than those of the validation set,

and the timestamps of the validation set are earlier than those of the test set [23]. Table 2 provides an overview of the temporal knowledge graph datasets.

Table 2 Overview of temporal knowledge graph datasets

Dataset	Statistical number					Time interval
	Entities	Relation types	Training samples	Validation samples	Testing samples	
YAGO	10623	10	161540	19523	20026	1 year
WIKI	12554	24	539286	67538	63110	1 year
ICEWS18	23033	256	373018	45995	49545	1 year
GDELT	7691	240	1734399	238765	305241	15 mins

4.2 Evaluation metrics

In this experiment, the link prediction task is used to evaluate the model's performance. The objective of this task is to measure the ranking of the true entity among all candidate entities. Mean Reciprocal Rank (MRR), along with Hits@1, Hits@3, and Hits@10, is employed as evaluation metrics. These metrics provide a comprehensive assessment of the model's ranking performance and prediction accuracy in knowledge graph completion tasks. The formula for calculating MRR is as follows:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{13}$$

where $|Q|$ represents the number of queries, and $rank_i$ denotes the rank of the target entity in the i-th query.

$Hits@K$ measures the proportion of correct answers ranked within the top K, and its formula is as follows:

$$Hits@K = \frac{1}{|Q|}\sum_{i=1}^{|Q|} I\left(rank_i \le K\right) \tag{14}$$

where $I(\cdot)$ is the indicator function, which takes the value of 1 when $rank_i \le K$, and 0 otherwise.

To ensure a fair comparison, the filtered setting, which is consistent with other baseline methods, is adopted. In this setting, all possible incorrect entities are removed during evaluation to ensure the validity of the model's predictions.

4.3 Experimental result

In the experimental evaluation, the proposed AMTPFNet method was compared with multiple static and temporal knowledge graph embedding methods, including classical models (such as TransE, TTransE, DistMult, ComplEx, HolE) as well as recently proposed temporal reasoning methods (such as CyGNet, RE-GCN, Timetraveler, and TIP). Experiments were conducted on four widely used temporal knowledge graph datasets: YAGO, WIKI, ICEWS18, and GDELT. The results of the experimental comparison of knowledge graph embedding methods are shown in Table 3. The results demonstrate that AMTPFNet achieves superior performance across most evaluation metrics.

Table 3 Experimental comparison of knowledge graph embedding methods

Method	YAGO				WIKI				GDELT				ICEWS18			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	0.373	0.191	0.531	0.622	0.374	0.289	0.443	0.508	0.089	0.000	0.117	0.253	0.117	0.008	0.154	0.344
DistMult	0.466	0.368	0.536	0.627	0.387	0.305	0.457	0.508	0.084	0.033	0.080	0.179	0.216	0.132	0.244	0.379
ComplEx	0.489	0.390	0.563	0.651	0.394	0.311	0.464	0.513	0.080	0.029	0.076	0.175	0.207	0.123	0.234	0.374
HolE	0.379	0.286	0.437	0.531	0.343	0.254	0.411	0.487	0.076	0.028	0.071	0.165	0.107	0.047	0.110	0.232
TTransE	0.318	0.162	0.443	0.538	0.271	0.182	0.325	0.429	0.081	0.000	0.101	0.228	0.115	0.005	0.152	0.342
CyGNet	0.548	0.435	0.613	0.778	0.375	0.284	0.426	0.539	0.187	0.114	0.199	0.328	0.250	0.155	0.286	0.189
RE-GCN	0.821	0.785	0.844	0.885	0.782	0.741	0.812	0.846	0.201	0.141	0.219	0.310	0.325	0.223	0.367	0.527
Timetraveler	0.875	0.852	0.897	0.901	0.746	0.723	0.762	0.777	0.197	0.125	0.209	0.339	0.293	0.213	0.328	0.442
TIP	0.905	0.879	0.930	0.936	0.832	0.791	0.865	0.887	0.211	0.140	0.230	0.347	0.317	0.221	0.359	0.505
AMTPFNet	0.914	0.895	0.932	0.935	0.838	0.805	0.865	0.884	0.219	0.142	0.240	0.370	0.296	0.200	0.337	0.485

Overall, AMTPFNet exhibits particularly strong performance on the YAGO and WIKI datasets. It outperforms all comparison models in key metrics such as MRR and Hits@1, achieving 0.91397 and 0.89489 (YAGO), and 0.83751 and 0.80518 (WIKI), respectively. This demonstrates its strong modeling capability on datasets with rich static structures. On the GDELT dataset, AMTPFNet also outperforms all comparison models, particularly by achieving 0.3701 on the Hits@10 metric, further proving its superiority on event-driven and more dynamic datasets. On the ICEWS18 dataset, AMTPFNet performs slightly below some of the best methods, but still achieves 0.4848 on the Hits@10 metric, showing its strong capability in predicting future facts.

The AMTPFNet model enables the prediction of fault information within the temporal knowledge graph of electromechanical equipment. Table 4 provides a sample comparison between actual object entities and predicted results for the fault data used in constructing the temporal knowledge graph. By utilizing fault information, causal relationships, and timestamps of fault occurrences associated with motor components—such as stators, rotors, drive-end bearings, and non-drive-end bearings—a temporal knowledge graph is established. This facilitates more timely and effective fault prediction, enhancing the accuracy and reliability of predictive outcomes.

The advantages of AMTPFNet mainly lie in its ability to model multi-scale temporal information and in the query-based adaptive edge weighting mechanism. This allows the model to capture short-term high-frequency dynamics while also perceiving long-term evolutionary trends, enabling it to make reasonable judgments across different time spans. Additionally, the introduction of the self-attention mechanism enables the model to flexibly adjust the importance of each edge in the path according to the query context, enhancing the specificity and accuracy of the reasoning.

Table 4 Comparison of actual object entities and predicted results

Time t	Equipment s	Fault information r	Actual fault cause o	Predicted fault cause o'
2020.9.5	Side bearing	None	None	None
2020.9.6	Side bearing	Overheating	Insulation	Insulation
2020.9.6	Side bearing	Overheating	Insulation	Insulation
...	...	...	...	...
2020.10.10	Side bearing	Abnormal noise	Friction	Friction
2020.10.11	Stator	None	Wear	None
2020.10.12	Rotor	Overheating	Insulation	Wear

5. Conclusion

This paper presents an innovative temporal knowledge graph reasoning method, AMTPFNet, which significantly improves the accuracy and robustness of reasoning tasks through techniques such as multi-scale historical graph construction, query-aware temporal path processing, and multi-resolution fusion. The implementation process of AMTPFNet includes the following key steps:

- Multi-scale Historical Graph Construction: The short-term historical graph G_{short} is constructed using continuous sampling, while the long-term historical graph G_{long} is constructed by sampling at intervals Δ.
- Temporal Path Definition and Abstraction: For each scale, a set of paths is extracted under the query $(s, r, ?, t+1)$. The representations $\boldsymbol{z}_{s\to o}^{(i)}$ are then obtained through an aggregation function.
- Query-aware Temporal Path Processing: Multi-layer self-attention message passing is used to update the paths. The update rules are defined in Eqs. 3 and 4, and the final representation is obtained in Eq. 5.
- Multi-resolution Fusion and Scoring Function: The representations of the two scales are fused (Eq. 6), then concatenated with the projection of the query relation and passed through a feed-forward network to compute the score (Eq. 7), which is mapped to a prediction probability (Eq. 8).
- Loss Function Design: The loss function consists of binary cross-entropy loss (Eq. 9), along with relationship orthogonality regularization (Eq. 10) and attention regularization (Eq. 11), resulting in the total loss (Eq. 12).

Experimental results show that AMTPFNet significantly outperforms existing baseline models on the YAGO, WIKI, and GDELT datasets, indicating its strong modeling capability on data with rich static structures and its superiority on event-driven, more dynamic datasets. At the same time, it is noted that there is still room for improvement on more dynamic datasets such as ICEWS18. Future work could focus on finer-grained temporal modeling, such as introducing adaptive time windows or continuous time representations, as well as more efficient structure-aware mechanisms. For instance, exploring online learning or incremental update mechanisms would allow the model to dynamically adapt to rapidly changing temporal data, further optimizing its performance. Finally, by using electromechanical equipment fault data as a case study, this work demonstrates that the proposed model can effectively predict future production outcomes and the quality of process manufacturing operations.

Beyond architectural innovation, AMTPFNet offers significant practical value for industrial workflows by achieving high efficiency in predicting future production facts and assessing process quality. Specifically, the model can infer root causes of failures (e.g., insulation, friction) for motor components, making it well-suited for predictive maintenance strategies. This utility is supported by quantitative evidence: AMTPFNet achieves superior performance on TKG benchmarks, including MRR scores of 0.914 on YAGO and 0.838 on WIKI, and a Hits@10 score of 0.370 on GDELT. Furthermore, the framework's computational efficiency, requiring only a few hours for training and seconds to minutes for inference, demonstrates its feasibility for industrial deployment. However, a key limitation is that its current empirical validation is restricted to electromechanical fault data. Real-world deployment in factories will face major challenges such as heterogeneous data quality, strict real-time requirements, and the necessity for robust incremental update mechanisms.

Funding

National Natural Science Foundation of China (71173177); The UIC Scientific Research Project (R113621H01019); Sichuan Science and Technology Program (2020JDR0076).

Data availability

The data that support the findings of this study are openly available at https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28075, https://datahub.aalto.fi/en/data-sources/the-gdelt-database, https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/ and https://github.com/Lee-zix/RE-GCN/raw/master/data-release.tar.gz

Acknowledgement

An acknowledgement section may be presented after the conclusion, if desired. This heading is not assigned a number. Use Cambria font of 9 pt size.

References

[1] Rosin, F., Forget, P., Lamouri, S., Pellerin, R. (2021). Impact of Industry 4.0 on decision-making in an operational context, *Advances in Production Engineering & Management,* Vol. 16, No. 4, 500-514, doi: 10.14743/apem2021.4.416.

[2] Meng, X., Liu, Q., Yang, C., Zhou, L., Cheung, Y.-M. (2024). A novel deep learning-based robust dual-rate dynamic data modeling for quality prediction, *IEEE Transactions on Industrial Informatics,* Vol. 20, No. 2, 1324-1334, doi: 10.1109/tii.2023.3275700.

[3] Tercan, H., Meisen, T. (2022). Machine learning and deep learning based predictive quality in manufacturing: A systematic review, *Journal of Intelligent Manufacturing*, Vol. 33, 1879-1905, doi: 10.1007/s10845-022-01963-8.

[4] Mlinarič, J., Pregelj, B., Boškoski, P., Dolanc, G., Petrovčič, J. (2024). Optimization of reliability and speed of the end-of-line quality inspection of electric motors using machine learning, *Advances in Production Engineering & Management*, Vol. 19, No. 2, 182-196, doi: 10.14743/apem2024.2.500.

[5] Xie, W., He, J., Huang, F., Ren, J. (2024). Supply chain financial fraud detection based on graph neural network and knowledge graph, *Tehnicki Vjesnik–Technical Gazette,* Vol. 31, No. 6, 2055-2063, doi: 10.17559/TV-20240606001759.

[6] Cao, Y., Lin, X., Wu, Y., Shi, F., Shang, Y., Tan, Q., Zhou, C., Zhang, P. (2024). A data-centric framework of improving graph neural networks for knowledge graph embedding, *World Wide Web,* Vol. 28, No. 1, Article No. 2, doi: 10.1007/s11280-024-01320-0.

[7] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data, In: *Proceedings of NIPS'13: Neural Information Processing Systems,* Lake Tahoe, Nevada, USA,

2787-2795, doi: 10.5555/2999792.2999923.
[8] Wang, Z., Zhang, J., Feng, J., Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes, In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence,* Québec City, Québec, Canada, 1112-1119, doi: 10.1609/aaai.v28i1.8870.
[9] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion, In: *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, USA, 2181-2187, doi: 10.1609/aaai.v29i1.9491.
[10] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G. (2016). Complex embeddings for simple link prediction, *ArXiv,* doi: 10.48550/arXiv.1606.06357.
[11] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S. (2018). Convolutional 2D knowledge graph embeddings, In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,* New Orleans, Lousiana, USA, 1811-1818, doi: 10.1609/aaai.v32i1.11573.
[12] Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M. (2018). Modeling relational data with graph convolutional networks, In: Gangemi, A., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.), *The Semantic Web. ESWC 2018. Lecture notes in computer science,* Vol 10843. Springer, Cham, Switzerland, doi: 10.1007/978-3-319-93417-4_38.
[13] Zhang, S., Zhang, Z., Zhuang, F., Shi, Z., Han, X. (2020). Compressing knowledge graph embedding with relational graph auto-encoder, In: *Proceedings of 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, 366-370, doi: 10.1109/ICEIEC49280.2020.9152323.
[14] Dasgupta, S.S., Ray, S.N., Talukdar, P. (2018). HyTE: Hyperplane-based temporally aware knowledge graph embedding, In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* Brussels, Belgium, 2001-2011, doi: 10.18653/v1/D18-1225.
[15] Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P. (2020). Diachronic embedding for temporal knowledge graph completion, In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence,* New York, USA, 3988-3995, doi: 10.1609/aaai.v34i04.5815.
[16] Liu, H., Meng, J., Sun, S. (2023). Recurrent event networks based on subgraph and attention enhancement, *IEEE Access,* Vol. 11, 130888-130898, doi: 10.1109/ACCESS.2023.3333365.
[17] Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y. (2021). Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks, In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence,* Vol. 35, No. 5, 4732-4740, doi: 10.1609/aaai.v35i5.16604.
[18] Li, Q., Wu, G. (2025). Explainable reasoning over temporal knowledge graphs by pre-trained language model, *Information Processing & Management,* Vol. 62, No. 1, Article No. 103903, doi: 10.1016/j.ipm.2024.103903.
[19] Liu, Q., Huang, M., Feng, S. (2024). An adaptive framework for temporal knowledge graph reasoning with dynamic event-driven dependencies, In*: Proceedings of 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC),* Xiamen, China, 298-304, doi: 10.1109/ICAIRC64177.2024.10900070.
[20] Tan, X., Wang, X., Liu, Q., Xu, X., Yuan, X., Zhang, W. (2024). Paths-over-graph: Knowledge graph empowered large language model reasoning, *ArXiv,* doi: 10.48550/arXiv.2410.14211.
[21] Liu, J., Mao, Q., Jiang, W., Li, J. (2024). KnowFormer: Revisiting transformers for knowledge graph reasoning, *ArXiv*, doi: 10.48550/arXiv.2409.12865.
[22] Chen, K., Wang, Y., Li, Y., Li, A., Yu, H., Song, X. (2024). A unified temporal knowledge graph reasoning model towards interpolation and extrapolation, *ArXiv,* doi: 10.48550/arXiv.2405.18106.
[23] Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., Cheng, X. (2021). 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*, Virtual event, Canada, 408-417, doi: 10.1145/3404835.3462963.
[24] Dong, H., Ning, Z., Wang, P., Qiao, Z., Wang, P., Zhou, Y., Fu, Y. (2023). Adaptive path-memory network for temporal knowledge graph reasoning, *ArXiv,* doi: 10.48550/arXiv.2304.12604.
[25] He, Y., Zhang, P., Liu, L., Liang, Q., Zhang, W., Zhang, C. (2021). HIP network: Historical information passing network for extrapolation reasoning on temporal knowledge graph, In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, Montreal, Canada, 1915-1921, doi: 10.24963/ijcai.2021/264.