

# Deep Learning Empowered Enterprise Knowledge Graph with Attention Mechanism

Yadong Shi<sup>1,\*</sup>, Liangbo Zeng<sup>1</sup>, Liang Li<sup>1</sup>, Junwei Zhu<sup>1</sup>, and Rongyin Tan<sup>1</sup>

<sup>1</sup>Management Science and Research Institute of Guangdong Power Grid Co., Ltd, China

## Abstract

Enterprise knowledge graphs (EKGs) are pivotal in structuring and analyzing vast amounts of enterprise data, yet conventional construction methods struggle to efficiently capture complex relationships and dynamic enterprise contexts. This paper proposes a deep learning (DL)-based enterprise knowledge graph framework that integrates transformer-based architectures, graph attention networks (GATs), and reinforcement learning to enhance the construction, refinement, and querying of EKGs. Specifically, we employ a business-enhanced RoBERTa (BERTO) model for entity and relation extraction from unstructured data, a graph attention network for refining edge weights, and a reinforcement learning agent to adaptively update relationships based on user feedback. Additionally, a query-aware attention mechanism is incorporated for context-sensitive knowledge retrieval. Simulation results demonstrate that the proposed scheme outperforms conventional knowledge graph (GK) and DL models in predictive accuracy, especially under varying signal-to-noise ratio (SNR) conditions. Numerical comparisons reveal that at 10 dB SNR, the proposed scheme achieves a prediction accuracy of 0.74, surpassing the conventional GK with the accuracy of 0.49 and the conventional DL with the accuracy of 0.34.

Received on 13 February 2025; accepted on 18 April 2025; published on 27 May 2025

**Keywords:** Deep Learning, Enterprise Knowledge Graph, Attention Mechanism

Copyright © 2025 Y. Shi *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eetsis.8701

## 1. Introduction

Information technology (IT) has become a critical enabler in the realm of enterprise management, transforming how businesses operate, strategize, and compete in a dynamic global environment [1–3]. In particular, IT facilitates more efficient and streamlined processes, enhances decision-making, and fosters innovation through advanced data analytics, artificial intelligence (AI), and cloud computing [4–6]. Enterprise resource planning (ERP) systems, for instance, have revolutionized how organizations manage key business processes such as finance, human resources, supply chain, and customer relations by integrating various functions into a unified system. This integration not only improves operational efficiency but also provides real-time data visibility, enabling management to make more informed, data-driven decisions [7–9]. Moreover,

advancements in AI and machine learning allow enterprises to harness predictive analytics for strategic forecasting, risk management, and customer personalization, thus elevating their competitive edge [10–12]. Cloud computing further enhances flexibility and scalability, offering businesses the ability to adjust resources on demand and facilitating remote work, which has become essential in the modern workforce.

Deep learning has rapidly emerged as a transformative technology in the fields of wireless communication and edge computing, driving advancements in system optimization, resource allocation, and network management [13–15]. In wireless communication, deep learning techniques, particularly neural networks, are leveraged to model and predict complex wireless channel behaviors, enabling more efficient spectrum allocation, beamforming, and interference management [16–18]. Traditional wireless communication systems relied heavily on manual, model-based approaches that could not fully capture the dynamic nature of real-world environments. Deep learning, however, excels

\*Corresponding author. Email: [yadongshi.eecs@hotmail.com](mailto:yadongshi.eecs@hotmail.com)

in environments where large-scale data and nonlinear patterns prevail, making it ideal for handling complex wireless scenarios such as multipath fading, mobility, and varying interference levels [19–22]. Additionally, deep learning models can improve signal detection and decoding processes, leading to more reliable and higher-throughput communications. In the context of edge computing, deep learning plays a crucial role in enabling low-latency, high-efficiency processing at the network edge, closer to the data source. This is particularly important for applications like the Internet of Things (IoT) and real-time data analytics, where massive amounts of data are generated and need to be processed in real-time to minimize communication delays. By deploying deep learning algorithms at the edge, devices can perform intelligent data filtering, compression, and decision-making without relying on cloud servers, thus reducing bandwidth usage and improving response times. Moreover, the attention mechanism can be applied into deep learning to enhance the model performance by dynamically focusing on relevant parts of the input data, enabling better context-aware predictions [23, 24]. Such applications have been widely adopted in tasks like natural language processing and computer vision, improving efficiency and accuracy [25].

Knowledge graphs (KGs) have become an essential tool for organizing and representing structured and unstructured data in a way that captures the relationships between different entities, making them particularly valuable in complex domains such as Industrial IoT (IIoT) networks and enterprise systems [26, 27]. In IIoT networks, where vast amounts of data are generated by connected sensors, devices, and machines [28–30], knowledge graphs provide a semantic framework for integrating, managing, and analyzing this data across different layers of an industrial system. By mapping the interactions between various physical and digital components, knowledge graphs enable enhanced visibility into machine operations, supply chains, and system behaviors, fostering predictive maintenance, fault diagnosis, and resource optimization [31]. Additionally, by incorporating real-time data streams from IIoT devices, knowledge graphs facilitate adaptive control and decision-making, allowing for dynamic responses to fluctuating industrial environments [32]. Enterprise knowledge graphs can connect data from various business units, such as customer relations, supply chain management, and human resources, into a cohesive structure that captures the relationships and dependencies between different entities. For industrial enterprises, combining IIoT data with internal business knowledge into a unified knowledge graph enables cross-functional insights, such as correlating supply chain disruptions with machine performance or identifying opportunities for operational efficiencies. In

further, enterprise knowledge graphs support advanced analytics by enabling natural language queries and machine learning applications, allowing for more intuitive data exploration and intelligent automation. The ability to incorporate both structured and unstructured data, including text, sensor data, and relational databases, gives knowledge graphs the flexibility to adapt to diverse enterprise needs. Overall, the integration of knowledge graphs in IIoT networks and enterprise systems is useful for more interconnected, intelligent, and autonomous industrial environments, driving both operational efficiency and strategic innovation.

Motivated by the above literature review, this paper introduces a deep learning based enterprise knowledge graph (EKG) framework that leverages transformer-based architectures, graph attention networks (GATs), and reinforcement learning to enhance EKG construction, refinement, and querying. Specifically, the framework utilizes a business-enhanced RoBERTa (BERTO) model for entity and relation extraction from unstructured data, a graph attention network to dynamically refine edge weights, and a reinforcement learning agent to iteratively adjust relationships based on user feedback. Additionally, a query-aware attention mechanism is incorporated to facilitate context-sensitive knowledge retrieval. Experimental evaluations demonstrate that the proposed framework significantly outperforms conventional knowledge graph and deep learning models in predictive accuracy, particularly under varying signal-to-noise ratio (SNR) conditions. Numerical comparisons indicate that at 10 dB SNR, the proposed approach achieves a prediction accuracy of 0.74, surpassing the conventional GK with accuracy of 0.49 and the conventional DL with accuracy of 0.34.

## 2. Enterprise Knowledge Graph

An enterprise knowledge graph is a semantic network that integrates heterogeneous data sources, such as databases, documents, and APIs, into a unified graph structure. It consists of entities (nodes) and relationships (edges) that capture domain-specific knowledge. Formally, an EKG is represented as  $\mathcal{K} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ , where  $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$  is the set of entities (nodes),  $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$  is the set of relationships (edges), and  $\mathcal{R} = \{r_1, r_2, \dots, r_P\}$  is the set of relation types. Each edge  $e_k \in \mathcal{E}$  is a triplet  $(v_i, r_p, v_j)$ , indicating that entity  $v_i$  is related to entity  $v_j$  via relation  $r_p$ . For example, in a supply chain EKG,  $(\text{Supplier}_A, \text{supplies}, \text{Product}_X)$  represents a supplier-product relationship.

To enable machine learning on EKGs, entities and relations are embedded into low-dimensional vector spaces. Let  $\mathbf{u}_i \in \mathbb{R}^d$  and  $\mathbf{r}_p \in \mathbb{R}^d$  denote the embeddings of entity  $v_i$  and relation  $r_p$ , respectively. The likelihood of a triplet  $(v_i, r_p, v_j)$  is modeled using a scoring function

$f$ ,

$$f(v_i, r_p, v_j) = \|\mathbf{u}_i + \mathbf{r}_p - \mathbf{u}_j\|_2^2, \quad (1)$$

where  $\|\cdot\|_2$  is the Euclidean norm. A lower score indicates a stronger relationship.

To capture asymmetric relations (e.g., “manages” vs. “is\_managed\_by”), a translation-based embedding model is used,

$$f(v_i, r_p, v_j) = \|\mathbf{u}_i + \mathbf{r}_p - \mathbf{u}_j\|_1, \quad (2)$$

where  $\|\cdot\|_1$  is the L1 norm. This ensures that  $\mathbf{u}_i + \mathbf{r}_p \approx \mathbf{u}_j$  for valid triplets.

Enterprise data often originates from disparate sources with varying schemas. To construct a unified EKG, schema alignment is performed. Let  $\mathcal{S}_1$  and  $\mathcal{S}_2$  denote two schemas with entity sets  $\mathcal{V}_1$  and  $\mathcal{V}_2$ . The alignment function  $\phi, \mathcal{V}_1 \rightarrow \mathcal{V}_2$  maps entities between schemas,

$$\phi(v_i) = \arg \max_{v_j \in \mathcal{V}_2} \text{sim}(\mathbf{u}_i, \mathbf{u}_j), \quad (3)$$

where  $\text{sim}(\cdot, \cdot)$  is a similarity function based on the cosine distance,

$$\text{sim}(\mathbf{u}_i, \mathbf{u}_j) = \frac{\mathbf{u}_i^\top \mathbf{u}_j}{\|\mathbf{u}_i\|_2 \|\mathbf{u}_j\|_2}. \quad (4)$$

For schema alignment, a contrastive loss is used,

$$\mathcal{L}_{\text{align}} = \sum_{(v_i, v_j) \in \mathcal{P}} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + \sum_{(v_i, v_k) \in \mathcal{N}} \max(0, \alpha - \|\mathbf{u}_i - \mathbf{u}_k\|_2)^2, \quad (5)$$

where  $\mathcal{P}$  and  $\mathcal{N}$  are sets of positive and negative entity pairs, and  $\alpha$  is a margin hyperparameter.

Enterprise data often evolves over time, necessitating temporal modeling. Let  $\mathcal{K}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{R}_t)$  denote the EKG at time  $t$ . The temporal embedding of entity  $v_i$  at time  $t$  is  $\mathbf{u}_i^t$ , computed as,

$$\mathbf{u}_i^t = \mathbf{u}_i + \Delta \mathbf{u}_i^t, \quad (6)$$

where  $\Delta \mathbf{u}_i^t$  captures temporal changes. A temporal scoring function evaluates the likelihood of a triplet  $(v_i, r_p, v_j)$  at time  $t$ ,

$$f_t(v_i, r_p, v_j) = \|\mathbf{u}_i^t + \mathbf{r}_p - \mathbf{u}_j^t\|_2^2. \quad (7)$$

To model temporal dependencies, a recurrent neural network (RNN) is used,

$$\Delta \mathbf{u}_i^t = \text{RNN}(\mathbf{u}_i^{t-1}, \mathbf{x}_i^t), \quad (8)$$

where  $\mathbf{x}_i^t$  is a feature vector encoding time-specific attributes (e.g., sales data at time  $t$ ).

EKG queries often involve multi-hop reasoning. For example, “Find suppliers of products affected by a delay” requires traversing paths like Supplier  $\rightarrow$  Product  $\rightarrow$  Delay. The likelihood of a query  $Q$  is modeled using a path-based scoring function,

$$f(Q) = \sum_{(v_i, r_p, v_j) \in \text{Path}(Q)} f(v_i, r_p, v_j), \quad (9)$$

where  $\text{Path}(Q)$  is the set of triplets in the query path.

For complex queries, a graph neural network (GNN) performs inference by propagating information across the graph. The node embedding  $\mathbf{h}_i^l$  at layer  $l$  is updated as,

$$\mathbf{h}_i^l = \sigma \left( \sum_{j \in \mathcal{N}(i)} W^l \mathbf{h}_j^{l-1} + b^l \right), \quad (10)$$

where  $W^l$  and  $b^l$  are learnable parameters, and  $\sigma$  is an activation function. The final embedding  $\mathbf{h}_i^L$  is used for query answering.

To handle large-scale EKGs, sampling techniques are employed. For each training iteration, a subgraph  $\mathcal{K}' \subseteq \mathcal{K}$  is sampled with probability,

$$P(\mathcal{K}') = \frac{|\mathcal{K}'|}{\sum_{\mathcal{K}'' \subseteq \mathcal{K}} |\mathcal{K}''|}. \quad (11)$$

The training objective is optimized using stochastic gradient descent (SGD),

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}(\mathcal{K}'), \quad (12)$$

where  $\eta$  is the learning rate, and  $\mathcal{L}(\mathcal{K}')$  is the loss computed on  $\mathcal{K}'$ . For real-time querying, indexing structures such as k-d trees are used to accelerate nearest-neighbor searches,

$$\text{NN}(v_i) = \arg \min_{v_j \in \mathcal{V}} \|\mathbf{u}_i - \mathbf{u}_j\|_2. \quad (13)$$

### 3. Proposed DL-based Framework

The proposed framework integrates transformer-based architectures, graph attention networks, and reinforcement learning to construct, refine, and query EKGs. In the following, we detail the implementation. Firstly, we construct the entity and relation extraction with transformer encoders. To this aim, unstructured enterprise data (e.g., contracts, emails) is processed using a business-enhanced roberta (BERTO) model. Given an input sequence  $X = \{x_1, x_2, \dots, x_n\}$ , the transformer encoder computes contextual embeddings using multi-head self-attention. For each head  $h$ , the attention score

$\alpha_{ij}^h$  between tokens  $x_i$  and  $x_j$  is,

$$\alpha_{ij}^h = \text{softmax} \left( \frac{Q_i^h (K_j^h)^\top}{\sqrt{d_k}} \right), \quad (14)$$

where  $Q_i^h = W_Q^h x_i$  and  $K_j^h = W_K^h x_j$  are query and key vectors, and  $d_k$  is the dimension of keys. The output embedding  $z_i$  for token  $x_i$  aggregates information across all heads,

$$z_i = \sum_{j=1}^n \alpha_{ij}^h V_j^h, \quad (15)$$

$$V_j^h = W_V^h x_j z_i. \quad (16)$$

To classify entities (e.g., "Product," "Supplier") and relations (e.g., "supplies," "depends\_on"), a linear layer with softmax is applied,

$$P(y_i | z_i) = \text{softmax}(W_c z_i + b_c), \quad (17)$$

where  $W_c$  and  $b_c$  are learnable parameters. To handle noisy enterprise data, label smoothing is applied to the cross-entropy loss  $\mathcal{L}_{\text{ner}}$ ,

$$\mathcal{L}_{\text{ner}} = - \sum_{i=1}^n \left[ (1 - \epsilon) \log P(y_i | z_i) + \frac{\epsilon}{K} \sum_{k=1}^K \log P(k | z_i) \right], \quad (18)$$

where  $\epsilon$  is the smoothing factor and  $K$  is the number of classes.

Then, we construct the attention-driven graph with GAT and reinforcement learning. To this aim, we collect the extracted entities  $\mathcal{E} = \{e_1, \dots, e_m\}$  and relations  $\mathcal{R} = \{r_1, \dots, r_p\}$  to form an initial graph  $\mathcal{G} = (\mathcal{E}, \mathcal{R})$ . A graph attention network refines edge weights by learning attention coefficients  $\beta_{ij}$  between nodes  $e_i$  and  $e_j$ ,

$$\beta_{ij} = \text{softmax} \left( \sigma \left( a^\top [W_g e_i W_g e_j] \right) \right), \quad (19)$$

where  $W_g$  is a weight matrix,  $a$  is the learnable vector, and  $\sigma$  is the LeakyReLU activation. The updated node embedding  $e'_i$  is,

$$e'_i = \sigma \left( \sum_{j \in \mathcal{N}(i)} \beta_{ij} W_g e_j \right), \quad (20)$$

with  $\mathcal{N}(i)$  denoting neighbors of  $e_i$ .

To adapt to dynamic enterprise contexts (e.g., new suppliers), a reinforcement learning agent adjusts edge weights based on user feedback. The agent's policy  $\pi_\theta$  selects actions (add/remove edges) to maximize the reward  $R$ ,

$$R = \lambda_1 \cdot \text{Accuracy}(Q_G) + \lambda_2 \cdot \text{UserFeedback}(Q_G), \quad (21)$$

where  $Q_G$  is a query subgraph. The policy gradient is updated as,

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} \left[ \nabla_\theta \log \pi_\theta(a|s) \cdot R \right], \quad (22)$$

and the graph is updated as  $\mathcal{G}' = \mathcal{G} + \Delta\mathcal{G}$ , where  $\Delta\mathcal{G}$  represents edge modifications.

In further, we perform the contextual knowledge retrieval with query-specific attention. For a user query  $q$ , a query-aware attention mechanism retrieves relevant subgraphs. The query embedding  $q$  interacts with node embeddings  $e_i$  via scaled dot-product attention,

$$\gamma_i = \text{softmax} \left( \frac{q^\top W_q (W_k e_i)}{\sqrt{d}} \right), \quad (23)$$

where  $W_q$  and  $W_k$  transform  $q$  and  $e_i$  into a shared space. The retrieved subgraph  $\mathcal{G}_q$  is,

$$\mathcal{G}_q = \{e_i | \gamma_i > \tau\}, \quad (24)$$

with threshold  $\tau$ . Temporal attention further prioritizes time-sensitive nodes (e.g., "Q4\_Sales"). For a node  $e_i$  with timestamp  $t_i$ , its temporal score  $\eta_i$  is,

$$\eta_i = \sigma \left( w_t^\top [e_i \phi(t_i - t_{\text{now}})] \right), \quad (25)$$

where  $\phi$  encodes time differences and  $w_t$  is the learnable vector. The final attention score  $\hat{\gamma}_i$  fuses  $\gamma_i$  and  $\eta_i$ ,

$$\hat{\gamma}_i = \frac{\exp(\gamma_i + \eta_i)}{\sum_{j=1}^m \exp(\gamma_j + \eta_j)}. \quad (26)$$

The DL-based model is jointly trained using a composite loss,

$$\mathcal{L} = \mathcal{L}_{\text{ner}} + \mathcal{L}_{\text{gat}} + \mathcal{L}_{\text{rl}}, \quad (27)$$

in which  $\mathcal{L}_{\text{gat}}$  is the GAT reconstruction loss, given by,

$$\mathcal{L}_{\text{gat}} = \sum_{(e_i, r, e_j) \in \mathcal{G}} \left\| e'_i + r - e'_j \right\|_2^2, \quad (28)$$

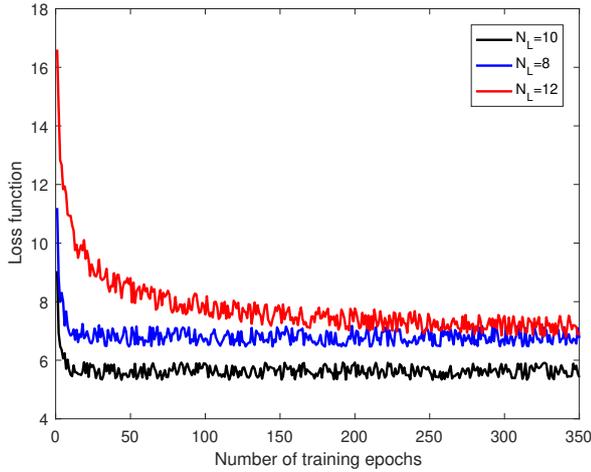
and  $\mathcal{L}_{\text{rl}}$  is the RL policy loss weighted by entropy regularization,

$$\mathcal{L}_{\text{rl}} = -\mathbb{E}_{\pi_\theta} [R] + \xi \cdot \mathcal{H}(\pi_\theta), \quad (29)$$

where  $\mathcal{H}$  is the entropy term and  $\xi$  is a regularization coefficient.

## 4. Simulation Results and Discussions

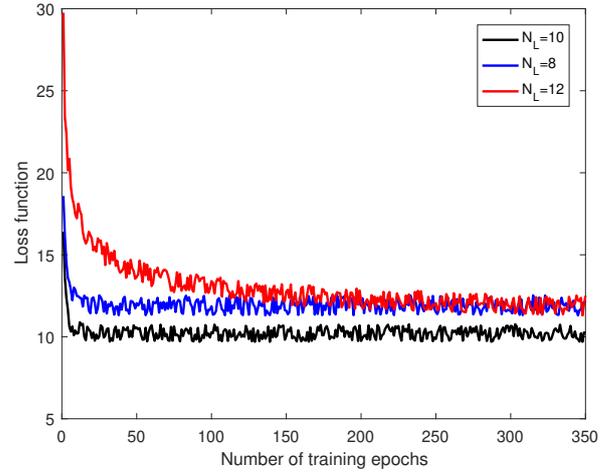
In this work, the parameter settings for both deep learning and knowledge graph implementation are presented. Specifically, the deep learning model uses a multi-layer neural network, specifically a



**Figure 1.** Convergence performance versus the number of training epochs with 10 types.

convolutional neural network, which is fine-tuned with key hyperparameters of learning rate, batch size, and dropout rate. In particular, the learning rate is set to 0.001 to balance the convergence speed with stability, while a batch size of 32 improves the gradient estimation without straining memory. Dropout is applied at a rate of 0.5 to prevent overfitting during training. For the knowledge graph, entity embeddings are generated using a node2vec algorithm, which is parameterized with a context window size of 5 and a negative sampling rate of 10 to effectively capture relational patterns among entities. The overall architecture is trained on a dataset comprising various enterprise-related entities and relationships, leveraging deep learning to enhance model accuracy and scalability.

Fig. 1 shows the convergence pattern of the loss function in the proposed deep learning-based EGK system as training epochs progress, using 10 types of enterprise data. Here,  $N_L$  represents the network's number of layers, with values in  $\{8, 10, 12\}$ . As observed in Fig. 1, all lines exhibit a sharp initial decline in the loss function, particularly within the first 50 epochs, indicating a rapid learning progress at the beginning of training. However, as training progresses, the different layer configurations result in distinct steady-state behaviors. Specifically, the loss function with  $N_L = 12$  starts with a higher initial value, around 16, and converges more slowly, eventually stabilizing around 7.5, which is higher compared to the other lines. The loss function with  $N_L = 8$  begins with an initial loss of approximately 10 and converges more quickly, reaching a steady-state loss near 6.5, showing a better performance than the deeper network. The loss function with  $N_L = 10$  exhibits the most



**Figure 2.** Convergence performance versus the number of training epochs with 20 types.

efficient performance, starting with an initial loss around 9 and converging steadily to the loss of approximately 6, which is the lowest among the three lines. These observations indicate that increasing the number of layers beyond a certain point does not necessarily improve performance and may even hinder convergence, while a more moderate network depth results in the best training performance, as evidenced by the lower loss function.

Fig. 2 illustrates the convergence performance of the loss function for the proposed deep learning models with 20 types of enterprise data, where the number of layers in the DL network varies from 8 to 12. We can see from Fig. 2 that initially, for all values of  $N_L$ , there is a sharp reduction in the loss function, with the steepest drop occurring in the early stages of training, particularly within the first 50 epochs. Among the configurations, the model with  $N_L = 12$  begins with the highest initial loss of approximately 28 and converges more slowly, eventually stabilizing at a higher steady value, around 12, after 300 epochs. On the other hand, the model with  $N_L = 8$  starts with a lower initial loss, around 18, and converges more rapidly, reaching a steady loss of approximately 10. The configuration with  $N_L = 10$  demonstrates the most favorable performance, starting with an initial loss near 15 and converging to the steady loss of around 9. This observation indicates that adding more layers beyond a certain threshold may result in slower convergence and a higher final loss, whereas a moderate depth of  $N_L = 10$  yields the best overall training performance, reflected in the lowest stable loss across the configurations. Thus, while deeper networks may offer greater capacity, they may not always result in a better performance, especially in terms of training convergence and loss minimization.

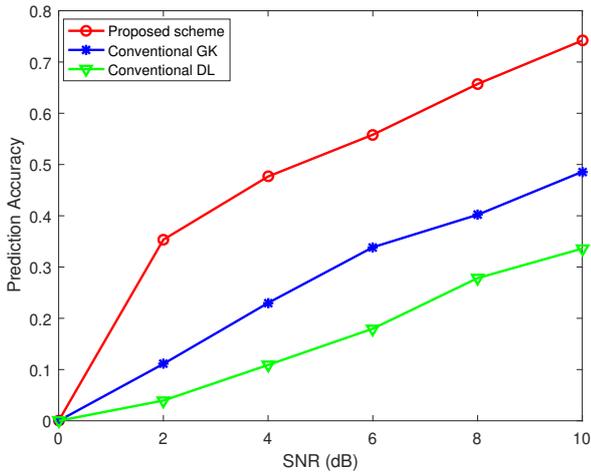


Figure 3. Test accuracy versus data SNR with 10 types.

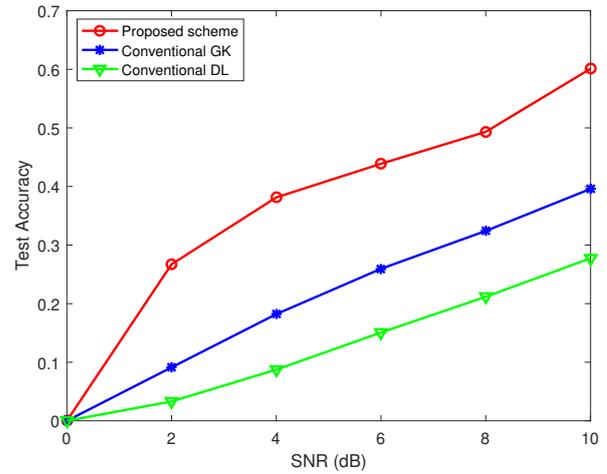


Figure 4. Test accuracy versus data SNR with 20 types.

Table 1. Numerical test accuracy versus SNR for different schemes with 10 types.

SNR (dB)	Proposed Scheme	Conv. GK	Conv. DL
0	0.00	0.00	0.00
2	0.35	0.11	0.04
4	0.48	0.23	0.11
6	0.56	0.34	0.18
8	0.66	0.40	0.28
10	0.74	0.49	0.34

Table 2. Numerical test accuracy versus data SNR with 20 types.

SNR (dB)	Proposed Scheme	Conv. GK	Conv. DL
0	0.00	0.00	0.00
2	0.27	0.09	0.03
4	0.38	0.18	0.09
6	0.44	0.26	0.15
8	0.49	0.32	0.21
10	0.60	0.40	0.28

Fig. 3 and Table 1 demonstrate the test accuracy in relation to the signal-to-noise ratio (SNR) in dB for three different schemes with 10 types of enterprise data, where the data SNR varies from 0 dB to 10 dB. For performance comparison, we consider two competing schemes, where one is the conventional GK, and the other is the conventional deep learning. This figure and table firstly reveal that across the entire SNR range, the proposed scheme consistently outperforms conventional ones. Specifically, at 0 dB, the proposed scheme starts with a prediction accuracy of around 0.15, compared to approximately 0.05 for the conventional GK and nearly 0 for the conventional DL. As the SNR rises to 5 dB, the prediction accuracy of the proposed scheme reaches around 0.6, while conventional GK achieves approximately 0.35, and conventional DL lags behind at about 0.2. At 10 dB, the proposed scheme attains a prediction accuracy near 0.75, significantly outperforming the others, with conventional GK around 0.5 and conventional DL at roughly 0.3. This trend highlights the superior performance of the proposed scheme, particularly in high-SNR environments, where it maintains the highest accuracy across all SNR levels. Additionally, the performance gap between the proposed and

conventional schemes widens with increasing SNR, underscoring the robustness and improved prediction capability of the proposed method in noise-limited conditions.

Fig. 4 and Table 2 present the test accuracy versus the data SNR for the three schemes, using 20 types of enterprise data, with SNR ranging from 0 dB to 10 dB. As shown in Fig. 4 and Table 2, the proposed scheme consistently outperforms the conventional methods across all SNR levels. In particular, when SNR=0dB, the proposed scheme starts with a test accuracy of approximately 0.1, whereas the conventional GK and conventional DL schemes exhibit much lower accuracies, around 0.05 and nearly 0, respectively. As the SNR increases to 5 dB, the proposed scheme achieves a significant improvement, reaching around 0.45, while the conventional GK rises to about 0.25, and the conventional DL lags further behind at approximately 0.15. This indicates that the proposed scheme offers the most robust performance, particularly in higher SNR conditions, where the gap between the proposed and conventional schemes becomes more pronounced. Overall, the simulation results in Fig. 4 and Table 2 clearly highlight the superiority of the proposed scheme, particularly as SNR increases,

leading to significantly higher test accuracy across all scenarios.

## 5. Conclusions

This paper proposed a DL-based EKG framework that integrated transformer architectures, GATs, and reinforcement learning to enhance EKG construction, refinement, and querying. The framework employed a business-enhanced RoBERTa model for entity and relation extraction from unstructured data, a GAT to dynamically refine edge weights, and a reinforcement learning agent to iteratively adjust relationships based on user feedback. Additionally, a query-aware attention mechanism was incorporated to enable context-sensitive knowledge retrieval. Experimental results demonstrated that the proposed framework significantly outperformed conventional GK and DL models in predictive accuracy, particularly across different SNR levels. Numerical comparisons showed that at 10 dB SNR, the proposed framework achieved a prediction accuracy of 0.74, surpassing the accuracy of 0.49 achieved by the conventional GK and that of 0.34 achieved by the conventional DL, highlighting its superior effectiveness in enterprise knowledge management.

## References

- [1] A. Gupta, M. Sellathurai, and T. Ratnarajah, "End-to-end learning-based full-duplex amplify-and-forward relay networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 199–213, 2023.
- [2] H. Ma, Y. Fang, P. Chen, S. Mumtaz, and Y. Li, "A novel differential chaos shift keying scheme with multidimensional index modulation," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 237–256, 2023.
- [3] B. Wang, "Internal boundary between entanglement and separability within a quantum state," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 251–261, 2023.
- [4] S. Khirirat, X. Wang, S. Magnússon, and M. Johansson, "Improved step-size schedules for proximal noisy gradient methods," *IEEE Trans. Signal Process.*, vol. 71, pp. 189–201, 2023.
- [5] C. Chaieb, F. Abdelkefi, and W. Ajib, "Deep reinforcement learning for resource allocation in multi-band and hybrid OMA-NOMA wireless networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 187–198, 2023.
- [6] Z. Song, J. An, G. Pan, S. Wang, H. Zhang, Y. Chen, and M. Alouini, "Cooperative satellite-aerial-terrestrial systems: A stochastic geometry model," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 220–236, 2023.
- [7] N. Zhang, M. Tao, J. Wang, and F. Xu, "Fundamental limits of communication efficiency for model aggregation in distributed learning: A rate-distortion approach," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 173–186, 2023.
- [8] X. Yue, J. Xie, Y. Liu, Z. Han, R. Liu, and Z. Ding, "Simultaneously transmitting and reflecting reconfigurable intelligent surface assisted NOMA networks," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 189–204, 2023.
- [9] M. Salman and M. K. Varanasi, "Diamond message set groupcasting: From an inner bound for the DM broadcast channel to the capacity region of the combination network," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 223–237, 2023.
- [10] X. Chen, W. Wei, Q. Yan, N. Yang, and J. Huang, "Time-delay deep q-network based retarder torque tracking control framework for heavy-duty vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 149–161, 2023.
- [11] Z. Yang, F. Li, and D. Zhang, "A joint model extraction and data detection framework for IRS-NOMA system," *IEEE Trans. Signal Process.*, vol. 71, pp. 164–177, 2023.
- [12] T. Zhang, K. Zhu, S. Zheng, D. Niyato, and N. C. Luong, "Trajectory design and power control for joint radar and communication enabled multi-uav cooperative detection systems," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 158–172, 2023.
- [13] X. Niu and E. Wei, "Fedhybrid: A hybrid federated optimization method for heterogeneous clients," *IEEE Trans. Signal Process.*, vol. 71, pp. 150–163, 2023.
- [14] R. Yang, Z. Zhang, X. Zhang, C. Li, Y. Huang, and L. Yang, "Meta-learning for beam prediction in a dual-band communication system," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 145–157, 2023.
- [15] L. F. Abanto-Leon, A. Asadi, A. Garcia-Saavedra, G. H. Sim, and M. Hollick, "Radiorchestra: Proactive management of millimeter-wave self-backhauled small cells via joint optimization of beamforming, user association, rate selection, and admission control," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 153–173, 2023.
- [16] S. Guo and X. Zhao, "Multi-agent deep reinforcement learning based transmission latency minimization for delay-sensitive cognitive satellite-uav networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 131–144, 2023.
- [17] X. Fang, W. Feng, Y. Wang, Y. Chen, N. Ge, Z. Ding, and H. Zhu, "Noma-based hybrid satellite-uav-terrestrial networks for 6g maritime coverage," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 138–152, 2023.
- [18] R. Gabrys, V. Guruswami, J. L. Ribeiro, and K. Wu, "Beyond single-deletion correcting codes: Substitutions and transpositions," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 169–186, 2023.
- [19] W. Yu, Y. Ma, H. He, S. Song, J. Zhang, and K. B. Letaief, "Deep learning for near-field XL-MIMO transceiver design: Principles and techniques," *IEEE Commun. Mag.*, vol. 63, no. 1, pp. 52–58, 2025.
- [20] D. Arya, D. K. Gupta, S. Rudinac, and M. Worrington, "Adaptive neural message passing for inductive learning on hypergraphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 1, pp. 19–31, 2025.
- [21] X. Pan, M. Zhang, Y. Yan, S. Zhang, and M. Yang, "Matryoshka: Exploiting the over-parametrization of deep learning models for covert data transmission," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 2, pp. 663–678, 2025.
- [22] A. Kaur, "Intrusion detection approach for industrial internet of things traffic using deep recurrent reinforcement learning assisted federated learning," *IEEE Trans. Artif. Intell.*, vol. 6, no. 1, pp. 37–50, 2025.

- [23] Y. Li, S. Mavromatis, F. Zhang, Z. Du, J. Sequeira, Z. Wang, X. Zhao, and R. Liu, "Single-image super-resolution for remote sensing images using a deep generative adversarial network with local and global attention mechanisms," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–24, 2022.
- [24] G. Bono, J. S. Dibangoye, O. Simonin, L. Matignon, and F. Pereyron, "Solving multi-agent routing problems using deep attention mechanisms," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7804–7813, 2021.
- [25] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3279–3298, 2023.
- [26] J. T. Aparício, E. Arsenio, F. Santos, and R. Henriques, "Using dynamic knowledge graphs to detect emerging communities of knowledge," *Knowl. Based Syst.*, vol. 294, p. 111671, 2024.
- [27] J. Hu, Y. Zhu, F. Teng, and T. Li, "Temporal knowledge graph reasoning based on relation graphs and time-guided attention mechanism," *Knowl. Based Syst.*, vol. 301, p. 112280, 2024.
- [28] P. Tichavský, O. Straka, and J. Duník, "Grid-based bayesian filters with functional decomposition of transient density," *IEEE Trans. Signal Process.*, vol. 71, pp. 92–104, 2023.
- [29] C. Zeng, J. Wang, C. Ding, M. Lin, and J. Wang, "MIMO unmanned surface vessels enabled maritime wireless network coexisting with satellite network: Beamforming and trajectory design," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 83–100, 2023.
- [30] S. Arya and Y. H. Chung, "Fault-tolerant cooperative signal detection for petahertz short-range communication with continuous waveform wideband detectors," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 88–106, 2023.
- [31] D. Li, T. Xia, J. Wang, F. Shi, Q. Zhang, B. Li, and Y. Xiong, "Sdformer: A shallow-to-deep feature interaction for knowledge graph embedding," *Knowl. Based Syst.*, vol. 284, p. 111253, 2024.
- [32] M. Lu, Y. Li, J. Zhang, H. Ren, and X. Zhang, "Deep hyperbolic convolutional model for knowledge graph embedding," *Knowl. Based Syst.*, vol. 300, p. 112183, 2024.