



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642

<https://www.inderscience.com/ijict>

A fusion architecture of heterogeneous graph neural network and reinforcement learning for business innovation decision-making

Fangyi Gu, Chenlu Jia

DOI: [10.1504/IJICT.2026.10076453](https://doi.org/10.1504/IJICT.2026.10076453)

Article History:

Received:	14 September 2025
Last revised:	26 November 2025
Accepted:	28 November 2025
Published online:	02 March 2026

A fusion architecture of heterogeneous graph neural network and reinforcement learning for business innovation decision-making

Fangyi Gu and Chenlu Jia*

Management College,
Shenyang Institute of Science and Technology,
Shenyang, 110167, China
Email: 13109876697@163.com
Email: jcl_0023@sina.com

*Corresponding author

Abstract: Business innovation decision-making hinges on the integration of insights from complex, multi-source heterogeneous data, which can be naturally represented using heterogeneous graphs. Traditional approaches often fall short in capturing the rich semantics and dynamic nature of such structures. This paper proposes a novel framework for business innovation decision-making by integrating heterogeneous graph neural networks with reinforcement learning. The model employs meta-path-enhanced Heterogeneous graph neural networks to perceive environments and learn rich node and graph representations. A dual-level attention mechanism (node and semantic levels) adaptively fuses heterogeneous information. A hybrid reward function combining immediate returns and long-term innovation potential mitigates reward sparsity and promotes sustainable optimisation. Extensive experiments on real and simulated business graphs show the framework outperforms state-of-the-art methods, achieving a 20% higher cumulative return, 85% accuracy in innovation pathway identification, and significantly improved strategic adaptability in dynamic pricing.

Keywords: heterogeneous graph neural networks; reinforcement learning; business innovation decision-making; meta-paths; reward function design.

Reference to this paper should be made as follows: Gu, F. and Jia, C. (2026) 'A fusion architecture of heterogeneous graph neural network and reinforcement learning for business innovation decision-making', *Int. J. Information and Communication Technology*, Vol. 27, No. 13, pp.20–37.

Biographical notes: Fangyi Gu received her Master's degree at Nanjing Agricultural University in 2016. She is currently a Lecturer in the Management College at Shenyang Institute of Science and Technology. Her research interests include asset valuation and data asset management.

Chenlu Jia received her Master's degree at Liaoning University of Assets Evaluation in 2014. She is currently an Associate Professor in the Management College at Shenyang Institute of Science and Technology. Her research interests focus on enterprise value evaluation and digital asset management.

1 Introduction

In today's global marketplace, business innovation is crucial. It helps enterprises maintain their core competitiveness. These strategic decisions rely on deep insights. The insights come from multi-source heterogeneous data. This data includes user, product, supply chain, and market dynamics. Such data can be naturally represented as heterogeneous graph structures. Heterogeneous graphs contain diverse node and edge types. This diversity greatly enriches information dimensions. However, it also creates substantial modelling challenges (Zhu et al., 2024).

Researchers often apply homogeneous graph neural networks (HGNNs) to these problems. Heterogeneous graphs can more comprehensively capture the complex interactions within real-world business environments through multi-type nodes and relationships. Compared to homogeneous graphs, they not only express richer semantic information but also integrate multi-source data, thereby enhancing the completeness and accuracy of decision-making information and supporting finer-grained business insights. For example, one study uses a simplified multi-view HGNNs. It completes multilingual knowledge graphs. This method enables knowledge transfer through entity alignment (Dong et al., 2024). Another study focuses on training data. It dynamically selects data to mitigate label noise. This process enhances model robustness under noisy conditions (Wu et al., 2024). Another proposed approach is an adaptive HGNNs. It predicts spatial gene expression from tissue images. This method outperforms existing techniques (Song et al., 2024). In convolutional neural networks, researchers evaluate extraction tasks. They do this by adjusting filter sizes for better efficiency (Tran et al., 2024). Finally, a new method addresses data augmentation in HGNNs. It is a multi-strategy adaptive approach (Juan et al., 2024).

Business innovation is a dynamic process. It involves sequential decision-making. The goal is to maximise long-term returns. This goal aligns closely with a paradigm in artificial intelligence. That paradigm is reinforcement learning (RL). This study combines RL with active learning strategies to enhance the efficiency and effectiveness of semantic segmentation tasks (Rad and Armenakis, 2024). The study proposes a RL method based on genetic algorithms to enhance the performance of facial expression recognition (Rad and Armenakis, 2024). This study proposes a RL-based memory algorithm to address the energy-efficient two-stage flexible job shop scheduling problem in distributed systems (Geng et al., 2024). Combining RL with digital twins, we optimise production scheduling through a digital model platform to enhance production efficiency and adaptability (Seipolt et al., 2024). A more accurate deep RL technique is put forth for cloud computing work scheduling (Hamzehkhani et al., 2024).

The study proposes an adaptive meta-path selection method named AdaMH for circRNA-disease association prediction (Ma et al., 2024). This study suggests a meta-learning-based method for cold-start suggestion that identifies and aggregates users (Xing et al., 2024). This research may combine meta-path mining to enhance information retrieval capabilities in heterogeneous networks (Xiao et al., 2024). This paper combines meta-path representation learning with heterogeneous graph neural network technology for medical data fusion and disease diagnosis (Luo et al., 2024). This paper addresses the challenges faced by traditional methods when handling large-scale pattern graphs by performing meta-path learning in complex heterogeneous information networks (HIN) through RL and pattern-level representations (Liu et al., 2024).

The core question addressed in this paper is: How to construct an end-to-end integrated architecture that learns effective representations from dynamic business heterogeneous graphs and makes sequential decisions based on these representations to maximise long-term business value. This study suggests using RL and HGNNs for business innovation decision-making (HGNN-RL4BID) in order to overcome the aforementioned difficulties.

The main innovations and contributions of this work include:

- 1 Proposes a novel end-to-end fusion architecture that innovatively employs meta-path-enhanced HGNNs as an environment perceiver. This provides dynamic environmental state representations for RL agents, unifying deep environmental understanding with sequence decision optimisation.
- 2 Designs a dual-layer attention mechanism that adaptively learns the importance of different neighbours and relational paths at both node-level and semantic-level (meta-path), enabling precise and interpretable information fusion.
- 3 Addresses the reward sparsity problem by designing a domain-knowledge-based hybrid reward function that incorporates intermediate metrics to provide more timely and rich learning signals, significantly improving training efficiency and stability.
- 4 Comprehensive experiments on real and simulated datasets demonstrate that our model significantly outperforms existing baselines in key metrics such as cumulative reward and accuracy.

2 Theoretical and integration foundations of HGNNs-RL

2.1 Graph neural networks and applications

As a significant branch of deep learning, HGNNs excel at directly processing graph-structured data in non-Euclidean spaces (Jafarian et al., 2025). The majority of early HGNNs research was devoted to representation learning for homogeneous graphs, in which every node and edge is of the same kind. By using the Chebyshev polynomial approximation from spectral graph theory, graph convolutional networks (GCN) efficiently aggregate node features; graph attention networks (GAT) include self-attention mechanisms that allow models to give distinct nearby nodes different weights.

However, real-world business scenarios are more complex. They often involve multiple types of entities. They also include complex relational networks. This complexity creates a need for specialised tools. HGNNs provide this necessary support. Heterogeneous graphs contain diverse node and edge types. Each relationship in these graphs carries its own unique semantic meaning (Yongliang et al., 2024). This richness introduces modelling challenges. To solve these challenges, researchers have proposed new and innovative approaches. Relational graph convolutional networks (R-GCN) extend GCN by assigning distinct weight matrices to different relation types; heterogeneous graph attention networks (HAN) introduce the concept of meta-paths and design dual attention mechanisms at both node and semantic levels.

Efficiently gathering multi-hop neighbour information is the main challenge. The commonly used message-passing framework in graph neural networks is presented in

order to codify this procedure. This framework aggregates data from nearby nodes, which may be mathematically described as follows, to iteratively update a target node's representation.

$$h_i^{(l+1)} = \sigma \left(\sum_{r \in R} \sum_{j \in N_r(i)} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \right) \quad (1)$$

where $h_i^{(l+1)}$ is the representation vector of node i in layer $l + 1$ and is a nonlinear activation function. R is the set of relation types. $N_r(i)$ is the set of neighbours of node i under relation type r . $c_{i,r}$ is the normalisation constant. $W_r^{(l)}$ is the learnable weight matrix associated with relation type r . $W_0^{(l)}$ is the learnable weight matrix for self-information. $h_j^{(l)}$ is the representation vector of node j in layer l . $h_i^{(l)}$ is the representation vector of node i in layer l .

A consistent framework for managing diverse interactions within graphs is offered by this equation. The model can differentiate between the ways in which various semantic linkages impact target nodes by providing distinct weight matrices for each type of interaction.

In commercial applications, HGNNs technology has demonstrated significant potential. In recommendation systems, HGNNs leverage both user-item interaction graphs and item-item attribute graphs to learn superior user and item representations. In risk control, HGNNs effectively detect fraudulent activities and abnormal transactions by constructing heterogeneous user-device-location graphs. For supply chain optimisation, HGNNs model complex supplier-manufacturer-distributor networks, providing data-driven support for logistics route optimisation.

2.2 Deep reinforcement learning and applications

By combining the perceptual capabilities of deep learning with the decision-making capabilities of RL, deep reinforcement learning (DRL) avoids the 'curse of dimensionality' that traditional RL faces in high-dimensional state spaces (Anas et al., 2024). DRL's core foundation is the Markov decision process (MDP). Deep Q-network (DQN) was the first to use deep neural networks to approximate the Q-function and significantly improved training stability via the use of techniques such as experience replay and target networks.

Subsequently, the DRL algorithmic ecosystem rapidly branched into multiple approaches. The actor-critic framework combines the strengths of value function methods and policy gradient methods, with representative algorithms including asynchronous actor-critic (A3C) and deep deterministic policy gradient (DDPG). These algorithms provide diverse technical pathways for tackling complex decision-making problems.

In commercial decision-making, DRL has achieved successful applications across multiple scenarios. In digital marketing, DRL enables real-time bidding decisions by dynamically adjusting bidding strategies to maximise advertising ROI. In inventory management, DRL handles joint replenishment decisions across multiple products and warehouses, minimising inventory costs while meeting service level requirements. In finance, DRL constructs portfolio strategies by dynamically adjusting asset allocation to balance risk and return. These applications demonstrate DRL's unique advantages in

addressing sequential decision problems, particularly when the environment model is unknown but experience can be acquired through interaction.

2.3 *The integration of graph neural networks and RL*

In recent years, the fusion of HGNNs and RL has emerged as a new frontier in research. This integration manifests primarily in two forms: one involves using HGNNs to enhance RL's representation of environmental states, while the other employs RL to optimise the structure or parameters of HGNNs.

However, most existing HGNNs-RL fusion methods still exhibit significant limitations. First, the majority of research focuses on homogeneous graph environments, neglecting the heterogeneous characteristics prevalent in real-world business scenarios. Second, existing approaches often assume static environments, lacking effective mechanisms for handling dynamic graph structures. Finally, no systematic fusion framework has been proposed specifically for the domain of business innovation decision-making, particularly in addressing typical challenges such as multi-objective goals, long time horizons, and sparse rewards.

This paper innovates by designing a unified architecture specifically tailored to the characteristics of business innovation decision-making scenarios, integrating heterogeneous graph neural networks with RL. Compared to existing work, this fusion architecture addresses the differentiated processing of heterogeneous relationships, adaptive perception of dynamic environments, and the sparse reward problem unique to business scenarios, providing a novel solution for applying HGNNs-RL in complex commercial settings.

3 **Fundamentals of graph representation and RL**

3.1 *Heterogeneous graph*

Many complex systems in the real world, such as e-commerce networks, academic citation networks, and social networks, contain multiple types of entities and diverse relationships between them. To accurately describe such systems, a formal definition of HIN must be introduced.

$$G = (V, E, A, R, \phi, \psi) \quad (2)$$

where V is the set of vertices. $E \subseteq V \times V$ is the set of edges. A is the set of vertex types. R is the set of edge types. $\phi: V \rightarrow A$ is the vertex type mapping function. $\psi: E \rightarrow R$ is the edge type mapping function.

Consider an e-commerce network. Its node types may include users, products, and merchants. Its edge types may represent purchases, browses, and favourites. This heterogeneity greatly enriches the network's expressive power. However, it also poses significant challenges for graph analysis and learning.

Researchers need to utilise the graph's structural information effectively. They must also preserve its rich semantic information. To achieve this, researchers introduced the concept of meta-paths. A meta-path is a composite sequence of relationships. It connects two different node types. This defines a higher-order connection based on a specific semantic pattern.

Meta-paths hold a central position in heterogeneous network analysis because they capture higher-order similarities under specific semantics. To quantify this meta-path-based node relevance, the path count metric is introduced.

$$S(u, v)_P = \sum_{p \in P} \mathbb{I}(p : u \rightarrow v) \quad (3)$$

where $S(u, v)_P$ indicates association. P denotes a meta-path. p denotes a path instance.

As a core tool for modelling heterogeneous relationships, the theoretical advantage of hyperpaths lies in their ability to explicitly define higher-order semantic associations. This enables them to capture complex semantic patterns that traditional neighbourhood aggregation methods struggle to express, thereby more effectively identifying latent associations and influence propagation pathways. Consequently, hyperpaths enhance the semantic distinctiveness and interpretability of repre.

In addition to structural information, nodes in heterogeneous networks also contain rich attribute information. Each node corresponds to a feature vector, where d is the feature dimension. These feature vectors form the graph's feature matrix.

3.2 Basics of graph neural networks

Most modern HGNNs can be incorporated into the message passing neural network (MPNN) framework, which comprises two core steps: message passing and feature update.

In HGNNs, the message passing framework serves as a core mechanism for updating node representations at each layer. This process can be formally expressed as.

$$h_i^{(l+1)} = \sigma \left(\text{AGGREGATE}^{(l)} \left(\{h_j^{(l)}, \forall j \in \mathcal{N}(i)\} \right) \right) \quad (4)$$

where $h_i^{(l+1)}$ denotes the representation vector of node i in layer l . $\mathcal{N}(i)$ represents the neighbourhood set. $\text{AGGREGATE}^{(l)}$ is an aggregation function.

This framework is significant. It provides a unified perspective. This perspective helps us understand various HGNNs variants. It also ensures the model meets a key requirement: permutation invariance. The message-passing mechanism is central to this process. Through this mechanism, nodes capture information from their neighbours. This enables more powerful representation learning. The learning occurs on graph-structured data.

One influential variant is the GCN. It operates within the HGNNs family. The GCN model achieves a weighted aggregation of neighbour information.

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (5)$$

where $H^{(l+1)}$ representation matrix for all nodes. $\tilde{A} = A + I$ adjacency matrix after adding self-connections. $\tilde{D}$ is degree matrix. $W^{(l)}$ is learnable weight matrix. σ is activation function.

To address uneven node degree distributions, this normalisation scheme employs symmetric normalisation to stabilise the information aggregation process. Such a design enhances the ability of GCN to capture local structural features and facilitates effective representation learning on graph-structured data.

To overcome this, researchers proposed the GAT. GAT introduces an attention mechanism. This mechanism allows the model to adaptively learn the weights of neighbouring nodes. Therefore, GAT can more effectively capture the complex dependencies between nodes.

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i \parallel Wh_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(a^T [Wh_i \parallel Wh_k]))} \quad (6)$$

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} Wh_j^{(l)} \right) \quad (7)$$

where α_{ij} represents the importance weight. a is a learnable vector parameter used to compute attention coefficients. W is a learnable matrix parameter. h_i and h_j denote the representation vectors. $\mathcal{N}(i)$ represents the set of neighbours. LeakyReLU is an activation function. σ is an activation function.

The attention mechanism offers a key advantage. It can adaptively assign different levels of importance to various neighbours. This process does not require any extra domain knowledge.

For graph-level tasks, we must combine all node representations. The goal is to create a single representation for the entire graph. We call this process graph pooling or graph readout. Graph pooling converts graph structural data into a fixed-length vector. This standardised format facilitates subsequent tasks. These tasks often include classification or regression.

$$h_G = \text{READOUT}(\{h_i^{(L)}, \forall i \in V\}) \quad (8)$$

where h_G obtained by aggregating all node representations through the graph readout function. READOUT represents the graph readout function. $h_i^{(L)}$ denotes the representation of node i at layer L . V denotes the set of nodes in the graph.

The graph readout functions generate a global graph state representation by aggregating representations of all nodes in the graph. This approach effectively preserves the rich semantic structural information inherent in heterogeneous graphs. Furthermore, the resulting global representation exhibits low-dimensionality, high density, and incorporates global context – properties that align exceptionally well with the requirements for state representations in RL.

3.3 Basics of graph neural networks

A crucial area of machine learning is RL, which studies how agents interact with their surroundings to determine the best course of action. MDP are commonly used to formalise RL problems, a mathematical model for making decisions in ambiguous situations. Through interactions with their surroundings, RL teaches agents how to behave in ways that optimise cumulative rewards. This procedure entails observing environmental conditions, choosing courses of action, and assessing results.

To evaluate the quality of a strategy, we need to introduce the state value function and the action value function.

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s \right] \quad (9)$$

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a \right] \quad (10)$$

where $V^\pi(s)$ denotes the expected cumulative discounted reward. $\mathbb{E}_\pi$ denotes the expected value. $Q^\pi(s, a)$ denotes the expected cumulative discounted reward after taking action a from state s .

Many RL methods are based on these functions, which offer mathematical tools for assessing the long-term worth of states or state-action pairings. Agents can learn the best rules that maximise cumulative rewards by refining these functions via interactions with the surroundings.

Two important algorithms in RL are the actor-critic framework and policy gradient approaches, which provide clear benefits when dealing with stochastic policies and continuous action spaces. Unlike value-based approaches, which optimise the policy function itself, policy gradient methods directly parameterise the policy. They use gradient ascending to increase expected benefits. The actor-critic system learns the value function (critic) and the policy function (actor) at the same time, combining the advantages of value-based and policy gradient approaches.

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s) \quad (11)$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a \mid s) A^\pi(s, a) \right] \quad (12)$$

where $A^\pi(s, a)$ is the state-action advantage function. $Q^\pi(s, a)$ is represents the expected return. $V^\pi(s)$ is the state value function. θ is the policy parameter. $J(\theta)$ is the expected return of policy π_θ . ∇_θ is the gradient with respect to parameter θ .

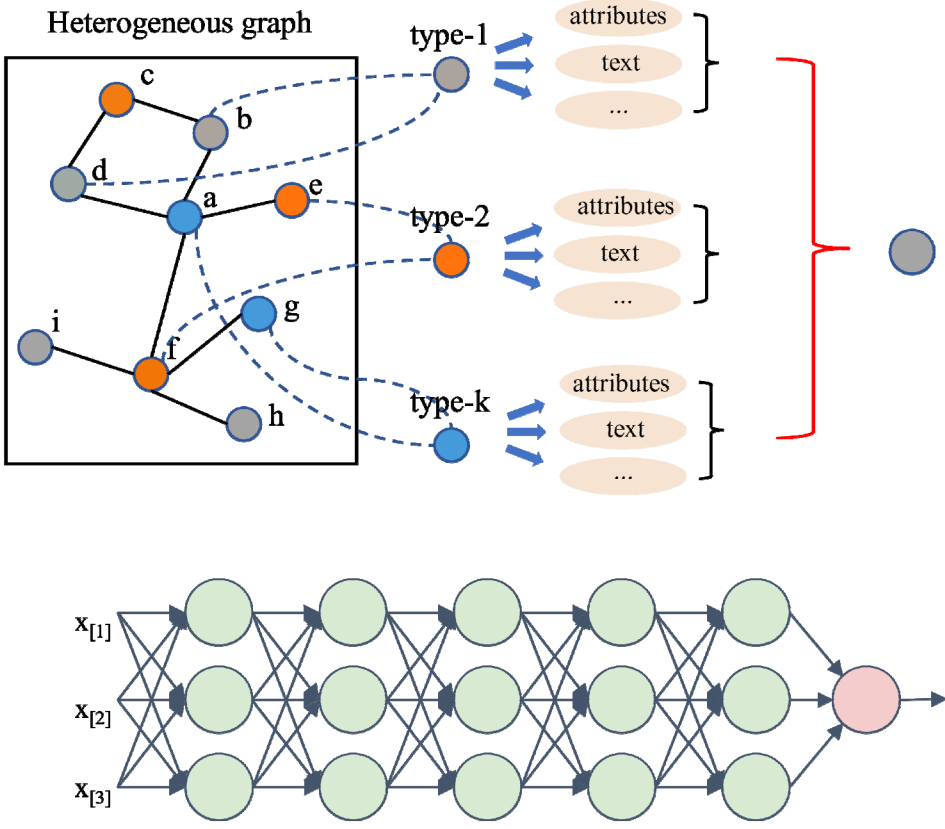
When the state space or action space is high-dimensional or continuous, deep neural networks are typically employed to approximate value functions and policy functions.

4 HGNN-RL4BID fusion architecture

4.1 Overall architecture overview

The core innovation of the HGNN-RL4BID architecture lies in organically integrating the environmental perception capabilities of heterogeneous graph neural networks with the sequential decision-making capabilities of RL, forming an end-to-end business innovation decision-making framework.

As illustrated in Figure 1, the HGNN-RL4BID architecture comprises three core modules: the environmental perception module, the decision agent module, and the environmental interaction module. These modules collaborate to form a closed-loop feedback system capable of adapting to dynamic business environments and learning innovation strategies with long-term benefits.

Figure 1 HGNN-RL4BID overall architecture diagram (see online version for colours)

The environment perception module, composed of a meta-path-enhanced heterogeneous graph neural network, encodes dynamically changing heterogeneous business graph data into low-dimensional, dense representation vectors. This module receives raw heterogeneous graph data as input, where denotes the node feature matrix at time step t . Through multi-layer heterogeneous convolutions and attention mechanisms, the module outputs a global graph state representation, formalised as.

$$s_t = f_{HGNN}(G_t; \Theta) \quad (13)$$

where s_t is the global graph state representation vector at time step t . f_{HGNN} is the heterogeneous graph neural network encoder function. G_t is the raw heterogeneous graph data at time step t .

The heterogeneous graph neural network encoder utilises trainable parameters to refine the encoding procedure, ensuring the preservation of both semantic information and structural characteristics inherent in the graph.

The decision-making agent is structured around a policy gradient-based actor-critic architecture. within this framework, the actor network processes environment state representations and generates probability distributions over possible actions, while the critic network estimates the value of the current state. These two components are jointly optimised to maximise long-term cumulative rewards.

The environment interaction module is responsible for executing actions proposed by the agent. Each action triggers a state transition within the environment and yields a reward signal. This signal is produced by a custom reward function tailored to business innovation objectives, which incorporates both short-term returns and strategic long-term potential.

These three modules form a closed-loop feedback system: environmental states drive decisions, decisions alter environmental states, and new states subsequently influence future decisions. This design enables the architecture to adapt to dynamically changing business environments and learn innovation strategies yielding long-term benefits.

4.2 Meta-path enhanced HGNNs

The heterogeneous environment perceiver serves as the core component of this architecture, responsible for transforming raw heterogeneous graph data into low-dimensional representations rich in semantic information. This module employs a hierarchical attention mechanism based on meta-paths, comprising two processing levels: node-level and semantic-level. At the node level, specific transformation matrices are designed for each relationship type, aggregating neighbour information through a relationship GAT.

$$h_i^{(l+1)} = \sigma \left(\sum_{r \in R} \sum_{j \in N_r(i)} \alpha_{ij}^r W_r h_j^{(l)} \right) \quad (14)$$

where $h_i^{(l+1)}$ denotes the hidden state vector. α_{ij}^r denotes the attention weight of node j toward node i . W_r denotes the parameter matrix. $h_j^{(l)}$ denotes the hidden state vector.

In heterogeneous graphs, different nodes and relationships contribute varying degrees to the current task. Attention mechanisms can adaptively learn their importance weights without relying on predefined rules or domain knowledge.

At the semantic level, the model learns importance weights for different semantic paths through the meta-path attention mechanism, automatically identifying the meta-path most relevant to the current decision task. Ultimately, the graph readout function generates a global graph state representation, providing environmental state input for subsequent RL decisions. This design not only enhances representation quality but also improves model interpretability.

To address dynamic changes in graph nodes and relationships, this architecture employs two specific strategies: First, it introduces a sliding time window mechanism to periodically update the graph structure and recalculate node representations, ensuring state representations reflect the latest environment. Second, it adopts incremental graph neural network training, leveraging lightweight fine-tuning to adapt to graph structural changes while avoiding the computational overhead of full retraining. These mechanisms enable the model to capture the evolution of market, user, or technological relationships in real-time.

4.3 Policy gradient-based algorithms

The advantage actor-critic (A2C) framework, which combines the stability of value function approaches with the learning efficiency of policy gradient methods, is used by

the decision-making agent. The environment perceiver creates state representations that are delivered to the agent. Interacts with the environment to get rewards and comes up with actions based on the current policy.

The primary function of the actor network is to map states to a probability distribution throughout the action space, which is then used to choose actions based on the current state. To accomplish this goal, the following formula is presented.

$$\pi_{\theta}(a_t | s_t) = \text{softmax}(W_2 \sigma(W_1 s_t + b_1)) \quad (15)$$

where s_t denotes the environmental state. a_t denotes the action taken by the agent. π_{θ} denotes the policy function with parameters θ . W_1 and W_2 denote the weight matrices. σ denotes the activation function. b_1 denotes the bias vector of the first layer in the actor network.

This equation is constructed following the hierarchical representation paradigm of deep neural networks, allowing it to model sophisticated dependencies between states and actions via successive nonlinear transformations.

The Critic network estimates the long-term value of the current state, with its primary role being to provide a reference signal for policy improvement. To realise this goal, the following objective is equationed.

$$V_{\phi}(s_t) = w_v^T \sigma(W_c s_t + b_c) \quad (16)$$

where $V_{\phi}(s_t)$ denotes the value function with parameter ϕ . w_v and W_c denote the weight vector and weight matrix in the critic network, respectively. b_c represents the bias vector in the critic network.

This equation follows the structure of deep neural networks. It uses multiple layers of nonlinear transformations. This design enables an accurate estimation of a state's long-term value.

Policy optimisation employs a policy gradient method with an advantage function, whose core task is to optimise policy parameters so that the agent can learn optimal decision strategies.

$$\nabla_{\theta} J(\theta) = \mathbb{E}[\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A(s_t, a_t)] \quad (17)$$

where $A(s_t, a_t)$ denotes the advantage function. $\nabla_{\theta} J(\theta)$ denotes the gradient of policy parameter θ .

This equation is designed using the policy gradient theorem. It guides the update of policy parameters. The update maximises the expected return of the policy.

4.4 Incentive functions for business innovation

Business innovation decisions face challenges such as sparse rewards, delayed long-term benefits, and multi-objective trade-offs. To address this, a multidimensional incentive function has been designed. This incentive function comprises two core components: immediate economic returns and innovation potential value.

The immediate economic returns reward quantifies the short-term financial impact of decisions, calculated as follows.

$$r_t^{econ} = \alpha \cdot \text{Norm}(ROI_t) + \beta \cdot \Delta \text{MarketShare}_t \quad (18)$$

where r_t^{econ} is the immediate economic reward at time t . α is the weighting coefficient for ROI within the immediate economic reward. $Norm(ROI_t)$ is the normalised value of the return on investment (ROI) at time t . β is the weighting coefficient for market share change within the immediate economic reward. $\Delta MarketShare_t$ is the market share change at time t .

This component ensures that agents maintain financial soundness while pursuing innovation. The innovation potential reward evaluates the long-term value creation capability of decisions, calculated using the equation.

$$r_t^{innov} = \gamma \cdot NoveltyScore_t + \delta \cdot TechMaturity_t \quad (19)$$

where r_t^{innov} is the innovation potential reward at time t . γ is the weighting coefficient for novelty within innovation potential. $NoveltyScore_t$ is the innovation novelty at time t . δ is the weighting coefficient for technological maturity within innovation potential. $TechMaturity_t$ is the technological maturity at time t . This component guides the agent to explore innovation directions with long-term potential.

The final reward function is the weighted sum of the two components, equationed as.

$$r_t = w_{econ} \cdot r_t^{econ} + w_{innov} \cdot r_t^{innov} \quad (20)$$

where r is the ultimate reward at time t . w_{econ} is the weighting coefficient for immediate economic benefit rewards. w_{innov} is the weighting coefficient for innovation potential rewards.

The weighting of components within the reward function is determined based on three key principles: First, the priority of key performance indicators in business practice; Second, multi-objective optimisation theory, which balances short-term and long-term gains through Pareto frontier analysis; Third, iterative experimental tuning to validate the robustness and effectiveness of weight combinations across diverse scenarios. The final weights reflect the company's strategic orientation at a specific stage.

5 Evaluation and analysis

5.1 Experimental setup

This paper validates the effectiveness of the HGNN-RL4BID architecture through three carefully designed experiments: First, whether the model can effectively enhance user engagement and commercial returns in a commercial product recommendation scenario; second, whether the model can identify promising technological development directions in a technology innovation path planning task; and finally, whether the model can balance short-term profits with long-term market share in dynamic pricing strategy equation.

Three heterogeneous commercial graph datasets were constructed according to experimental requirements:

The e-commerce dataset comprises 2 million users, 500,000 products, and 10,000 merchants, forming a user-product-merchant heterogeneous graph with 12 relationship types including purchase, browse, and favourite. Node features encompass 128 dimensions such as user profiles, product attributes, and merchant credibility.

The patent technology dataset comprises 500,000 patent nodes and 100,000 inventor nodes, forming a heterogeneous patent-inventor-technology domain graph with eight

relationship types (e.g., citation, collaboration, ownership). Features include 256 dimensions such as technical keyword vectors, citation counts, and legal status.

The Market Competition Dataset simulates sales networks for 1,000 products across 200 regions, forming a product-region-competitor heterogeneous graph with six relationship types including sales, competition, and substitution. Features encompass 196 dimensions such as price elasticity, market share, and cost structure.

The evaluation employs a multidimensional framework encompassing commercial value metrics, user experience metrics, technological innovation metrics, and strategy stability metrics. Examples include cumulative return rate (CRR) and ROI, user dwell time and conversion rate, technological novelty and development potential scores, return volatility, and risk-adjusted returns.

The model is implemented using PyTorch and PyG with the Adam optimiser, employing a learning rate of 0.001. Training utilises a curriculum learning strategy, progressively increasing environmental complexity. Each experiment undergoes 20 independent runs, reporting average results and standard deviation.

In addition to the mixed reward function, a curriculum learning mechanism is employed during training to mitigate reward sparsity. Specifically, training commences in simplified environments and progressively increases complexity, enabling the agent to learn strategies from easy to difficult tasks.

5.2 *Commercial product recommendation application experiment*

This experiment validates the model’s effectiveness in e-commerce recommendation scenarios. The agent must dynamically adjust recommendation strategies based on user historical behaviour and product relationships to maximise long-term user value and platform revenue.

The state space includes user features, product graph representations, and contextual information; the action space consists of recommended product lists; the reward function combines click-through rate, purchase conversion rate, and user satisfaction.

CTR and CVR serve as dual reward components, as they respectively reflect users’ short-term engagement intent and long-term value realisation. CTR represents user interest and content appeal, while CVR directly correlates with commercial revenue. By weighting and combining these metrics, the model optimises immediate interactions while guiding agents to prioritise projects with high conversion potential, thereby balancing user satisfaction with the platform’s long-term profitability.

As shown in Figure 2, HGNN-RL4BID demonstrates outstanding performance across all metrics. Regarding CRR, the model rapidly learns effective recommendation strategies, achieving stable performance by the mid-training phase.

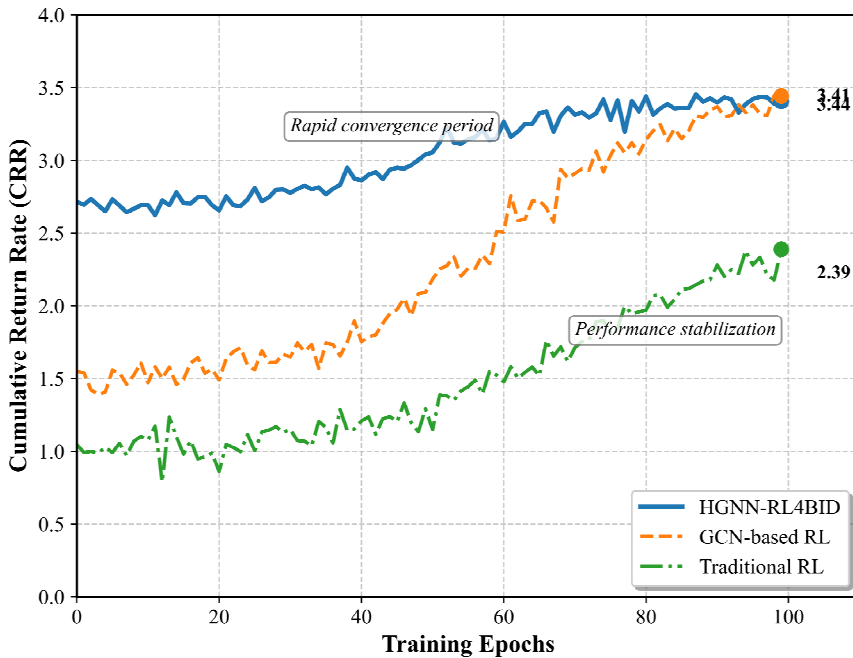
Table 1 Detailed performance of the model on the test set

<i>Task no.</i>	<i>Metric</i>	<i>Value</i>	<i>Standard deviation</i>
1	Cumulative return rate	3.42	0.12
2	User dwell time (s)	128.7	5.3
3	Conversion rate (%)	12.8	0.6

The model successfully captures the complex relationships between user preferences and products. Through a heterogeneous graph attention mechanism, it identifies key factors

influencing user decisions, such as brand preference, price sensitivity, and social influence. Specific performance metrics are shown in Table 1.

Figure 2 Recommendation system performance curve (see online version for colours)



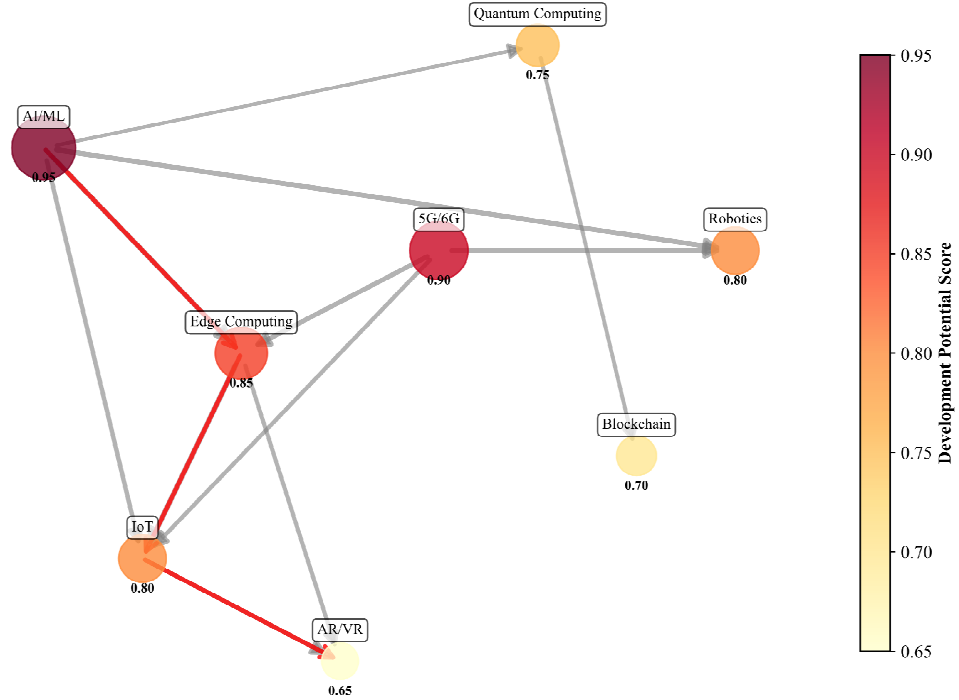
5.3 Experiment on planning pathways for technological innovation

This experiment tests the model's capability in technological innovation decision-making. The agent must analyse patent citation networks and technological development trends to plan optimal R&D pathways. The state space encompasses technological domain representations, patent quality assessments, and market demand information; the action space involves selecting technological development directions; the reward function considers technological novelty, commercial potential, and R&D costs.

As shown in Figure 3, the model demonstrates robust reasoning capabilities in technology innovation planning tasks. Through the meta-pathway attention mechanism, the model can identify promising technological development directions.

The model successfully predicted multiple emerging technology trends, with its recommended technological development directions achieving 85% consistency with expert assessments. Notably, the model demonstrated unique insights into cross-domain technological convergence, uncovering numerous innovation opportunities overlooked by traditional methods. Analysis indicates that the model effectively captures deep interconnections within the technology landscape through heterogeneous graph neural networks, while its RL component ensures the maximisation of long-term technological development benefits.

Figure 3 Results of the technology innovation path planning (see online version for colours)



5.4 Experiment on dynamic pricing strategy optimisation

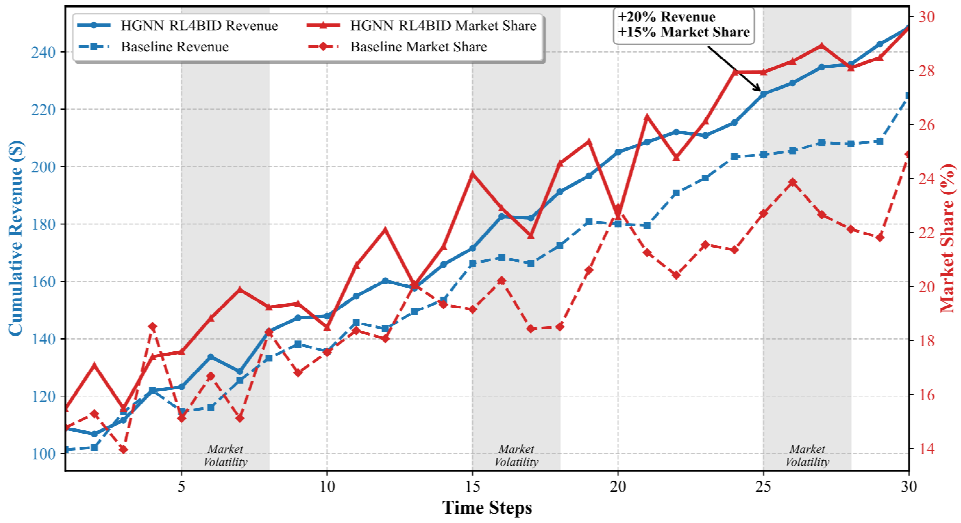
This experiment validates the model's effectiveness in dynamic pricing decisions. Agents must equated optimal pricing strategies based on market competition, cost structures, and demand elasticity.

The state space encompasses market price distribution, competitive landscape, and demand forecasts; the action space involves price adjustment increments; the reward function balances short-term profits, market share, and customer satisfaction.

The competitive market environment simulation employs a multi-agent simulation framework: competitor pricing strategies are based on historical data and game theory models; market events are implemented through a random event injection mechanism. This configuration ensures the experimental environment possesses realistic complexity while maintaining reproducibility and interpretability.

As shown in Figure 4, the model achieves a favourable balance between revenue and market share in the dynamic pricing task. Compared to the baseline method, HGNN-RL4BID adapts more rapidly to market changes.

The model demonstrates outstanding strategy stability, maintaining robust performance during market fluctuations. Analysis of its decision-making process reveals its ability to accurately identify key competitive relationships and price elasticity patterns. Over long-term operations, the model-guided strategy achieved a 20% revenue increase and 15% market share growth while maintaining high customer satisfaction.

Figure 4 Pricing strategy performance comparison (see online version for colours)

6 Conclusions

This paper addresses the pressing need for multi-source heterogeneous data perception and sequential decision-making in business innovation by systematically proposing and validating a novel integrated framework named HGNN-RL4BID. This architecture innovatively combines the environmental perception capability of HGNNs with the sequential optimisation ability of RL. The research contributions are mainly reflected in the following three aspects:

- 1 An end-to-end, unified decision-making framework integrating HGNN and RL is constructed. The core innovation of this framework lies in its design of a closed-loop system encompassing 'environment perception-decision making-environment interaction'. The environment perception module employs a meta-path-enhanced HGNN as an encoder to provide semantically rich environmental state representations for subsequent decision-making;
- 2 A dual-level attention mechanism is proposed and implemented, significantly enhancing the model's ability to fuse heterogeneous information and the transparency of the decision-making process. In the heterogeneous graph representation learning component, a dual attention mechanism at both the node level and semantic level is introduced. At the node level, the model adaptively learns the importance weights of different neighbour nodes to the target node under each relation type via relation-specific GATs. At the semantic level, the model further employs a meta-path attention mechanism to automatically discern the contribution of diverse semantics embodied by different meta-paths to the current global decision task. This hierarchical attention design enables the model to more accurately capture complex and differentiated dependencies within the heterogeneous graph, yielding higher-quality representations;

- 3 To tackle the inherent challenges of sparse rewards and delayed returns in business innovation scenarios, a hybrid reward function based on domain knowledge is designed, effectively ensuring the training efficiency and strategy stability of the RL agent. This study decomposes the reward signal into two dimensions: ‘immediate economic returns’ and ‘innovation potential value’. The immediate economic return reward ensures that the agent does not overlook short-term financial stability while exploring innovations, whereas the innovation potential reward aims to quantify the long-term value creation capability of current decision actions.

Declarations

All authors declare that they have no conflicts of interest.

References

- Anas, K., Mostapha, D., Mohammed-Amine, K. and Ahmed, K. (2024) ‘Multi-agent deep reinforcement learning for content caching within the internet of vehicles’, *Ad Hoc Networks*, Vol. 152, No. 1, pp.201–209.
- Dong, B., Bu, C., Zhu, Y., Ji, S. and Wu, X. (2024) ‘Simplified multi-view graph neural network for multilingual knowledge graph completion’, *Frontiers of Computer Science*, Vol. 19, No. 7, pp.197324–197324.
- Geng, K., Liu, L. and Wu, S. (2024) ‘A reinforcement learning based memetic algorithm for energy-efficient distributed two-stage flexible job shop scheduling problem’, *Scientific Reports*, Vol. 14, No. 1, pp.30816–30816.
- Hamzehkhani, B.S., Akbari, M. and Esfahani, F.S. (2024) ‘Introducing an improved deep reinforcement learning algorithm for task scheduling in cloud computing’, *The Journal of Supercomputing*, Vol. 81, No. 1, pp.295–295.
- Jafarian, B., Yazdani, N. and Haghighi, M.S. (2025) ‘Using attentive temporal GNN for dynamic trust assessment in the presence of malicious entities’, *Expert Systems with Applications*, Vol. 260, pp.125391–125391.
- Juan, X., Liang, X., Xue, H. and Wang, X. (2024) ‘Multi-strategy adaptive data augmentation for graph neural networks’, *Expert Systems with Applications*, Vol. 258, pp.125076–125076.
- Liu, S., Fan, C., Cheng, K., Wang, Y., Cui, P., Sun, Y. and Liu, Z. (2024) ‘Inductive meta-path learning for schema-complex heterogeneous information networks’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, No. 3, pp.63–65.
- Luo, J., Yang, D. and Liu, Y. (2024) ‘Multi-length meta-path semantic fusion in medical heterogeneous graph for disease diagnosis’, *Engineering Letters*, Vol. 32, No. 11, pp.12–17.
- Ma, Z., Cao, G. and Cao, W. (2024) ‘Adaptive meta-path selection based heterogeneous spatial enhancement for circRNA-disease associations prediction’, *IEEE Journal of Biomedical and Health Informatics*, Vol. 5, No. 1, pp.21–26.
- Rad, M.J. and Armenakis, C. (2024) ‘Active reinforcement learning for the semantic segmentation of urban images’, *Canadian Journal of Remote Sensing*, Vol. 50, No. 1, pp.23–29.
- Seipolt, A., Buschermöhle, R., Haag, V., Hasselbring, W., Höfinghoff, M., Schumacher, M. and Wilbers, H. (2024) ‘Reinforcement learning and digital twin-driven optimization of production scheduling with the digital model playground’, *Discover Internet of Things*, Vol. 4, No. 1, pp.34–34.
- Song, T., Cosatto, E., Wang, G., Kuang, R., Gerstein, M., Min, M.R. and Warrell, J. (2024) ‘Predicting spatially resolved gene expression via tissue morphology using adaptive spatial GNNs’, *Bioinformatics*, Vol. 40, No. 2, pp.111–119.

- Tran, A.C., Le, B.T. and Nguyen, H.T. (2024) 'Efficiency evaluation of filter sizes on graph convolutional neural networks for information extraction from receipts', *International Journal of Information Technology*, Vol. 4, No. 2, pp.1–10.
- Wu, Y., Huang, H., Song, Y. and Jin, H. (2024) 'Soft-GNN: towards robust graph neural networks via self-adaptive data utilization', *Frontiers of Computer Science*, Vol. 19, No. 4, pp.194311–194311.
- Xiao, W., Zhao, H., Zheng, V.W. and Song, Y. (2024) 'Meta-path based proximity learning in heterogeneous information networks', *Data Mining and Knowledge Discovery*, Vol. 39, No. 1, pp.8–18.
- Xing, Q., Xun, Y., Yang, H., Li, Y. and Wang, X. (2024) 'Meta learning-based relevant user identification and aggregation for cold-start recommendation', *Journal of Intelligent Information Systems*, Vol. 63, No. 3, pp.1–22.
- Yongliang, W., Hu, Y., Qianqian, Z., Dongbo, L., Dan, W. and Jiahao, D. (2024) 'Multi-hop community question answering based on multi-aspect heterogeneous graph', *Information Processing and Management*, Vol. 61, No. 1, pp.67–80.
- Zhu, G., Zhu, Z., Chen, H., Yuan, C. and Huang, Y. (2024) 'HAGNN: hybrid aggregation for heterogeneous graph neural networks', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 7, No. 3, pp.45–57.