Intelligent Question-Answering and Knowledge Transfer in Computer Programming Courses Based on ChatGPT and Knowledge Distillation

Haiyan Liu[1], Yanna Wei[1], Penghua Zhu[2]
[1]North China Institute of Aerospace Engineering, Langfang 065000, Hebei, China
[2]Cangzhou Normal University, Cangzhou 061000, Hebei, China

Corresponding author: Penghua Zhu (e-mail: mnzghi@163.com)

ABSTRACT Current intelligent question-answering and knowledge transfer in programming courses suffer from low efficiency, insufficient model lightweighting, and semantic consistency deviations, limiting the practical deployment of large language models in specialized engineering education. As semantic reasoning and hierarchical information representation become increasingly important for intelligent information processing and knowledge interaction in advanced engineering systems, this paper proposes a ChatGPT semantic transfer optimization method based on hierarchical knowledge distillation. By constructing a multi-level semantic knowledge graph covering programming concepts, syntax, and task logic, the proposed framework performs hierarchical semantic modeling and introduces a multi-layer semantic alignment loss to achieve precise knowledge transfer. Parameter pruning constraints and a dynamic temperature adjustment strategy are further incorporated to improve model lightweighting and training stability, while a semantic consistency judgment mechanism supports intelligent question-answering in programming teaching scenarios. Experimental results demonstrate significant reductions in concept-, syntax-, and task-level distillation losses, together with substantial parameter compression and inference acceleration while maintaining high semantic consistency and answer accuracy. The convergence of semantic alignment corresponds to measurable improvements in practical question-answering performance. The proposed approach provides an effective technical pathway for intelligent programming education and offers methodological references for hierarchical knowledge transfer and semantic information processing in advanced engineering and electromagnetic information systems.

INDEX TERMS ChatGPT, hierarchical knowledge distillation, intelligent question-answering, semantic information processing, knowledge transfer

I. INTRODUCTION

As programming engineering training deepens, computer programming classes have gradually become essential for mastering the method design, process regulation, and intelligent manufacturing techniques required to develop new functional materials and production lines in the industry. But, programming training is often limited by few teachers, huge student foundation variations and fast knowledge updates. This leads to ineffective classroom questions-answering and inadequate knowledge transferring, which hinders college students' deep understanding of programming logic and method algorithms [1,2]. It is vital to develop a teaching method with intelligent questions-answering and knowledge transferring capabilities to enhance computer programming instruction, fulfill individuals' studying wants and foster digital training in programming engineering [3,4]. Semantic transfer and automated questions-answering mechanisms can lighten teachers' burden and support efficient course knowledge dissemination and dynamic knowledge structures updating [5,6]. In this study, semantic transfer denotes the process in which the hierarchical semantic space of the student model is reshaped to follow the ordered semantic structure of the teacher model. The term multi-layer semantic alignment refers to the constraint that preserves the correspondence among concept-level, syntax-level and task-level representations when the student model receives the semantic signals of the teacher model.

The current intelligent question-answering application in programming teaching still have many problems to be solved. As can be seen, programming knowledge has the characteristics of complex semantic level, tight hierarchy relationship and deep logic dependence. Traditional question-answering methods based on KEYWORDS matching or similar retrieval cannot well understand the programming semantics and logic processing [7,8]. Although the large language model has the ability of strong semantic generation, it still has problems of semantic diffusion, low distillation accuracy and large model size in the transfer of professional courses, which restrict the practical application of teaching [9,10]. In addition, the semantic consistency of teaching scene and the stability of reasoning requirement are high, but the existing model still has problems of poor understanding of long texts and poor preservation of semantics in multiple question-answering [11,12]. Recently, the above problems have been alleviated from the multi-dimensional angle. Some work has finetuned course materials based on the BERT (Bidirectional Encoder Representation from Transformers) model to improve the accuracy of programming semantic understanding and question-answer generation [13,14]. Another type of research uses graph neural networks to construct programming knowledge graphs, realizing semantic associations between the concept layer, syntax layer, and task layer [15,16]. Other studies have adopted knowledge distillation strategies to transfer knowledge from ChatGPT or other large teacher models to lightweight student models, thereby improving reasoning efficiency while maintaining semantic features [17,18]. However, these methods still have problems in programming teaching, such as unclear semantic hierarchy division, single distillation objectives, insufficient semantic alignment during the transfer process, and lack of consistency constraints, making it difficult to form a high-quality knowledge transfer system for teaching scenarios [19,20].

Research on lightweight question-answering and knowledge transfer models for programming-oriented learning has explored compressed transformer structures, pruned attention layers and compact reasoning modules to reduce computational scale, yet these approaches maintain only partial semantic stability when handling hierarchical programming information. Distilled question-answering models reported in existing studies retain coarse semantic patterns but lack constraints on multi-level embedding alignment, which creates gaps between concept semantics, syntax relations and high-level task representations. Long-text reasoning and multi-step semantic continuity remain unstable under these lightweight structures, and the transfer process often limits optimization to shallow output distributions rather than deeper semantic spaces. These issues indicate that current lightweight designs have not formed a unified mechanism to preserve hierarchical semantic coherence during knowledge transfer, leaving their applicability in programming teaching scenes restricted.

ChatGPT is adopted as the teacher model because its semantic coverage, cross-domain reasoning fluency and unified representation ability form a more stable source of hierarchical semantics than models pre-trained solely on program corpora. Domain-specific models trained on code corpora usually contain fixed syntactic distributions and restricted abstraction depth, and their embeddings tend to remain close to local structural patterns of programming statements. These characteristics limit their ability to generate stable semantic transitions across concept, syntax and task levels. ChatGPT provides a broader semantic space in which programming logic, textual explanations and operational descriptions are expressed in a coherent representational form. The internal distribution of its embeddings maintains long-range semantic continuity, which strengthens the alignment of hierarchical semantics during distillation. This property improves the preservation of multi-level structural relations, especially in concept descriptions and task-level semantic chains that require integrated reasoning across procedural and descriptive information. Preliminary comparative experiments performed during the early stage of this research show that distillation from a code-oriented model produces a narrower embedding distribution and weaker stability in concept-level semantic alignment. The distillation loss fluctuates more significantly across epochs, and the transferred syntax–task chain exhibits inconsistent transitions in long-text reasoning. These observations confirm that ChatGPT provides a more complete semantic basis for hierarchical alignment. The selection of ChatGPT therefore originates from the need for a teacher model that sustains stable hierarchical mapping, uniform semantic density and high-level reasoning continuity throughout the distillation process. To address the key issues of low semantic transfer accuracy and insufficient model lightweighting, this study proposes a ChatGPT semantic transfer optimization method based on hierarchical knowledge distillation. The study first constructs a multilayer semantic knowledge graph covering the concept layer, syntax layer, and task layer of programming. Transformer encoding is used to generate contextual semantic representations, and a graph convolutional network is used to extract structured symbolic semantics to achieve multi-layer semantic fusion. Subsequently, a multi-layer semantic alignment loss function is introduced during the distillation process of the teacher and student models to ensure embedding consistency between different semantic layers. Parameter pruning constraints are then combined to achieve model structure sparsification. The pruning step is integrated into the alignment stage so that structural reduction follows the semantic signals transmitted from the teacher model. To improve the stability and transfer effect of the distillation process, a dynamic temperature regulation strategy is designed to control the changes in learning intensity during the distillation stage. Finally, an intelligent question-answering generation module is constructed in conjunction with a semantic consistency judgment mechanism. Through the joint optimization of semantic retrieval, answer generation, and consistency judgment, high-precision knowledge transfer and semantically consistent question-answering for programming teaching scenarios are achieved, providing technical support and methodological foundation for industry-specific intelligent teaching systems.

II. CONSTRUCTION OF AN INTELLIGENT QUESTION-ANSWERING SYSTEM FOR PROGRAMMING DRIVEN BY CHATGPT SEMANTIC DISTILLATION

The semantic transfer optimization method described in this work refers to a structured procedure that guides the student model to form an ordered semantic hierarchy inherited from the teacher model. This hierarchy is composed of three layers, and the formation of this ordered structure relies on the multi-layer semantic alignment mechanism introduced in the distillation stage.

HIERARCHICAL SEMANTIC MODELING AND REPRESENTATION OF PROGRAMMING KNOWLEDGE

In order to realize the systematic expression of programming course knowledge, this paper constructs a hierarchical semantic modeling system, which is divided into three parts: concept layer, syntax layer and task layer.

The concept layer defines basic entities such as weaving parameters, control variables, data nodes and algorithm symbols; the syntax layer describes the dependency relationship between programming statements and process logic; the task layer integrates the operation procedures, data processing and control strategies in the course modules to form a multi-layer semantically associated teaching task chain. The semantic representation adopts a dual embedding structure. The textual semantics is encoded by the Transformer model to generate a contextual representation, and the symbolic semantics is generated by the Graph Convolutional Network (GCN) to generate a structured representation [21,22]. Assuming that the course text collection is D, and the vocabulary is V, then ithe contextual semantic representation of the corpus is:

$$h_i = \text{Transformer}(x_i) \tag{1}$$

In formula (1), x_i is the input text sequence, h_i is the hidden state vector of the encoding layer. The semantic entity set is represented as a graph structure $G = (V, E)$, where V is the set of semantic nodes, and E is the set of semantic relationship edges. Its graph embedding is represented by GCN as:

$$H_v = \phi\left(\sum_{u \in N(v)} \frac{1}{c_{vu}} W H_u\right) \tag{2}$$

In formula (2), $N(v)$ is the set of neighbor nodes of a node v, c_{vu} is the normalization coefficient, W is the convolution weight matrix, and $\phi(\cdot)$ represents the nonlinear activation function.

The three layers of semantics are unified into a comprehensive semantic representation through the fusion function:

$$H = \alpha H_t + (1 - \alpha) H_g \tag{3}$$

In formula (3), $\alpha \in [0, 1]$ is the semantic fusion coefficient, which is used to balance the weights of text semantics H_t and graph semantics H_g. The multi-layer semantic knowledge graph finally constructed realizes the semantic consistency mapping between the concept, grammar and task layers [23,24]. To measure the proximity between different semantic nodes, a node-level semantic similarity function is defined S_{node}:

$$S_{\text{node}}(a, b) = \frac{H_a \cdot H_b}{\|H_a\| \|H_b\|} \tag{4}$$

In formula (4), H_a and H_b are the semantic embedding representations of nodes a, b, respectively. This similarity is used for semantic alignment and transfer loss calculation in the distillation stage, providing highly consistent semantic features for knowledge transfer [25,26]. The hierarchical order inherent in the semantic knowledge graph is embedded into each representation as a level index that preserves the directed relation among concept, syntax and task semantics. This structural order generates alignment signals that guide the embedding updates and restrict their movement during training. The hierarchy therefore acts as an active constraint that shapes the fused representation and prepares the embeddings for hierarchical alignment in the distillation stage.

KNOWLEDGE DISTILLATION AND LIGHTWEIGHT OPTIMIZATION MECHANISM BASED ON MULTI-LAYER SEMANTIC ALIGNMENT

The multi-layer semantic alignment loss in this study is a constraint that regulates the distances among the embeddings of the three semantic layers so that the student model maintains the structural ordering of the teacher model during distillation. This paper uses ChatGPT as the teacher model and the student model after structural compression as the target. It designs a knowledge distillation mechanism for multi-layer semantic alignment to achieve semantic-level knowledge transfer and model lightweighting [27,28]. The teacher model outputs a high-dimensional semantic distribution, and the student model performs aligned learning in the same semantic space to ensure multi-layer semantic consistency.

The pruning constraint is applied during distillation because structural reduction must occur while the hierarchical semantic relations are being transferred. The sparsified parameter space restricts redundant activation routes and prevents the student model from forming embedding transitions that deviate from the ordered progression established by the teacher model. Post-training pruning would remove this alignment trajectory, whereas pruning inside distillation preserves the semantic structure formed during the transfer process.

Assume that the semantic distribution of the teacher model is P_t, the student model is P_s, and the soft target temperature is T. The distillation loss is defined as:

$$L_{kd} = \sum_i P_t^T \log \frac{P_t^T}{P_s^T} \tag{5}$$

In formula (5), $P_t^T = \text{softmax}(z_t/T)$, $P_s^T = \text{softmax}(z_s/T)$, z_t and z_s are the logits outputs of the teacher and student models respectively. To strengthen the consistency between semantic layers, a multi-layer semantic alignment loss is introduced:

$$L_{\text{align}} = \sum_{l=1}^{3} \beta_l \| H_t^l - H_s^l \|_2^2 \tag{6}$$

In formula (6), H_t^l and H_s^l are the embedding representations of the teacher and student models at the lth layer, β_l is the level weight coefficients, and satisfy $\sum_l \beta_l = 1$. The semantic knowledge graph determines the alignment path for each layer by supplying the structural relations that define how embeddings at different levels should converge. The alignment of concept, syntax and task embeddings follows the dependency pattern encoded in their respective subgraphs, and the cross-level links restrict the gradient direction so that the student model preserves the ordered semantic transitions present in the teacher model. The hierarchy therefore governs the alignment trajectory rather than serving as a simple grouping of data.

The overall optimization goal of the student model is:

$$L = L_{ce} + \lambda_1 L_{kd} + \lambda_2 L_{\text{align}} \tag{7}$$

In formula (7), L_{ce} is the cross entropy loss of the student model on the true label, λ_1, λ_2 are the loss balance coefficient. The complete objective propagates gradients along the inter-level relations encoded in the knowledge graph, and this restricts the representational drift across semantic levels. The concept–syntax and syntax–task dependencies confine the cross-level updates and establish a unified alignment path that mirrors the hierarchical structure. The hierarchy thus directly influences how the student model reforms the semantic space during distillation. To reduce the complexity of the model, parameter pruning constraints are introduced. Let the student model parameter matrix be Θ, the pruning mask matrix be M, and the effective parameters after pruning be $\Theta' = M \odot \Theta$, where $\odot$ represents the element-by-element product. The pruning regularization term is defined as:

$$L_{sp} = \gamma \| \Theta' \|_1 \tag{8}$$

In formula (8), γ is the sparsification coefficient. This constraint achieves parameter sparsification and model compression while maintaining semantic fidelity [29,30].

The sparsification constraint guides the gradient flow toward parameters that contribute to concept–syntax and syntax–task mapping, reducing embedding drift across semantic levels. The pruning mask limits the internal variation of the student model and strengthens the correspondence between the multi-layer semantic distributions of the teacher and student models, forming a unified structural and semantic reduction process.

In order to balance the stability of the initial distillation and the generalization ability in the later stage, a dynamic temperature adjustment strategy is adopted:

$$T_t = T_0 + (T_{\max} - T_0) \left(\frac{t}{T_{\text{total}}} \right)^{\eta} \tag{9}$$

In formula (9), t is the current training step number, and η is the adjustment index. As training progresses, the temperature gradually increases, first maintaining the transfer stability of the teacher's semantic distribution and then enhancing the independent learning ability of the student model. Through joint optimization and dynamic temperature regulation, the student model achieves lightweight structure and efficient transfer while maintaining semantic consistency [31,32].

The interaction between alignment and pruning forms a compact semantic space that preserves long-range coherence. The reduced parameter routes suppress shifts that disturb the semantic chain connecting the three levels, allowing the student model to retain the hierarchical structure of the teacher model even under reduced capacity.

INTELLIGENT QUESTION-ANSWERING GENERATION MODULE FOR SEMANTIC CONSISTENCY IN PROGRAMMING TEACHING

The distilled student model uses the programming semantic embedding as input to construct a semantic consistency-driven question-answer generation module [33,34]. The module consists of four core processes: question encoding, semantic retrieval, answer generation, and consistency judgment. Let the input question be Q, whose encoding is represented by h_q, and the question-answer level semantic similarity S_{QA} is calculated with the knowledge graph node embedding H_v:

$$S_{QA}(Q, v) = \frac{h_q \cdot H_v}{\|h_q\| \|H_v\|} \tag{10}$$

When $S_{QA}(Q, v)$ is not lower than the threshold δ, the node v is included in the candidate knowledge set. The question-answer generation stage uses conditional language modeling to output the answer sequence $Y = (y_1, \ldots, y_T)$. The generation probability is defined as:

$$P(Y \mid Q) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, h_q) \tag{11}$$

The generated objective function is:

$$L_{\text{gen}} = -\sum_{t} \log P(y_t \mid y_{<t}, h_q) \tag{12}$$

In order to constrain the semantic consistency between the answer and the question, a semantic discriminator is introduced $D([\hat{h}_q; \hat{h}_y])$, whose output is defined as:

$$D = \text{sigmoid}(W_d[\hat{h}_q; \hat{h}_y] + b_d) \tag{13}$$

In formula (13), sigmoid $(\cdot)$ is the discriminant activation function, $\hat{h}_q$ and $\hat{h}_y$ are the semantic embeddings of the question and the generated answer respectively.

The semantic consistency judgment mechanism functions as an independent discriminative module that receives the question embedding and the generated answer embedding from the shared semantic space established during hierarchical distillation. The module applies a projection layer that maps the two embeddings into a unified latent space. A bilinear interaction layer then measures their internal correspondence through weighted semantic interactions. The resulting interaction vector is processed by a scoring unit that outputs a consistency value reflecting the degree of semantic alignment. The mechanism is optimized jointly with the answer generation process so that the gradients from the consistency score regulate the semantic trajectory of the generated output. The level indices derived from the semantic knowledge graph are fused into the latent representation to preserve the hierarchical structure required by programming semantics. This mechanism forms a stable constraint that maintains semantic alignment and hierarchical coherence throughout the question-answering process.

The discriminant loss is:

$$L_{\text{cons}} = -[r \log D + (1-r)\log(1-D)] \tag{14}$$

In formula (14), r represents the semantic consistency label. The overall optimization goal is:

$$L_{\text{total}} = L_{\text{gen}} + \mu L_{\text{cons}} \tag{15}$$

To support semantic continuity in multi-round question-answering, a contextual semantic adaptation mechanism is introduced and a semantic memory matrix is defined:

$$M_t = \beta M_{t-1} + (1-\beta)h_t \tag{16}$$

In formula (16), β represents the memory decay coefficient. This structure ensures contextual association and semantic consistency across multiple rounds of question-answering. Through this mechanism, the model can achieve intelligent question-answering output with high-precision semantic matching and logical consistency in the programming teaching scenario, providing stable semantic support and transfer capabilities for the intelligent teaching system [35,36].

III. SEMANTIC TRANSFER EXPERIMENT AND PERFORMANCE EVALUATION IN PROGRAMMING TEACHING SCENARIO

DATA SOURCE AND EXPERIMENTAL SETUP

To verify the effectiveness of the ChatGPT knowledge distillation system based on multi-layer semantic alignment proposed in this paper in the task of intelligent question-answering in computer programming, the experimental data were completely selected from publicly available datasets, covering four aspects: programming semantic understanding, question and answer generation, process data analysis, and material performance calculation, to ensure the reproducibility and data compliance of the research.

Data source and composition, the experimental corpus is integrated from the following four public datasets: CodeSearchNet Dataset provides code snippets and natural language description pairing samples in multiple programming languages, which are used for programming semantic modeling and algorithm logic understanding training of the model. Stack Overflow VQA contains a large number of programming-related question and answer samples and topic labels, which are used for supervised training of intelligent question-answering modules and multi-round dialogue generation tasks. UCI Textile Industry Dataset contains production parameters, quality inspection records and process control variables in the textile industry, which are used to verify the generalization ability of the semantic distillation mechanism in the semantic migration task of structured process data. Materials Project Dataset contains the physical and chemical properties, crystal structure and computational simulation results of fiber and fabric-related materials, which are used to support semantic migration experiments for material performance calculation tasks. After unified formatting and semantic cleaning, the four types of data total approximately 118,000 valid samples, covering four semantic types of natural language description, program code, control parameters and experimental attributes, forming a multi-layer semantic fusion training corpus.

Data Preprocessing and Hierarchical Division: To support multi-layer semantic distillation modeling, samples were divided into three levels based on knowledge complexity and semantic hierarchy: the concept level, which included 41,000 variable definitions, algorithm descriptions, and process parameter descriptions; the syntax level included 52,000 program statements, control structures, and algorithm logic samples; the task level included 25,000 complete programming tasks, data processing flows, and material property calculation examples. Additionally, 10,000 "question-answer-knowledge point" triplets were extracted from the Stack Overflow corpus for supervised learning in the question-answer generation module. All sample were split into training set, validation set and test set with 8:1:1 ratio.

Model Training and Distillation Parameter Settings: this paper uses ChatGPT-4 as teacher model, and student model used light-weighted Transformer-based architecture with about 680 million parameters. Multi-layer semantic alignment loss function was adopted in distillation process, to make teacher-student model embeddings consistent in concept, grammatical and task levels. The layer weight coefficients were $\{0.3, 0.4, 0.3\}$. The dynamic temperature initial value was 1.0, and the scaling exponent was 0.15. This paper used AdamW as optimization algorithm with learning rate as 3×10^{-5}. The batch size was 32 and the maximum epoch was 50. In the process of model compression, this paper adopted the constraint of parameter pruning, and set the sparsification factor as 0.25. Sparse mask was used to achieve structural compression while preserving semantics. The distillation training process was divided into two parts, and the distillation training process in the first 30 epochs, and structural compression redistillation optimization in the next 20 epochs. Experimental Environment and Hardware Environment. The experimental environment is based on the Ubuntu 22.04 operating system and uses two NVIDIA A100 GPUs (video memory 80GB), AMD EPYC 7742 CPU and 256GB system memory. Model training and inference were implemented in the PyTorch 2.2 framework. The graph semantic modeling module used Deep Graph Library (DGL), and the semantic retrieval and similarity calculation module used vector indexing based on FAISS (Facebook AI Simplicity Search). To examine the suitability of ChatGPT as the teacher model, a preliminary comparison experiment was conducted in which a code-pretrained model with similar parameter scale served as an alternative teacher. The student model distilled from the code-pretrained teacher exhibits weaker semantic alignment at the concept layer and produces unstable embedding transitions at the syntax layer. The distillation curve shows an increase in intermediate fluctuations, and the final syntactic alignment loss remains higher than the ChatGPT-based distillation. The student model distilled from ChatGPT maintains a smoother convergence trajectory and retains the hierarchical separation among concept, syntax and task semantics more effectively. These results demonstrate that ChatGPT provides a more coherent semantic structure for hierarchical alignment than the domain-specific teacher model, and this benefit becomes prominent when transferring long-range semantic dependencies required for programming teaching tasks. The experimental results can be analyzed from three perspectives: semantic alignment effect, reasoning efficiency and question-answer generation quality to verify the migration ability and application performance of semantic distillation system based on public corpus.

MULTI-LAYER SEMANTIC ALIGNMENT DISTILLATION EFFECT

In the multi-layer semantic distillation process, in order to analyze the alignment degree of teacher model and student model in semantic space, and explore the convergence characteristic of training process, this paper records and analyzes the changing trend of concept layer, grammatical layer, task layer and the whole distillation loss in the process of 50 rounds of training. Under the same semantic embedding constraint, the model went through a process of declining from high to stable, which reflects the fact that the convergence characteristic and the complexity of different semantic layers in transfer learning are different. The dynamic changes in multi-layer loss demonstrate the impact of the semantic alignment mechanism on the stability of model learning during knowledge transfer, as shown in Figure 1:

As shown in the loss trend in Figure 1, the total distillation loss was 1.284 in the first round, decreased to 0.613 in the 20th round, and further stabilized at 0.431 in the 50th round, showing a smoothly decreasing convergence trajectory. The concept layer loss decreased from an initial 0.452 to 0.160, the grammatical layer loss decreased from 0.523 to 0.175, and the task layer loss converged from 0.309 to 0.099, with each layer showing a steady decline. The grammatical layer loss was consistently higher, indicating that the semantic alignment of grammatical structure and logical dependencies was more complex during the distillation process; the task layer loss was the lowest, indicating that the model had developed good abstract expression capabilities in the transfer of high-level task semantics. The smooth decline of the overall loss curve reflects the stability of distillation training and verifies the effective convergence characteristics of the multi-layer semantic alignment mechanism during the knowledge transfer stage.

To examine the distribution characteristics of the student model in the semantic space after multi-layer semantic alignment distillation, this study performed two-dimensional dimensionality reduction visualization based on the semantic embedding representations of the teacher and student models. Using the t-distributed Stochastic Neighbor Embedding (t-SNE) method, the embedding results at the concept, grammatical, and task levels were mapped to observe the spatial clustering between semantic layers and cross-layer transition characteristics. This process was completed within a unified semantic embedding coordinate system. By comparing the spatial distribution differences between the two types of models, the degree of semantic alignment and the continuity of the semantic structure after distillation were revealed. This is shown in Figure 2:

As shown in the semantic embedding distribution results in Figure 2, concept-level samples are concentrated in the upper left region of the t-SNE space, with coordinates ranging from -9.8 to -7.1 for X and 4.0 to 7.2 for Y. This shows a highly clustered distribution, reflecting the stable structure of conceptual semantics in the reduced-dimensional space. The grammatical layer is located in the middle region, with X ranging from -3.5 to 2.5 and Y ranging from -2.4 to 2.9, and contains the most samples. This indicates that the semantic structure of this layer is complex and the feature differences are relatively subtle. After distillation, the student model forms an embedding similar to that of the teacher model in this region. Task-level samples are mainly located in the lower right region, with X ranging from 4.3 to 6.6 and Y ranging from -4.3 to -2.3, showing a relatively discrete distribution. It shows that the higher semantics can have larger embedding fluctuations in abstract expression stage. In summary, the three layer semantic clusters transfer from the upper left to lower right in space, and the embedding of teacher model and student model are very adjacent. It shows that the multi-layer semantic alignment distillation process can effectively keep the hierarchy of semantic space and transfer consistency. These convergence results reflect the

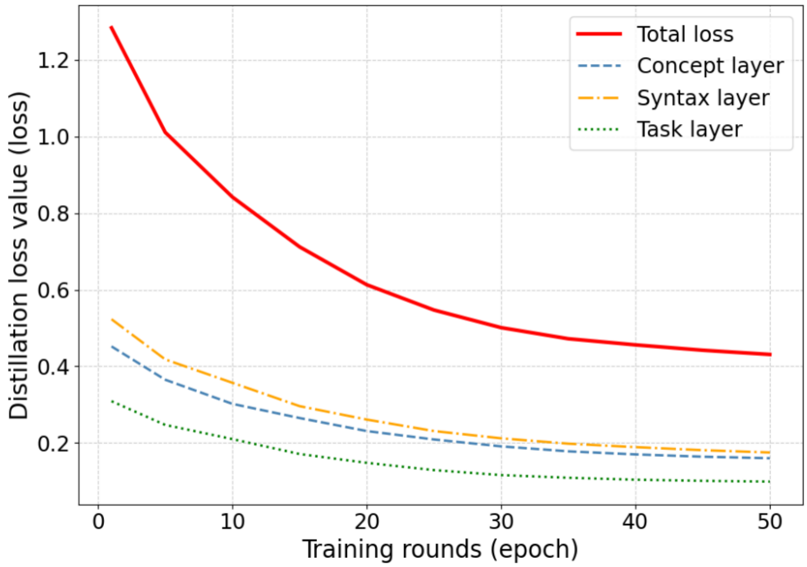

FIGURE 1. Loss convergence curve of the multi-layer semantic alignment distillation process

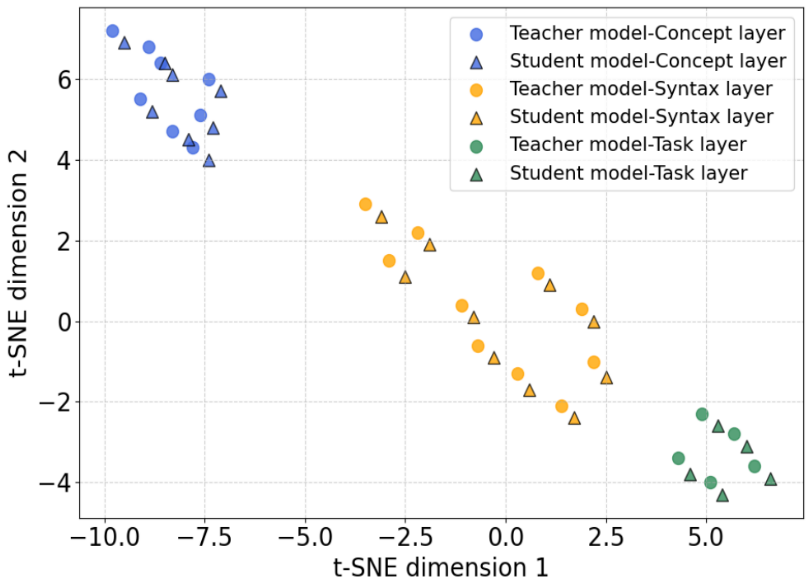

FIGURE 2. Visualization of semantic embedding distribution of teacher and student models

stabilization of semantic transfer, yet their significance must be confirmed through practical question-answering behavior. The following evaluation extends the analysis from loss reduction to observable answer quality, allowing the effect of hierarchical alignment to be assessed in real tasks.

MODEL LIGHTWEIGHT PERFORMANCE EVALUATION

In model lightweighting research, compression rate is often used to characterize the overall reduction brought about by parameter pruning and distillation, while the number of parameters reflects the actual size of the compressed model. The two are not strictly equivalent during the experiment, because sparse constraints at different levels, weight quantization errors, and non-uniform pruning strategies can lead to differences between the set compression rate and the final parameter retention ratio. In order to accurately describe this mapping from theoretical compression targets to actual structural changes, this section records the number of model parameters and multiple inference performance indicators under multiple sets of compression configurations, revealing the comprehensive impact of lightweighting on inference efficiency, stability, and batch processing response, as shown in Figure 3 below:

Figure 3 shows that as the compression ratio increases from 0 % to 80 %, the number of model parameters decreases from 680 million to 160 million, and the average inference time decreases from 2.09 seconds to 1.19 seconds, indicating that compression significantly reduces the computational load. The fastest inference time fluctuates within 1.94-1.12 seconds, and the slowest inference time fluctuates within 2.28-1.37 seconds. The fluctuation is relatively large, which shows that lightweighting can largely influence the peak latency. While the inference standard deviation decreases to 0.08 seconds in the moderate compression range, it shows that the inference stability can be improved when the model is moderately sparsified. When compression ratio continues to increase than 68%, the standard deviation rises to 0.14 seconds. It shows that too much pruning can lead to fluctuations in graphics memory scheduling. The average batch time decreases to 1.59 seconds. It shows that when compression ratio increases, the computational bottleneck can change in I/O (Input/Output) and scheduling. The overall trend shows that model lightweighting can lead to performance improvement. While there is a boundary between compression ratio and performance stability in the later part, it shows that the balance point between parameter size and inference response is reached. The computational changes introduced by lightweighting show its impact on execution, yet the suitability of the compressed structure depends on its behavior in question-answering tasks. The next section presents performance indicators that reflect this behavior directly.

VERIFICATION OF QUESTION-ANSWERING SEMANTIC CONSISTENCY AND KNOWLEDGE TRANSFER EFFECT

Since the convergence patterns in distillation only describe internal learning behavior, their practical value must be examined through the quality of generated answers. This evaluation reveals how the learned semantic structure influences end-user performance. After training the multi-layer semantic distillation system, this paper performs question-answing evaluation on three kinds of questions: logic question, algorithm question and data question. In addition, in order to analyse the performance of teacher model and student model in semantic transfer, this study selects four kinds of basic indicators: semantic consistency, answering accuracy, semantic fidelity and logical coherence to establish compared system to reflect degree of preservation and structural generalization ability of knowledge distillation on different semantic level of questions. Figure 4 shows the response characteristics and performance differences of semantic transfer across different semantic dimensions, providing a comprehensive comparison of the teacher and student models across various question types.

As shown in Figure 4, the teacher model outperforms the student model in all indicators. The semantic consistency and answer accuracy are 0.938 and 0.931 respectively, indicating that the student model has a good ability to maintain semantic structure. The student model obtains 0.902 and 0.905 for both metrics in the other two question types. High convergence level shows that the semantic alignment process after distillation transfers the semantic representation of teacher model. The four metrics for algorithmic questions are generally lower than those for logic and data questions. The semantic consistency of student model decreases to 0.857 and logical coherence is 0.848. It shows that semantic distillation has higher structural complexity and more semantic compression loss in algorithmic logic and control statement levels. The student model can achieve a balance between parameter compression and semantic transfer, and the student model's semantic consistency and answering accuracy are close to the teacher model level. It verifies the effectiveness of multi-layer semantic distillation method.

TEACHING SCENARIO ADAPTABILITY AND SYSTEM APPLICATION

In the adaptability verification link of the teaching system, in order to visually reflect the multi-dimensional effect of the model in different types of courses, this study carried out dimensional quantitative analysis of the intelligent question-answering system based on semantic distillation optimization in a unified dimension. A cross-scenario performance matrix was constructed based on five indicators: knowledge coverage, semantic transfer stability, system operational reliability, student satisfaction, and teaching promotion index. The four typical courses of Programming Fundamentals, Weaving Process Control, Material Performance Calculation, and Data Analysis and Simulation were collected. The indicator value was obtained by extracting the score value, aiming to reflect the overall balance characteristics of the system's teaching knowledge coverage, semantic transfer robustness, and educational support effect as shown in Figure 5:

Figure 5 shows that compared with other courses, Weaving Process Control course has higher indicators in most items, and its knowledge coverage is 0.908, and the semantic transfer stability is 0.937. It shows that when the knowledge coverage and semantic hierarchy are clear, the mapping of stable transfer is more likely to be formed. The satisfaction of Data Analysis and Simulation course was 0.918, and it was consistent with the system operation reliability of 0.901, which was close to 1, showing that the interactive consistency of processing multi-source data and semantic fusion was good. The Teaching Facilitation Index of Material Properties Calculation module was 0.841, which was the lowest among the four courses, indicating that the knowledge abstraction degree was high and the physical parameter was complex. In addition, the system indicator value of other courses was between 0.841 and 0.937, and the distribution was relatively stable, and the variance was not large. It shows that the semantically distilled model has strong universal adaptability and teaching reliability in various teaching scenarios.

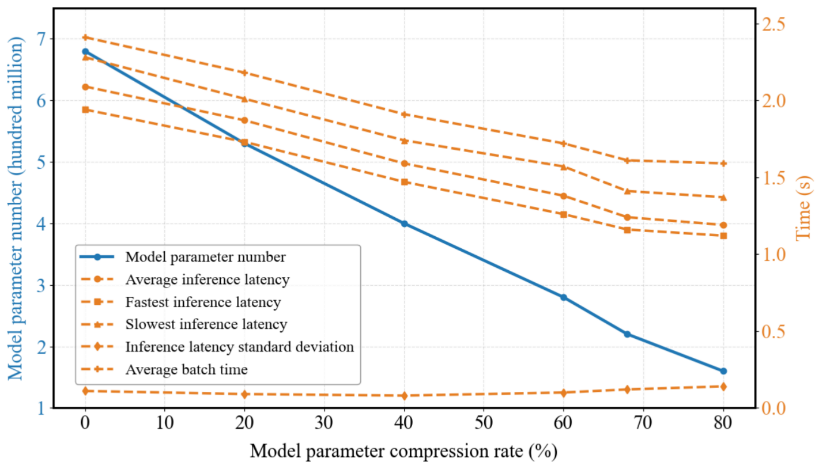

FIGURE 3. Dual vertical axis curve chart of model lightweight performance evaluation

FIGURE 4. Comparison of semantic consistency and knowledge transfer effects between the teacher model and the student model under different question types. Figure 4 (1). Teacher Model; Figure 4 (2). Student Model

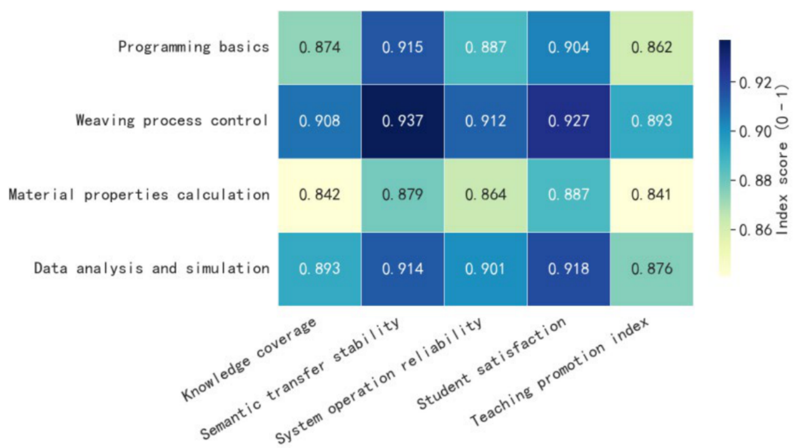

FIGURE 5. Thermal diagram of dimensional comprehensive performance of system adaptability in teaching scenario

IV. CONCLUSION

The findings obtained from the hierarchical alignment evaluation and the comparative distillation experiment confirm that ChatGPT offers a semantic structure that supports stable multi-level transfer. Its embedding space preserves cross-level coherence that cannot be fully reproduced by code-focused models, which tend to generate concentrated syntactic distributions and limited conceptual variability. ChatGPT supplies a representational environment in which conceptual descriptions, syntactic patterns and task-level reasoning are tightly connected, and this cohesion strengthens the fidelity of hierarchical semantic transfer in programming education scenarios. This paper aims to leverage ChatGPT to improve the effectiveness and stability of knowledge transfer in programming courses. By building multi-layer semantic knowledge graph and multi-layer semantic alignment distillation method, unified modeling and transfer of concept, grammatical and task level semantics are realized. The experimental results show that the total loss of semantic convergence in the process of distillation drops from 1.284 to 0.431, and the stability of semantic convergence is very strong. In the model compression experiment, the number of parameters is compressed from 680 million to 160 million, and the average inference time drops from 2.09 seconds to 1.19 seconds. At the same time, computational efficiency is greatly improved without compromising the stability of semantics. In the teaching adaptability evaluation, the system obtains the semantic transfer stability of 0.937 in the Weaving Process Control course and the student satisfaction of 0.918 in the Data Analysis and Simulation course. These observations link loss convergence with practical answer quality, confirming that the distillation process produces stable performance in real question-answering scenarios. The stability and application value of the model in teaching practice are verified. The experimental results show that the multi-layer semantic alignment mechanism can effectively improve the consistency of hierarchical representation and transfer of knowledge distillation, and provide a scalable technical route for lightweight intelligent question-answering system. However, there are still some limitations in this research. Firstly, the experimental corpus scenario is still idealized. Secondly, the covered complex interdisciplinary semantics are not enough. Finally, the current dynamic question-answering memory mechanism is still based on fixed weights. In the future, we will introduce multimodal semantic reinforcement learning and adaptive distillation strategies to make the model more responsive to the unique semantic changes in the industry classroom and build a more flexible and open intelligent teaching knowledge transfer system for specialized technical content.

AUTHOR CONTRIBUTIONS

Conceptualization –Haiyan Liu, Yanna Wei and Penghua Zhu; methodology – Haiyan Liu, Yanna Wei and Penghua Zhu; investigation – Haiyan Liu and Penghua Zhu; writing-original draft preparation – Haiyan Liu, Yanna Wei and Penghua Zhu. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research was supported by Research Projects of the 14th Five-Year Plan for Educational Science under the Hebei Provincial Department of Education, "Research on the Construction of Industry-Education Integration Community in the Field of Aerospace Cybersecurity Testing" (No. 2503105).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] A. Keune, "Material syntonicity: Examining computational performance and its materiality through weaving and sewing crafts," Journal of the Learning Sciences, vol. 31, no. 4-5, pp. 477-508, 2022, doi: 10.1080/10508406.2022.2100704.

[2] J. Kärnä-Behm and E. Harjuniemi, "Interactive textiles: Learning e-textiles with higher education art and design students," FormAcademisk, vol. 16, no. 1, pp. 1-15, 2023, doi: 10.7577/formakademisk.5017.

[3] J. Jin and M. Kim, "GPT-empowered personalized eLearning system for programming languages," Applied Sciences, vol. 13, no. 23, Art. no. 12773, 2023, doi: 10.3390/app132312773.

[4] M. Dabingaya, "Analyzing the effectiveness of AI-powered adaptive learning platforms in mathematics education," Interdisciplinary Journal Papier Human Review, vol. 3, no. 1, pp. 1-7, 2022, doi: 10.47667/ijphr.v3i1.226.

[5] S. Al Faraby, A. Adiwijaya, and A. Romadhony, "Review on neural question generation for education purposes," International Journal of Artificial Intelligence in Education, vol. 34, no. 3, pp. 1008-1045, 2024, doi: 10.1007/s40593-023-00374-x.

[6] C. R. Navas Bonilla, L. M. Viñan Carrasco, J. C. Gaibor Pupiales, and D. E. Murillo Noriega, "The Future of Education: A Systematic Literature Review of Self-Directed Learning with AI," Future Internet, vol. 17, no. 8, pp. 366, 2025, doi: 10.3390/fi17080366.

[7] Y. Fan, H. Zhang, R. Li, Y. Wang, G. Zhang, H. Tan, et al., "Weakly-supervised explainable question answering via question aware contrastive learning and adaptive gate mechanism," Information Sciences, vol. 697, Art. no. 121763, 2025, doi: 10.1016/j.ins.2024.121763.

[8] H. Yang, L. Xu, C. Liu, and L. Huangfu, "Query-oriented two-stage attention-based model for code search," Journal of Systems and Software, vol. 210, Art. no. 111948, 2024, doi: 10.1016/j.jss.2023.111948.

[9] J. Liu, C. Zhang, J. Guo, Y. Zhang, H. Que, K. Deng, et al., "Ddk: Distilling domain knowledge for efficient large language models," Advances in Neural Information Processing Systems, vol. 37, pp. 98297-98319, 2024, doi: 10.52202/079017-3119.

[10] K. Huang, X. Guo, and M. Wang, "Towards efficient pre-trained language model via feature correlation distillation," Advances in Neural Information Processing Systems, vol. 36, pp. 16114-16128, 2023.

[11] Z. Jiang, J. Araki, H. Ding, and G. Neubig, "How can we know when language models know? On the calibration of language models for question answering," Transactions of the Association for Computational Linguistics, vol. 9, pp. 962-977, 2021, doi: 10.1162/tacl_a_00407.

[12] D. Wang, Q. Huang, M. Jackson, and J. Gao, "Retrieve what you need: A mutual learning framework for opendomain question answering," Transactions of the Association for Computational Linguistics, vol. 12, pp. 247-263, 2024, doi: 10.1162/tacl_a_00646.

[13] I. H. Hsiao and C. Y. Chung, "AI-infused semantic model to enrich and expand programming question generation," Journal of Artificial Intelligence and Technology, vol. 2, no. 2, pp. 47-54, 2022, doi: 10.37965/jait.2022.0090.

[14] M. Yousef, K. Mohamed, W. Medhat, E. H. Mohamed, G. Khoriba, and T. Arafa, "BeGrading: Large language models for enhanced feedback in programming education," Neural Computing and Applications, vol. 37, no. 2, pp. 1027-1040, 2025, doi: 10.1007/s00521-024-10449-y.

[15] Y. Wu, H. Zhu, C. Wang, F. Song, H. Zhu, Y. Chen, et al., "Programming knowledge tracing based on heterogeneous graph representation," Knowledge-Based Systems, vol. 300, Art. no. 112161, 2024, doi: 10.1016/j.knosys.2024.112161.

[16] N. Li, Q. Shen, R. Song, Y. Chi, and H. Xu, "MEduKG: A deep-learning-based approach for multi-modal educational knowledge graph construction," Information, vol. 13, no. 2, pp. 91, 2022, doi: 10.3390/info13020091.

[17] R. Cantini, A. Orsino, and D. Talia, "Xai-driven knowledge distillation of large language models for efficient deployment on low-resource devices," Journal of Big Data, vol. 11, no. 1, pp. 63, 2024, doi: 10.1186/s40537-024-00928-3.

[18] C. Jia, "Adversarial moment-matching distillation of large language models," Advances in Neural Information Processing Systems, vol. 37, pp. 112184-112216, 2024, doi: 10.52202/079017-3562.

[19] K. He, N. Pu, M. Lao, E. M. Bakker, and M. S. K. Lew, "Dual selective knowledge transfer for few-shot classification," Appl. Intell, vol. 53, no. 22, pp. 27779-27789, 2023, doi: 10.1007/s10489-023-04994-7.

[20] P. Khodaee, H. L. Viktor, and W. Michalowski, “Knowledge transfer in lifelong machine learning: A systematic literature review,” Artificial Intelligence Review, vol. 57, no. 8, pp. 217, 2024, doi: 10.1007/s10462-024-10853-9.
[21] B. Liu, W. Guan, C. Yang, Z. Fang, and Z. Lu, “Transformer and graph convolutional network for text classification,” International Journal of Computational Intelligence Systems, vol. 16, no. 1, pp. 161, 2023, doi: 10.1007/s44196-023-00337-z.
[22] Y. YaPing, Z. U. Abideen, A. Ali, T. Aoun, T. Mazhar, and T. Shahzad, “A hybrid model combining GCN transformer and Word2Vec for Chinese sequence labeling with deep linguistic knowledge,” International Journal of Speech Technology, vol. 28, no. 3, pp. 729-743, 2025, doi: 10.1007/s10772-025-10209-w.
[23] Y. Zhao, H. Zhou, A. Zhang, R. Xie, Q. Li, and F. Zhuang, “Connecting embeddings based on multiplex relational graph attention networks for knowledge graph entity typing,” IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 5, pp. 4608-4620, 2022, doi: 10.1109/TKDE.2022.3142056.
[24] H. Zhu, D. Xu, Y. Huang, Z. Jin, W. Ding, and J. Tong, “Graph structure enhanced pre-training language model for knowledge graph completion,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 4, pp. 2697-2708, 2024, doi: 10.1109/TETCI.2024.3372442.
[25] G. Yang, S. Yu, Y. Sheng, and H. Yang, “Attention and feature transfer based knowledge distillation,” Scientific Reports, vol. 13, no. 1, Art. no. 18369, 2023, doi: 10.1038/s41598-023-43986-y.
[26] W. Jooste, R. Haque, and A. Way, “Knowledge distillation: A method for making neural machine translation more efficient,” Information, vol. 13, no. 2, pp. 88, 2022, doi: 10.3390/info13020088.
[27] G. Guo, L. Han, L. Wang, D. Zhang, and J. Han, “Semantic-aware knowledge distillation with parameter-free feature uniformization,” Visual Intelligence, vol. 1, no. 1, pp. 6, 2023, doi: 10.1007/s44267-023-00003-0.
[28] X. Liang, L. Wu, J. Li, T. Qin, M. Zhang, and T. Y. Liu, “Multi-teacher distillation with single model for neural machine translation,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 992-1002, 2022, doi: 10.1109/TASLP.2022.3153264.
[29] Z. Yang, Y. Zhang, D. Sui, Y. Ju, J. Zhao, and K. Liu, “Explanation guided knowledge distillation for pre-trained language model compression,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 23, no. 2, pp. 1-19, 2024, doi: 10.1145/3639364.
[30] H. Li, J. Zhang, H. Shen, K. Cheng, and X. Huang, “KEFT: Knowledge-Enhanced Fine-Tuning for Large Language Models in Domain-Specific Question Answering,” Transactions of the Association for Computational Linguistics, vol. 13, pp. 1056-1067, 2025, doi: 10.1162/TACL.a.31.
[31] Y. Lin, S. Yin, Y. Ding, and X. Liang, “ATMKD: Adaptive temperature guided multi-teacher knowledge distillation,” Multimedia Systems, vol. 30, no. 5, pp. 292, 2024, doi: 10.1007/s00530-024-01483-w.
[32] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, “Panda: Prompt transfer meets knowledge distillation for efficient model adaptation,” IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 9, pp. 4835-4848.https://doi.org/10.1109/TKDE.2024.3376453, 2024.
[33] W. Li, J. Xu, and Q. Chen, “Knowledge distillation-based multilingual fusion code retrieval,” Algorithms, vol. 15, no. 1, pp. 25, 2022, doi: 10.3390/a15010025.
[34] J. Roh, M. Kim, and K. Bae, “Towards a small language model powered chain-of-reasoning for open-domain question answering,” ETRI Journal, vol. 46, no. 1, pp. 11-21, 2024, doi: 10.4218/etrij.2023-0355.
[35] S. Liu, X. Guo, X. Hu, and X. Zhao, “Advancing generative intelligent tutoring systems with GPT-4: Design, evaluation, and a modular framework for future learning platforms,” Electronics, vol. 13, no. 24, pp. 4876, 2024, doi: 10.3390/electronics13244876.
[36] J. Wang, Y. Zhang, and W. Liu, “Question answering system based on the combination of large language model and knowledge graph,” Applied Intelligence, vol. 55, no. 15, pp. 1000, 2025, doi: 10.1007/s10489-025-06828-0.