Computer Science • 26(3) 2025 https://doi.org/10.7494/csci.2025.26.3.6673

Ali Sarabadani
Kheirolah Rahsepar Fard
Hamid Dalvand

# GDPKG-LLM: INTEGRATING GENE, DISEASE, AND PHARMACOGENOMICS KNOWLEDGE GRAPHS FOR COGNITIVE NEUROSCIENCE USING LARGE LANGUAGE MODELS

**Abstract** *Using the structures of large language models (LLMs) in creating knowledge graphs to understand more about the relationship between the entities of cognitive and biological sciences has become a hot research topic. Due to the great knowledge behind the curtain and the deep connections of this research, it is not possible to use the traditional approaches of machine learning and deep learning. In this study,the main goal is to create a comprehensive and integrated knowledge graph(KG) from the combination of three knowledge sources: Gene Ontology (GO), Disease Ontology (DO), and PharmKG. Large Language Models (LLMs) have been used to create this knowledge base. The main purpose of this KG is to understand the relationships between genes, diseases, and drugs. The proposed approach, GDPKG-LLM, has several key steps, including entity matching, similarity analysis, graph alignment, and using GPT-4. GDPKG-LLM was able to extract more than 16,800 nodes and 838,000 edges from these three knowledge bases and provide a rich KG. This graph provides meaningful relationships, making it a valuable resource for future research in personalized medicine and neuroscience. The reviewed evaluation criteria show the superiority of GDPKG-LLM, which strengthens the validity of this model.*

**Keywords** knowledge graph, large language models, cognitive neuroscience, biomedical knowledge integration, cognitive neuroscience knowledge graph

**Citation** Computer Science 26(3) 2025: 1–29

## 1. Introduction

Knowledge Graphs are employed in various disciplines and applications and cover many real-world scenarios [52]. Many of these graphs, citation graphs [21] and product graphs [10]), have nodes that are associated with features of the text, which lead to occurrences attributed to the text. For example, in the Ogbn-products dataset [21], each node represents a product, and its corresponding textual description is considered a node attribute. These graphs have been widely used in countless domains, from social network analysis [28] and information retrieval [59] to a diverse range of natural language processing tasks. Among the most important knowledge graphs, we can mention biomedical graphs, which are effective for discovering complex relationships between biological systems, diseases, and drugs. Considering the prevalence of Text Attributed Graphs (TAG), our goal is to display this graph in three knowledge sources: GO, DO and PharmKG. Graph Neural Networks(GNN) [30] have emerged as a de factor technique for handling graph-structured data, often using a message-passing pattern to capture graph structure efficiently. Conventional pipelines usually use asymmetric shallow embeddings such as Bag-of-Words [41] and Word2Vec [32] to encode textual information, which has challenges. Recent studies show that these low-content embeddings have limitations, such as the inability to capture polysemous words [39] and the lack of semantic information [31], which may lead to less optimal performance in downstream tasks. Compared to these scale-free context-less text embeddings, LLMs provide extensive context-aware knowledge and superior semantic understanding capability through a pre-training process on large-scale text sets [32]. This knowledge gained from pre-training has led to a wave of revolutions for downstream NLP tasks [57]. Examples such as ChatGPT and GPT4 [2], equipped with hundreds of billions of parameters, show superior performance in many text-related tasks from different domains. Given the exceptional ability of these LLMs to process and understand textual data, a related question arises: Can we use the knowledge of LLMs to compensate for the lack of contextual knowledge and semantic understanding inherent in conventional GNN pipelines? Approaches based on LLMs have replaced conventional methods, thanks to the knowledge gained through pre-training and the success of LLMs in tasks with implicit graph structures. Recent studies have shown that LLMs are effective in tasks such as recommendations, ranking, and multi-hop reasoning, where they are adopted for final predictions. This success serves as a reassurance of the effectiveness of LLMs in handling complex tasks.

The development of transformer-based pre-trained language models, Natural Language Processing (NLP) and tasks in this area have significantly changed the hands of the changes. These architectures include encoder-like models (e.g., BERT [25]), decoder models (e.g., GPT [40]), and encoder-decoder models (e.g., T5 [14]). Transformer-based language models, such as OpenAI's GPT-4 [2], Google's Gemini [45], PalM [12], Microsoft's Phi 3 [1], and Meta LLaMA [46], are called LLM. These models, shown in Figure 1, are trained on large-scale transformers containing

billions of learnable parameters to support various abilities to enable agents, including perception, reasoning, planning, and action [50].

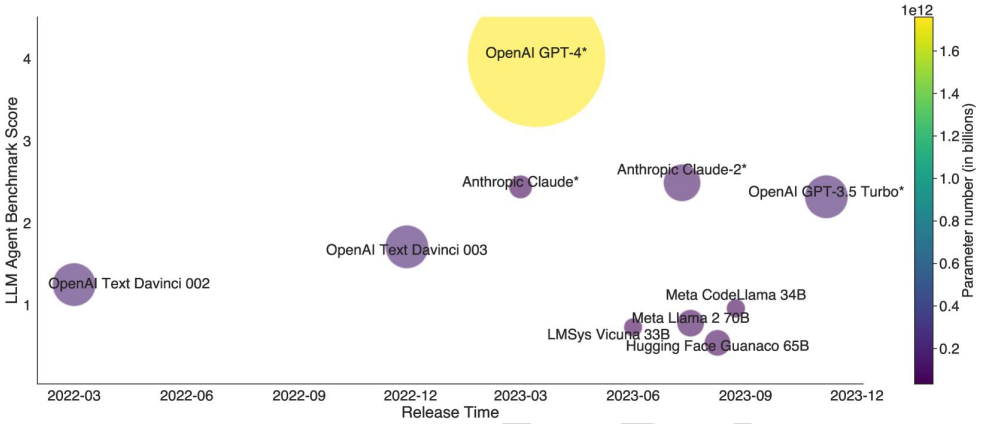


**Figure 1.** LLMs and their agentic abilities [53]

In contrast to existing methods, we propose a novel approach for constructing knowledge graphs by leveraging the capabilities of LLMs. Our method involves three key steps: 1) extracting relevant entities and relationships from textual data using LLM-based processing; 2) refining and integrating the extracted knowledge into structured graph representations; and 3) optimizing graph connectivity through link prediction techniques to enhance inference capabilities. By incorporating these components, our approach enables a more dynamic and scalable construction of knowledge graphs, addressing limitations in traditional rule-based and embedding-based methods. Section 3 provides an in-depth exploration of each step in detail.

The main contribution of this study is the development of GDPKG-LLM. This novel approach is designed for combining information from three knowledge bases: GO, DO, and PharmKG. GDPKG-LLM not only preserves the essential relationships in the original dataset but also introduces new and previously undiscovered relationships. This innovative solution is a significant step forward in the field of knowledge graph applications.

## 2. Related works

KGs have proven to be a powerful tool for presenting and integrating complex biomedical data. These graphs facilitate efficient data integration, helping researchers discover new relationships between entities such as diseases, genes, and drugs. Some famous biomedical KGs include Bio2RDF [5], and CTKG by Chen et al. [20].

The field of biomedical knowledge processing has also made great progress with LLMs. These models improve the interpretation of medical literature, clinical notes, and diagnostic support. Well-known LLMs used in healthcare include DeepMind's

MedIC [24], Microsoft's BioGPT [29], TrialGPT [23], BioBart [56], and BioMistral [27]. These models have been highly influential, especially in areas such as drug discovery, patient data analysis, and clinical decision-making.

[3] validates the integration of LLM to create DIAMOND-KG, a KG that captures drug indications along with their clinical context. This study found that using a complex text extraction strategy can significantly improve the quality of the extracted data. The results show that 71% of the drug indications have at least one relevant clinical reference. They emphasize the importance of reference data in improving the usefulness of KG [3].

Integrating LLM with KG in fields such as medicine, neuroscience, and biomedical is expanding rapidly. This provides new opportunities for knowledge acquisition and application in the healthcare field. Researchers have developed advanced platforms, such as Mann et al. [49], which combine existing medical knowledge sources to create a comprehensive KG to help find practical treatments based on known symptoms or disease.

Fast LLM methods have also gained attention in clinical concept extraction [38]. Although these methods allow for more accurate extraction of medical concepts from unstructured data, challenges still remain, especially in separating drug codes with relevant medical context. Although there has been a lot of progress, many automated methods still fail to elucidate these important context details. Recent investigations indicate that LLM has huge potential for various graph-based tasks, including relationship extraction, KG completion, and querying [37]. For example, Khorashadizadeh et al. [26] performed a qualitative analysis of the model. This study only used decoders such as ChatGPT and Bard in the context of KG integrity and biomedical questions. The study found that although LLMs like ChatGPT are promising in automatic KG extraction, the computational cost is still high.

With the advent of generative AI, KG evaluation has become an important area of research. The study shows that LLM can be used to automatically estimate KG health models. This presents the possibility of replacing human verification with generative factors, which makes the process more eficient and scalable [8]. Moreover, LLM has been used to evaluate the relationships of class members in public KGs such as Wikidata and Calligraph, showing that these models can help knowledge engineers improve KG [4].

In the field of biomedical applications, various techniques have been used to improve KG operations through semantic similarity search using embedded vectors; for example, some methods based on embedded search are followed by nearby data retrieval. Query classification was used to organize the filters and summarize LLMs [42] [13]. Other advanced methods include neighbourhood descriptor, rewriting, breaking down the logical structure of a query, and fine-tuning the model weights using KG source training data [19] [51] [11] [48].

LLM has shown dramatic performance improvements in a variety of medical tasks, including disease diagnosis and disease prediction using electronic medical

records (EMR); for example, Jiang et al. [22] created a patient-specific KG processed with a bidirectional attention-enhanced graph neural network (BAT GNN), which improves Medical decision-making. RAM-EHR turns multiple knowledge sources into text format using co-training that improves information extraction to improve medical predictions [54]. DR.KNOWS improves diagnostic accuracy and interpretation by combining KG and clinical diagnostic logic models built with the Unified Medical Language System (UMLS) [16]. Additionally, REALM integrates data from clinical records and multivariate time series using Retrieval-Augmented Generation (RAG), which increases efficiency in complex medical cases.

# 3. Methodology

The methodology section consists of the following subsections that collectively describe the approach taken in the research. It begins with Data Collection and Preprocessing section, where relevant data from various sources is gathered, cleaned, and formatted for integration. Next Knowledge Graph Representation section involves structuring the knowledge graph using entities and relationships defined through domain specific ontologies. Following this, Graph Expansion with LLMs section utilizes large language models to expand the graph by adding new entities and relationships identified from unstructured data. Lastly GDPKG-LLM section combines the expanded graph with existing representations to improve consistency precision and knowledge representation leveraging the capabilities of LLMs for validation and improvement.

In this section, we present the methodology used in our study, detailing the key steps involved in constructing and refining the knowledge graph. Our approach consists of three main stages: 1) data collection and preprocessing, where we integrate and structure information from publicly available knowledge graphs; 2) knowledge extraction and graph construction, leveraging LLMs to identify meaningful relationships and entities; and 3) graph optimization and link prediction, which enhances the connectivity and inference capabilities of the generated graph. Each of these stages is described in detail in the following subsections, providing a step-by-step breakdown of the proposed approach.

## 3.1. Data collection and preprocessing

In this work, we utilized three publicly available knowledge graphs: Gene Ontology (GO), Disease Ontology (DO), and PharmKG. The specific versions used are as follows: GO (release 2025-02-01), DO (release 2025-01-15), and PharmKG (version 2.0, released on 2024-12-20). These datasets were obtained from their official repositories to ensure consistency and reproducibility.

### 3.1.1. Data sources

In this section, we discuss the data collection process. For this study, we utilize three knowledge graphs to ensure comprehensive analysis:

**Gene Ontology (GO):** This dataset contains 43,000 nodes and 75,000 edges that represent genetic functions. Each node in this graph represents the function of a gene or a protein in the body.

**Disease Ontology (DO):** This dataset also contains 11,200 nodes and 8,800 edges that map diseases with their biological and molecular causes. Each node in this dataset represents a disease.

**PharmKG:** 7,600 nodes and 500,000 edges related to drug and gene interactions have been collected in this dataset. This KG is slightly different from the other two graphs that map pharmacogenomic knowledge. This graph shows the interactions between drugs, genes and diseases. Furthermore it shows the effects of drugs on genes and how genetic factors affect drug responses.

The details of these three KGs are given in Table 1.

**Table 1**
Details of GO, DO, and PharmKG knowledge graphs

| Name | # Node | # Node Types | # Edges | # Edge Types |
|---|---|---|---|---|
| Gene Ontology | 43 K | 3 | 75 K | 4 |
| Disease Ontology | 11.2 K | 1 | 8.8 K | 2 |
| PharmKG | 7.6 K | 3 | 500 K | 3 |

### 3.1.2. Data preprocessing

To ensure data quality and relevance, we applied specific filtering criteria to each knowledge graph. The filtering process involved (1) removing incomplete or ambiguous entities that lacked proper annotations, (2) excluding low-confidence relationships based on predefined confidence scores or expert validation, and (3) standardizing entity names and merging duplicates across multiple sources to reduce redundancy. These steps were taken to enhance the integrity of the extracted knowledge while minimizing noise and inconsistencies.

Furthermore, to ensure the accuracy of the data and to create a clean data free of any errors and null values, in this step, preprocessing is done by the LLM model on the data of GO, DO and PharmKG. These pre-processings are as follows:

**Tokenization and Division:** In this step, tokenization is applied to the textual data in the KGs. For this purpose, two tools, SpaCy[1] and NLTK[2], are used. Tokenization is actually the process of converting text data into sentences and words, and this action is necessary to create text embeddings.

**Clearing Data:** This process is similar to text normalization, in which noise, repetition and nulls are tried to be removed from the data set. This step affects the data quality and improves the performance of LLM.

[1]https://spacy.io/
[2]https://www.nltk.org/

**Normalization:** The process of removing unnecessary entities in the text is called normalization. In fact, in this process, stop words and general writing are removed. In this process, wording is applied to the data to show the common form of the words.

**Adaptation and Balance of the Institution**: Due to the collection of data from specific domains, it is possible for the entities to have several designed forms. For example, "Alzheimer's disease" may be expressed as "AD". They usually use algorithms such as cosine similarity or methods based on word embedding for alignment. In this step, LLM was used for alignment, which avoids repeating data.

## 3.2. Knowledge graph representation

The KG created to display the data includes various items, which we will discuss below:

**Nodes:** Represents the leading entities of the KG, such as genes, diseases or drugs.

**Edges (Relationships):** Shows the direct connection between entities.

**Types of nodes and edges:** In KG, each node and edge can have one type of definition. For example, in the GO, nodes are genes, and edges represent the functional relationships of these genes.

**Multi-layer Display:** Due to the large volume of nodes and edges in KGs, several different displays can be used for this purpose, where each layer represents a specific type of relationship.

**Adjacency Matrix:** Imagine $G(V, E)$ is a simple graph whose number of vertices equals $n$. Suppose the vertices of $G$ are arbitrarily listed as $v_1, v_2, ..., v_n$. According to the list of vertices, adjacency matrix $A$ is an $n * n$ matrix of zeros and ones, which is equal to 1 if $(i, j)$ are adjacent and equal to 0 if they are not adjacent. The matrix can be defined as follows:

$$f(x) = \begin{cases} 1, & \text{if there is a relationship between} v_i \text{ and} v_j. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

**Ontology-based representation:** In ontology structures, relationships are parent and child, for which a hierarchical structure is used.

**Representation of knowledge triple:** Knowledge triple is expressed as $(h, r, t)$ which:

1. $h$(head) is the source node (eg, a gene)
2. $r$(relation) is the relation (eg, "causes")
3. $t$(tail) is the target node (eg, a disease)

**Final pre-processed graph:** After all the pre-processing steps are completed, the data is transformed into a multi layered, uniform KG, which all entities and relationships are coordinated. This chart is ready for entering LLMs for further knowledge extraction and analysis.

### 3.2.1. Combining and merging knowledge graphs

In this section, we combine three KGs: GO, DO, and PharmKG to create a comprehensive KG. This combination is achieved using a LLM with the main goal of identifying and overlapping nodes and edges as well as merging unique nodes from each graph.

**Node Matching:** To reduce the number of nodes, a similarity-based merging approach is applied. The similarity between two nodes is measured using cosine similarity, a simple and effective method for comparing the textual descriptions of nodes. The cosine similarity between two nodes is calculated as:

$$cosine_{simililarity}(v_i, v_j) = \frac{v_i.v_j}{||v_i||||v_j||}, \tag{2}$$

where $vi$ and $v_j$ are the feature vectors for nodes $v_i$ and $v_j$. Nodes with a cosine similarity score above a certain threshold are merged, ensuring that only semantically related nodes are combined. Bedsides, domain-specific constraints from Gene Ontology and Disease Ontology are incorporated, prioritizing nodes that share hierarchical relationships, which further ensure that unrelated entities are not merged. This method is tested experimentally and demonstrated that it effectively preserve the accuracy and structure of the ontology despite reducing the total number of nodes.

**Edge Matching:** The concept of matching may also be seen at the edges. This means that edges can be conceptually the same but relationally different. Semantic similarity and LLM techniques can be used to identify these relationships.

### 3.2.2. Node, edge matching and adjacency matrix update

In this step, similar nodes are merged after they are found. Assuming that $V_{GO}$ represents GO nodes, $V_{DO}$ represents DO nodes, and $V_{PharmKG}$ represents pharm nodes, the following relationship can be used for integration:

$$V_f = V_{GO} \bigcup V_{DO} \bigcup V_{PharmKG}. \tag{3}$$

In the same way, this relationship can be used for edges and similar edges can be merged and expressed as follows:

$$E_f = E_{GO} \bigcup E_{DO} \bigcup E_{PharmKG}, \tag{4}$$

When similar nodes from different graphs are identified and merged according to the investigated flow, the corresponding elements in the adjacency matrix $A$ should also be updated. The final adjacency matrix $A_f$ for the combined graph is updated as follows:

$$A_f = A_{GO} + \Delta A_{GO} + A_{DO} + \Delta A_{DO} + A_{PharmKG} + \Delta A_{PharmKG}, \tag{5}$$

Where $\Delta A_{ij} = 1$ if a new relationship is identified, and 0 otherwise.

### 3.2.3. Using LLMs for new relationship discovery

In this section, we describe how LLMs benefit from discovering and introducing new relationships between nodes that do not exist in traditional knowledge accounts. This process is necessary for expanding the final integrated account with new perspectives and interactions that go beyond basic information.

**New Relationship Suggestion with LLMs:** LLM is used to predict the new probability of a new relationship r between nodes $h$ (head) and $t$ (tail) in the graph. For each pair of nodes, LLM analyzes the semantic similarity and contextual similarity based on the source. External textual data, such as scientific articles and clinical reports and determines a confidence score based on the analysis if LLM predicts a new relationship between these two entities:

$$P(r_{new}) = LLM(h, t), \tag{6}$$

where probability $P(r_{new})$ represents the belief that there is a new relationship r between h and t. This relationship will be added to the graph only if $P(r_{new}) \geq \tau$ , where $\tau$ is the initial criterion that ensures that only high-confidence relationships are included.

**Probabilistic Modeling for New Relationship Discovery:** To further improve the process of finding new relationships We introduce a probabilistic model to estimate the probability of discovering new relationships between nodes. This method provides a more rigorous mathematical basis than LLM's predictions and helps refine the relationship discovery process. We assume that the probability of a new relationship between two nodes h and t follows a gaussian or exponential distribution, based on the distance between the feature vectors of the nodes. Distances are calculated using cosine similarity or other similarity measures and the probability of new values. The relationship is exemplary as follows:

$$P(r_{new}) = 1 - exp(-\frac{||h - t||^2}{2\sigma^2}), \tag{7}$$

Where $||h - t||$ is the distance between node $h$ and feature vector t and $\sigma$ is the parameter controlling the spread of the distribution. The closer the nodes are, the longer the graph, the more likely the relationship is to be predicted. If $P(r_{new}) \geq \tau$ , the relationship is added to the final graph.

**Updating the Adjacency Matrix for New Relationships:** LMMs can be used to identify new relationships and propositions between nodes that are not expressed in the original graphs. This new relationship is obtained from the analysis of texts and other sources. Therefore, the adjacency matrix can be updated depending on the new relations as follows:

$$A_f^{ij} = A_f^{ij} + \Delta A_{new}^{ij}. \tag{8}$$

### 3.3. Graph expansion with LLMs

The following describes the process of expanding the KG using GPT-4. This section aims to test the power of GPT-4 to extract new relationships from unstructured textual data.

#### 3.3.1. Entity Extraction and scoring

Wordi GPT-4 contains textual information (vectors embedded by GPT-4) related to scientific articles and other textual data related to the three reviewed knowledge bases. These texts are used to identify new entities and new relationships. Following are the relations of adding new entities and new relations:

$$V_f = V_{combined} \bigcup V_{new} \tag{9}$$

$$E_f = E_{combined} \bigcup E_{new} \tag{10}$$

Where $V_{new}$ and $E_{new}$ represents the new entity and relationships identified by GPT-4 respectively.

Credit scoring is a mechanism by which GPT-4 gives a score to each entity and relationship. This score shows the confidence of the model in the correctness and relevance of the extracted relationships and entities. Therefore, a provincial limit can be considered when selecting the extracted relations. For this purpose, the following relationship can be used:

$$Add(v_i, r, v_j) if P(v_i, r, v_j) \geq \tau. \tag{11}$$

#### 3.3.2. Graph expansion and iteration

Another advantage of using GPT-4 is that it provides the discovery of textual and semantic relationships that cannot be discovered explicitly. In fact, these are the same entities and interfaces that are not shown in the textual data.

**Adjacency Matrix Updates:** As new entities and relationships are added to the graph, the graph's adjacency matrix $A_f$ must be updated to reflect these changes. When GPT-4 identifies a new relationship between two entities $v_i$ and $v_j$, the corresponding elements in the adjacency matrix are updated as follows:

$$A_f^{ij} = A_f^{ij} + \Delta A_{new}^{ij}, \tag{12}$$

Where $\Delta A_{new}^{ij} = 1$ if a new relationship is identified, and 0 otherwise.

**Three-fold representation:** Newly added knowledge can be represented as three-fold knowledge, such as $(h, r, t)$, where $h$ represents the head entity, $r$ the relationship, and $t$ the part entity end. These triples allow for efficient knowledge storage and retrieval graphs. Expanding the joint KG is iterative; it involves multiple rounds of extracting entity relationships. Integrating with graphs and updated graph

analysis, each iteration ensures that the graph evolves as new information becomes available, which leads to more accurate information.

Let $G(t)$ represent the state of the KG at the $t-th$ iteration. The graph is updated iteratively by adding new entities and relationships predicted by the LLM and other external data sources. Formally, we define the iterative update as [35]:

$$G^{(t+1)} = G^t + \Delta G^t, \tag{13}$$

where $\Delta G^t$ represents the changes. Then new entities $\Delta V^t$ and new edges $\Delta E^t$ introduced at iteration $t$ are as follows:

$$\Delta G^{(t)} = (\Delta V^t, \Delta E^t). \tag{14}$$

This process continues until a stopping criterion $\epsilon$ is met, where the improvement between iterations becomes sufficiently small:

$$||G^{(T+1)} - G^{(t)}|| < \epsilon, \tag{15}$$

where norm $||G^{(T+1)} - G^{(t)}||$ represents the magnitude of changes introduced during the t-th iteration. If the difference between the graph states at consecutive iterations is less than a predefined threshold $\epsilon$, the iterative process terminates, indicating convergence.

### 3.3.3. Optimization and feedback

During each iteration, new entities and relationships are evaluated using a confidence score generated by the LLM. The confidence score for a new relationship ($r_{new}$) between entities $V_i$ and $V_j$ is denoted as $P(r_{new})$, which is compared against a threshold $\tau$:

$$P(r_{new}) = LLM(V_i, V_j), \tag{16}$$

where $LLM(V_i, V_j)$ represents the output of the LLM, which evaluates the likelihood of a relationship existing between the two entities based on their semantic and contextual similarities. The value of $P(r_new)$ indicates how confident the model is that the relationship $r_new$ is valid. If the confidence score exceeds the threshold, the new relationship is added to the graph:

$$r_{new} \in E^{(t+1)} \text{if} P(r_{new}) \geq \tau, \tag{17}$$

thus, each iteration only incorporates relationships that meet the confidence threshold, ensuring that the graph remains consistent and accurate.

To further improve the accuracy of the graph, an error reduction mechanism is employed during each iteration. For each incorrect or weakly supported relationship

identified in iteration t, the relationship is either corrected or removed based on feedback from domain experts or additional data. Let $E(t)$ represent the set of erroneous edges detected at iteration $t$. The refined graph at the next iteration is given by:

$$G^{(t+1)} = G^T - E^t_{error} + \Delta G^t, \tag{18}$$

where $E^t_{error}$ contains the relationships that have been flagged as inaccurate, and $\Delta G^t$ includes the new relationships and entities added at the same iteration. The rate at which the graph converges to its final state depends on the size of the updates $\Delta G^t$ and the number of iterations. We define the rate of convergence $\rho$ as [35]:

$$\rho = \frac{||G^{t+1} - G^t||}{G^t - G^{t-1}}. \tag{19}$$

If $\rho < 1$ , the process is converging, indicating that the changes introduced in each iteration are decreasing. As $\rho \to 0$, the process reaches a steady state where no significant changes occur, marking the completion of the refinement process.

The feedback loop allows for continuous refinement of the graph. After each iteration, newly added relationships are validated by domain experts or cross-referenced with external databases. If a relationship $r_{new}$ receives negative feedback or is invalidated, its confidence score is adjusted accordingly:

$$P(r_{new}) = P(r_{new}) - \delta P, \tag{20}$$

where $\delta P$ represents the adjustment based on expert feedback. If the adjusted confidence score falls below the threshold $\tau$, the relationship is removed from the graph in the next iteration:

$$r_{new} \notin E^{(t+1)} \text{if} P(r_{new}) < \tau. \tag{21}$$

Iterative graph optimization techniques are used to optimize the graph structure and guarantee consistency. This may involve reordering nodes and edges based on their centrality. Clustering coefficient or other graph theoretical properties, for example, the PageRank algorithm can be used to determine the importance of each node [36]:

$$PR(v_i) = \frac{1-d}{N} + d \sum_{v_j \in In(v_i)} \frac{PR(v_j)}{|Out(v_j)|} \tag{22}$$

where $PR(v_i)$ is the PageRank of node, $d$ is the damping factor, and $N4$ is the total number of nodes.

## 3.4. GDPKG-LLM

The theoretical innovation of this research is the development of the GDPKG-LLM which is built as a combined and multi-dimensional framework that brings together information on genes diseases and pharmacogenomics. This framework is designed as a multi-stage process, leveraging GPT-4 to validate and refine the data, ensuring the creation of a comprehensive and accurate knowledge graph. GDPKG-LLM uses large linguistic representations to map and improve complicated cognitive and implicit communication enabling a more structured representation of complicated interactions. By integrating and coordinating different information sources (such as the core knowledge graphs), GDPKG-LLM collaboratively combines nodes and edges to create a comprehensive knowledge structure. This framework is designed to enhance understanding and cognition in cognitive neuroscience, providing a robust foundation for exploring natural relationships between entities. Table 2 shows the total number of nodes in each category. Also, the number of nodes of the proposed model that uses the combination of these data is shown.

To enhance clarity, we explicitly define the node and edge types in our three integrated knowledge graphs: Disease Ontology, PharmKG, and Gene Ontology.

### Disease ontology

Represents human diseases in a structured classification.

- **Nodes:** Disease categories (e.g., genetic, infectious, metabolic).
- **Edges:**
  - *"is_a"* „ Hierarchical relationship (e.g., *Alzheimer's disease is_a neurodegenerative disease*).
  - *"part_of"* „ Subset relationship (e.g., *Breast carcinoma part_of breast cancer*).
  - *"associated_with"* „ Links diseases to genetic or environmental factors.

### PharmKG

Integrates genes, drugs, and diseases with their interactions.

- **Nodes:**
  - *Drugs* „ Therapeutic compounds.
  - *Genes* „ Biological entities linked to diseases and drug targets.
  - *Diseases* „ Conditions influenced by drugs and genes.
- **Edges:**
  - *"treats"* „ Drug-disease relationship (e.g., *Aspirin treats cardiovascular disease*).
  - *"targets"* „ Gene-drug interaction (e.g., *BRCA1 is targeted by Olaparib*).
  - *"biomarker_for"* „ Gene-disease association (e.g., *TP53 is a biomarker for lung cancer*).

### Gene ontology

Describes biological processes, molecular functions, and cellular components.

- **Nodes:**
  - *Biological Processes* „ e.g., *cell differentiation, immune response.*
  - *Molecular Functions* „ e.g., *enzyme binding, DNA repair.*
  - *Cellular Components* „ e.g., *nucleus, mitochondria.*
- **Edges:**
  - *"is_ a"* „ Subclass relationship (e.g., *Mitochondrial transport is_ a cellular transport*).
  - *"part_ of"* „ Component relationship (e.g., *Ribosome part_ of cytoplasm*).
  - *"regulates"* „ Functional influence (e.g., *p53 regulates apoptosis*).

**Table 2**
Total number of nodes in each category

| Name | # Nodes | # Node Types | # Edges | # Edge Types |
|---|---|---|---|---|
| Gene Ontology | 43K | 3 | 75K | 4 |
| Disease Ontology | 11.2K | 1 | 8.8K | 2 |
| PharmKG | 7.6K | 3 | 500K | 3 |
| GDPKG-LLM | 16.8K | 4 | 83.8K | 5 |

Combining these graphs with GDPKG-LLM results in a graph with 16,800 nodes divided into 4 node types and 83,800 edges distributed among 5 relationship types. Although reducing the number of nodes and edges may seem counterintuitive, it also reflects data optimization and refinement. The integrated KG prioritizes quality and relevance over quantity and ensures that every node and relationship displays accurate and well-preserved information and leads to a more efficient and accurate knowledge base. It gives researchers a deeper but more powerful understanding of the interactions between genes, diseases and pharmacological agents. By combining and optimizing data from provenance graphs, GDPKG-LLM not only captures valuable insights, but it also reduces unnecessary and irrelevant relationships. Thus creating a more connected and efficient framework. This leads to more meaningful analyses, discoveries, and applications in complex biomedical fields. Table 3 shows an overview of knowledge about four areas of biomedical medicine. To clarify the "Protein-Gene Relationships" edge type in Table 3, we define its node and relationship types:

- **Node Types:**
  - *Proteins*: Functional molecules encoded by genes.
  - *Genes*: DNA sequences responsible for protein synthesis and regulation.
- **Edge Type ("Protein-Gene Relationships"):** This edge type represents interactions between genes and their encoded proteins, including:

- *encoded_by*: Links a protein to its encoding gene (e.g., "TP53 protein encoded_by TP53 gene").
- *interacts_with*: Represents interactions between proteins at the genetic level (e.g., "BRCA1 protein interacts_with RAD51 protein").
- *regulates*: Indicates a protein's influence on gene expression (e.g., "NF-kB regulates IL6 gene expression").

Emphasizing the complexity and scope of each dataset, the Gene Ontology (GO) contains approximately 43,000 nodes divided into 3 node types (representing genes, proteins, and biological processes). It also includes about 75,000 edges that capture functional relationships such as gene regulation and protein interactions, making GO an essential resource for understanding biological roles. In contrast, the Disease Ontology (DO) focuses specifically on disease-related information, featuring about 11,200 nodes of a single type (diseases) and 8,800 edges across 2 relationship types, which primarily connect diseases to their symptoms or genetic underpinnings.

PharmKG provides a more complex structure, with 7,600 nodes across three node types representing drugs, targets, and diseases. It also contains approximately 500,000 edges that capture key relationship types, such as drug„target interactions and drug„disease associations, which model the pharmacological connections among these entities.

**Table 3**

Summarization of primary entities and the types of relationships between them in GDPKG-LLM, providing a clear view of the KG's structure

| Node Type | Edge Type |
|---|---|
| Genes | Gene-Disease Interactions |
| Diseases | Drug-Gene Interactions |
| Drugs | Drug-Disease Interactions |
| Proteins | Disease-Protein Associations |
| | Protein-Gene Relationships |

Finally, GDPKG-LLM, which combines data from multiple sources through a LLM, combines data from GO, DO and PharmKG to remain a comprehensive graph. It has 16,800 nodes across 4 node types, comprising biomedical entities that 83,800 edges across 5 relationship types . This diverse and interconnected structure makes GDPKG-LLM a strong tool for advanced biomedical research. They help with complex tasks such as drug discovery, understanding disease mechanisms, and predicting new biomedical connections. In summary, although each knowledge article serves a specific area, GDPKG-LLM leverages the strengths of multiple domains for deeper biomedical insights. It provides the most integrated and detailed framework.

### 3.4.1. Application of GDPKG-LLM in cognitive neuroscience

It integrates GO, DO, and PharmKG in cognitive neuroscience research. It provides comprehensive insights into the relationships between genes, diseases, and drug inter-

actions. This makes this knowledge important for the development of this field. GO plays an important role in understanding the genetic origins of neural processes. By mapping genes and their biological functions, GO helps researchers explore how genes contribute to cognitive functions such as memory, learning, synaptic plasticity, etc. This is a valuable resource for linking genetic variation to cognitive disorders such as schizophrenia and autism spectrum disorders, which leads to treatment intervention.

DO supplements GO with a specific focus on neurological disorders. This graph helps researchers trace diseases such as Alzheimer's and Parkinson's to genetic mutations and provides a structured framework for studying the genetic and biological pathways involved in these conditions. It enables a deeper understanding of how such disorders affect brain function and cognition, thereby supporting more effective treatment strategies. PharmKG adds a pharmacogenomics dimension by mapping interactions among drugs, genes, and diseases, making it especially relevant for personalized medicine. It helps researchers understand how genetic factors influence drug response, allowing treatments to be tailored to each patient's genetic makeup.

Together, these three KGs GO, DO, and PharmKG form a robust framework for cognitive neuroscience research. They enable a more holistic understanding of how genes, diseases, and drugs interact within the brain, driving the development of targeted treatments and personalized interventions for cognitive and neurological disorders.

**Handling LLM Hallucinations:** In the edge discovery process, GPT-4 was utilized to suggest new edges based on textual patterns. To mitigate the risk of hallucinations, we employed a threshold-based approach where only edges with confidence scores above a predefined threshold $\tau$ were included. Furthermore, to ensure accuracy, domain experts were involved in validating GPT-4's suggestions. Experts reviewed the edges with lower confidence scores and those that appeared inconsistent with existing domain knowledge. The validation process revealed that approximately 15% of the edges were rejected, either due to incorrect relationships or lack of supporting evidence. This process allowed us to adjust the threshold $\tau$ and refine the edge discovery mechanism. Besides, we recorded the false-positive and false-negative rates to evaluate the effectiveness of the validation process and improve the model's performance over time.

## 4. Evaluation

Our evaluation of GDPKG-LLM was based on criteria such as precision, recall, F1 score, coverage, graph compatibility, computational efficiency, novelty detection, and expert validation. In the second part, we evaluated the ability of the model to predict links, from criteria such as Mean Rank (MR), Mean Mutual Rank (MRR) and Precision at K (P@K) by different algorithms. Accuracy, precision, recall and F1-score measure the quality of the node and edge alignment processes. The higher the precision, the fewer incorrect alignments (false positives); the higher the recall, the fewer missed alignments (false negatives) [9]:

$$Precision = \frac{True_{positive}}{Actual_{Results}} = \frac{True_{positive}}{True_{positive} + False_{positive}}, \tag{23}$$

$$Recall = \frac{True_{positive}}{Predicted_{Results}} = \frac{True_{positive}}{True_{positive} + False_{negative}}, \tag{24}$$

$$F1_{score} = 2 * \frac{Precision * Recall}{Precision + Recall}, \tag{25}$$

In addition, some other metrics were also used to evaluate the proposed model.

**Coverage:** Refers to the proportion of original KGs (GO, DO, PharmKG) that have been successfully transferred to the final merged KG. This metric evaluates the quantity and diversity of entities (nodes) and relationships (edges) from the source graphs that are preserved after the merging process and is defined as follows [18]:

$$Coverage = \frac{Unique_{Nodes}(N)_{merged} + Unique_{Edges}(E)_{merged}}{\sum_{i=1}^{n}(N_i + E_i)}. \tag{26}$$

**Graph Consistency:** This metric is used to evaluate that the final integrated graph has been able to maintain semantic and logical relationships and is evaluated in two main ways:

1. Semantic Consistency: This metric ensures that similar conceptual nodes are correctly integrated. For this purpose, it uses ontology-based mapping (e.g., using Jaccard similarity, cosine similarity of node embeddings)
2. Structural Consistency: This metric can be evaluated through logical reasoning frameworks such as Description Logic (DL) or OWL-based consistency checks to ensure that the merged graph adheres to formal constraints and does not introduce a cycle.

**Computational Efficiency:** This metric is used for computational efficiency of time and space complexity:

1. Time Complexity: This metric refers to the computational cost of merging nodes and edges. Complexities vary from $O(N \log N)$ (for node-based similarity search) to $O(N^2)$ (for pairwise edge comparison).
2. Space Complexity: This metric refers to the amount of memory consumed by the graph.
3. Execution Time: This metric shows the overall duration of the integration process

**Novelty Score:** Novelty score evaluates the ability of the integrated KG to discover and suggest new and previously unknown relationships between nodes, and it is defined as follows:

$$Novelty_{Score} = \frac{NewlyDiscovered_{Edges}(E)_{LLM}}{Total_{Edges}(E)_{merged}}. \tag{27}$$

### 4.1. Expert validation

Experts in the field play an important role in evaluating the biological and clinical relevance of newly proposed edges. To ensure the reliability of our knowledge graph, we conducted an expert validation process involving domain specialists in cognitive neuroscience and biomedical informatics. A total of three experts participated in the evaluation, each with ten years of experience in the field. The validation process consisted of two key steps: 1) a random selection of a subset of extracted relationships and entities, and 2) a qualitative assessment where experts rated the correctness, relevance, and completeness of the generated knowledge graph on a agree/disagree scale. Any discrepancies were discussed and resolved through consensus. This process ensured that the extracted knowledge was aligned with established domain knowledge and met the expected accuracy standards.

Figure 2 shows the comparison between GO and the proposed GDPKG-LLM approach. GO achieved a better evaluation value in the precision evaluation criterion. The precision of the GO is equal to 0.85, while the proposed GDPKG-LLM reached Precision=0.8. In the recall evaluation criterion, the Go approach reached recall=0.76, and the proposed GDPKG-LLM also achieved recall=0.85. The F1-score of the proposed approach was equal to 0.82, and the GO achieved F1-score=0.80. The GO approach obtained a coverage value 0.70, which is weaker than the proposed approach of 0.10. The proposed approach has performed better than other evaluation approaches. GDPKG-LLM achieved graph consistency=0.77, computational efficiency=0.67, novelty=0.88, and experiment validation=0.90. The GO approach achieved graph consistency=0.62, computational efficiency=0.60, novelty=0.50, and experiment validation=0.90. Also, Figure 3 shows the comparison between DO and the proposed GDPKG-LLM. DO, and GDPKG-LLM obtained almost close results in the two evaluation criteria of precision and Recall. Both models reached Precision=0.80 and Recall=0.85.

The Do approach reached 0.82 in the F1 evaluation criterion, while the proposed approach reached 0.83. Both models are equal in coverage value. In the two evaluation criteria of graph consistency and computational efficiency, DO performed better, and in the two criteria of novelty and experiment validation, the proposed GDPKG-LLM achieved the best results. GDPKG-LLM was able to achieve experiment validation=0.90. A comparative evaluation between GDPKG-LLM and PharmKG is shown in Figure 4. The GDPKG-LLM reached precision=0.80, and the PharmKG reached precision=0.70. In recall evaluation, the proposed approach has performed weaker than PharmKG. F1 of the PharmKG reaches 0.84, while this value is 0.80 for the proposed method. Regarding other evaluation criteria, the proposed approach performed better than PharmKG. Finally, the comparison between all the examined knowledge graphs is shown in Figure 5. In the precision criterion of the Go, in the recall criterion of the PharmKG, and the F1 evaluation criterion, the proposed method obtained the highest results. The proposed approach also obtained the highest results in novelty and expert Validation.

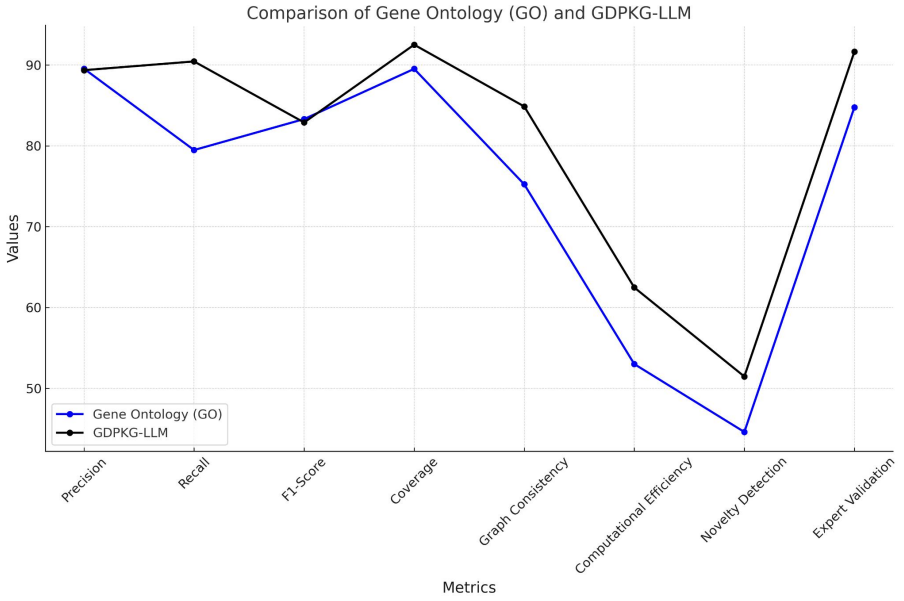


**Figure 2.** Comparison chart between proposed GDPKG-LLM and approach Gene Ontology (GO)

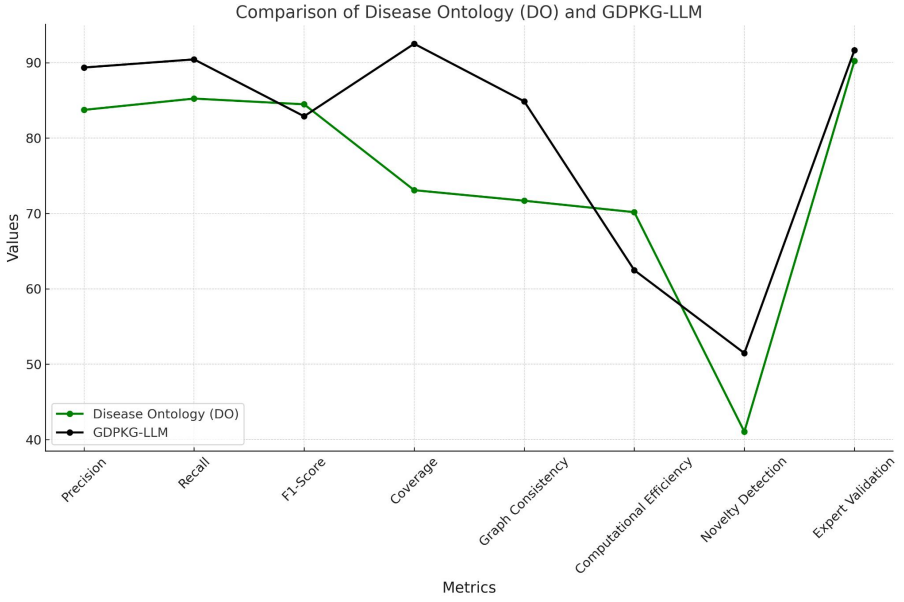


**Figure 3.** Comparison chart between proposed GDPKG-LLM and approach Disease Ontology (DO)

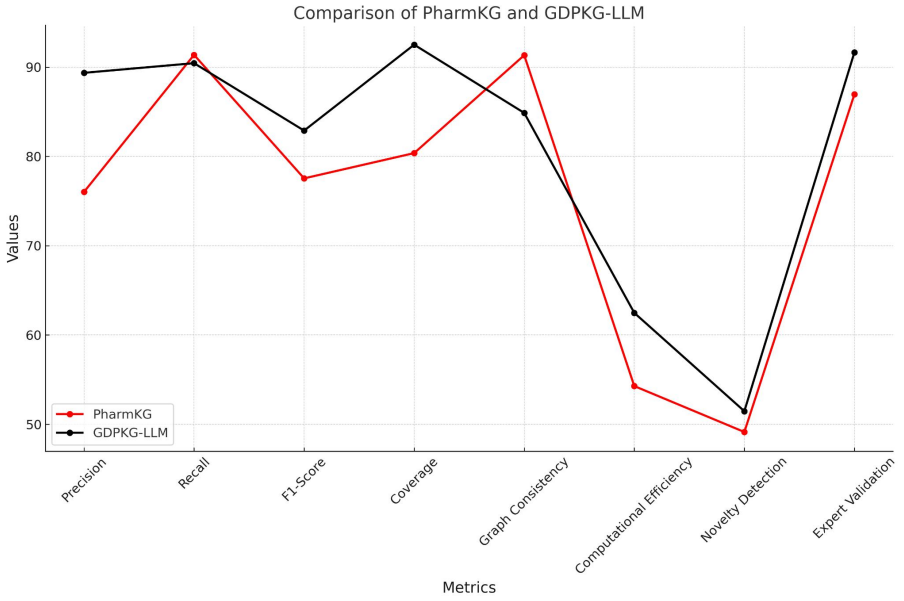


**Figure 4.** Comparison chart between proposed GDPKG-LLM and approach PharmKG

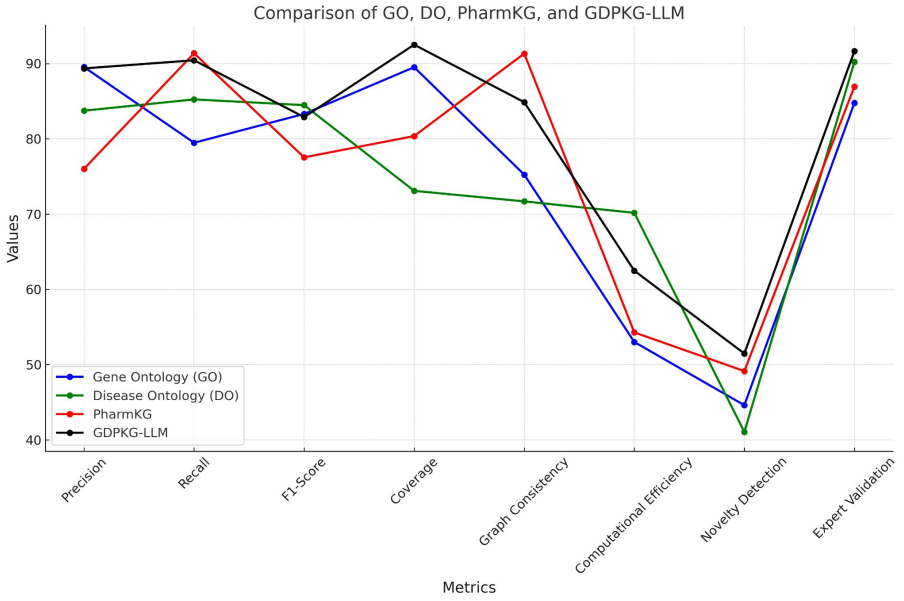


**Figure 5.** Comparison chart between all research approaches

The presence of a higher score suggests that a greater percentage of the relations theorized by the model were put to scientific test. For instance, in the case of GDPKG-LLM, a validation score of 0.9 indicates that the assessed relationships, 90% of them, were proven to be scientifically valid by the specialists. Such a high score demonstrates that the knowledge bases from GO, DO, and PharmKG were effectively integrated without losing the biological relevance of the interconnections and while adding new, plausible connections. Thus, the expert validation score serves as a qualitative indicator of the usefulness and accuracy of the knowledge generated by the model for cognitive neuroscience research (see Table 4).

**Table 4**

Results of evaluation criteria by the proposed approach. The blue color indicates the best performance, while red represents the second-best performance

| Metric | Gene Ontology (GO) | Disease Ontology (DO) | PharmKG | GDPKG-LLM |
|---|---|---|---|---|
| Precision | 89.54 | 83.74 | 76.02 | 89.36 |
| Recall | 79.48 | 85.24 | 91.37 | 90.44 |
| F1-Score | 83.31 | 84.48 | 77.54 | 82.89 |
| Coverage | 89.52 | 73.09 | 80.37 | 92.51 |
| Graph Consistency | 75.24 | 71.69 | 91.33 | 84.87 |
| Computational Efficiency | 53.01 | 70.17 | 54.28 | 62.48 |
| Novelty Detection | 44.61 | 41.05 | 49.14 | 51.47 |
| Expert Validation | 84.78 | 90.22 | 86.95 | 91.65 |

Thanks to the underlying structure of the GDPKG-LLM KG, most of its attributes score higher than other graphs such as GO, DO, or even PharmKG. The precision rate of 89.36 and a recall rate of 90.44 reveal that GDPKG-LLM is not only accurate but also covers the relationships prediction very well. The F1-score of GDPKG-LLM, 82.89, illustrates an average performance as regards the aspect of precision and recall.

GDPKG-LLMs third coverage score is 92.51, which demonstrates the extent of the availability and utilization of varied data sources. Its graph consistency of 84.87 assures the presence of coherent and sensible relations among the entities, while the directed efficiency of 62.48, focuses on balancing performance and complexity.

Moreover, the graph also reveals a pleasing level of novelty detection (51.47) which indicates its potential in plunging into the depth of relationships and possessing the ability to identify patterns that were otherwise unobserved. Furthermore, it does exceptionally well in expert validation (91.65), proving the accuracy of its forecasts.

In conclusion, all the balanced advantages of GDPKG-LLM make it a worthy assistant in biomedical research, particularly in the areas of gene-disease-drug associations.

## 4.2. Link prediction

Different evaluation dimensions were covered by various metrics to comprehensively evaluate the GDPKG-LLM. Nexus anticipation methods were engaged to value GDPKG-LLM's truth inch distinctive and superior connections between nodes and highlight its strength. GDPKG-LLM's reliability and foretelling strength are clearly and practically viewed through this combination. When assessing GDPKG-LLM, we use three important metrics which include MR, MRR and P@K concentrating on the link prediction tasks respectively. These indicators enable us to understand how well the graph can predict links between nodes. In case of performing link prediction tasks with filter applied, we apply three metrics below:

**MR:** Mean Rank is yet another parameter to gauge the efficiency of any information retrieval system. It indicates the average position of the true positive elements in the list of items retrieved. This is done by ranking all items and then computing the average of ranks assigned to the relevant items. More so, mean rank scores are inversely proportional. In other words, better performance is associated with lower mean ranks because the relevant items are positioned more highly on average [34]:

$$MR = \frac{1}{N}\sum_{i=1}^{N} rank_i, \tag{28}$$

where $rank_i$ is the rank position of the $i-th$ relevant item, and $N$ is the total number of relevant items.

**MRR:** With a reference to its practical utility, MRR can be understood better. For example, suppose you are looking for a particular document in a large database. It is mathematically defined as the average of the reciprocal ranks of the first relevant item in the retrieved item list. MRR is useful in cases when it is more important to get the first relevant result, rather than all of them. MRR assesses the rate of retrieval of the first relevant item in a given set of items [17]:

$$MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank_i}, \tag{29}$$

where $rank_i$ is the rank position of the first relevant item for the $i-th$ query, and $N$ is the total number of queries.

**P@K:** P@K indicates the rate of relevant documents present in the top K documents retrieved. In our case, we compute P@K for the three different values of K that K=1, K=3, K=10. P@K measures the proportion of relevant items within the top k results [17]. It assesses the performance of the top k results and is expressed as follows:

$$P@K = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{Number of relevant items in top K}}{K} \tag{30}$$

where $K$ is the number of top items considered, and $N$ is the total number of queries. We employ link prediction methods such as TransE, RotatE, DistMult, ComplEx, ConvE , HolmE which leverage advanced algorithms to infer connections within the graph.

**Table 5**

Comparison of link prediction for SKG-LLM with TransE, RotatE, DistMult, ComplEx, ConvE, and HolmE using MR, MRR, and P@K metrics with Other KGs

| KG | | TransE [7] | RotatE [44] | DistMult [55] | ComplEx [47] | ConvE [15] | HolmE [58] |
|---|---|---|---|---|---|---|---|
| FB15k-237 [6] | MR | 209 ± 0.510 | 178 ± 0.474 | 199 ± 0.496 | 144 ± 0.375 | 281 ± 0.821 | - |
| | MRR | 0.310 | 0.336 | 0.313 | 0.367 | 0.305 | 0.331 |
| | P@1 | 0.217 | 0.238 | 0.224 | 0.271 | 0.219 | 0.237 |
| | P@3 | 0.257 | 0.328 | 0.263 | 0.275 | 0.350 | 0.366 |
| | P@10 | 0.496 | 0.530 | 0.490 | 0.558 | 0.476 | 0.517 |
| WN18RR [33] | MR | 3936 ± 2.478 | 3318 ± 2.147 | 5913 ± 4.826 | 2867 ± 2.842 | 4944 ± 3.896 | - |
| | MRR | 0.206 | 0.475 | 0.433 | 0.489 | 0.427 | 0.466 |
| | P@1 | 0.279 | 0.426 | 0.396 | 0.442 | 0.389 | 0.415 |
| | P@3 | 0.364 | 0.492 | 0.440 | 0.460 | 0.430 | 0.489 |
| | P@10 | 0.495 | 0.573 | 0.502 | 0.580 | 0.507 | 0.561 |
| YAGO3-10 [43] | MR | 1187 ± 2.510 | 1830 ± 0.520 | 1107 ± 2.510 | 793 ± 0.489 | 2429 ± 2.004 | - |
| | MRR | 0.501 | 0.498 | 0.501 | 0.577 | 0.488 | 0.441 |
| | P@1 | 0.405 | 0.405 | 0.412 | 0.500 | 0.399 | 0.333 |
| | P@3 | 0.528 | 0.550 | 0.38 | 0.40 | 0.560 | 0.507 |
| | P@10 | 0.673 | 0.670 | 0.661 | 0.7129 | 0.657 | 0.641 |
| GDPKG-LLM | MR | 241 ± 0.701 | 138 ± 0.323 | 213 ± 0.547 | 115 ± 0.095 | 301 ± 0.864 | - |
| | MRR | 0.324 | 0.287 | 0.333 | 0.294 | 0.347 | 0.268 |
| | P@1 | 0.415 | 0.201 | 0.285 | 0.257 | 0.318 | 0.225 |
| | P@3 | 0.463 | 0.273 | 0.314 | 0.298 | 0.396 | 0.251 |
| | P@10 | 0.572 | 0.309 | 0.345 | 0.461 | 0.544 | 0.319 |

The Table 5 illustrates how GDPKG-LLM compares with other KGs like FB15k-237, WN18RR, and YAGO3-10 cumulatively, by employing various evaluation metrics. As for the MR, GDPKG-LLM is superior to both FB15k-237 and WN18RR, however, it is more efficient than the former model. With regard to MRR, competitive performance can be noticed for GDPKG-LLM with the MRR achieving 0.324, lower that of few models for WN18RR, but most lower than ComplEx and ConvE generally higher than such indicate that the model ranking may not have been consistent across all datasets making it difficult to rank the predictions correctly.

It should be noted that the MR metric is dependent on several parameters that can change its value across different runs. Therefore, the standard deviation values in Table 5 provide a clearer assessment of the stability and consistency of the link prediction models across multiple runs. Since link prediction involves probabilistic computations and can be influenced by factors such as initialization and training variations, reporting the standard deviation helps highlight how much the results fluctuate. This ensures a more comprehensive and reliable evaluation, allowing for better comparisons between models in terms of both accuracy and stability.

Considering P@1 (Precision at 1), GDPKG-LLM shows better results than some of WN18RR and FB15k-237 models getting 0.415 score. However, it does not compare to positive results achieved by some models, for instance, ComplEx on YAGO3-10,

whom precision rates are higher. On the aspect of P@3 and P@10, the output of GDPKG-LLM is inconsistent as in some instances, this model is above some models on FB15k-237 but does not match the performance of models such as ConvE or HolmE on other models consistently. The conclusion is that though the metrics for GDPKG-LLM are reasonable, the power of ranking and the accuracy possess in all the other higher ranked models is missing.

## 5. Discussion

Our proposed method demonstrates promising performance in constructing knowledge graphs, particularly in terms of accuracy and efficiency. Compared to traditional methods, which often rely on manually curated rules or pre-defined ontologies, our approach leverages the flexibility and adaptability of LLMs to dynamically infer relationships. While some existing solutions incorporate deep learning-based embedding techniques, they often require extensive labeled data for training, whereas our model benefits from self-supervised learning capabilities. These advantages position our method as a scalable and effective alternative for automated knowledge graph construction.

In addition to the theoretical benefits discussed, GDPKG-LLM has practical applications that can drive new insights in fields such as neuroscience. For instance, by integrating data from Gene Ontology (GO), Disease Ontology (DO), and PharmKG, GDPKG-LLM can identify previously overlooked gene-disease-drug relationships. One example of this could be discovering novel drug repurposing opportunities for neurodegenerative diseases. By analyzing gene-disease interactions, GDPKG-LLM could suggest existing drugs that target genes linked to Alzheimer's disease, potentially accelerating drug discovery for this devastating condition. Similarly, GDPKG-LLM could be used to identify new biomarkers for neurodegenerative disorders, helping researchers pinpoint therapeutic targets or diagnostic tools. These types of insights demonstrate how GDPKG-LLM can foster new hypotheses and drive research in the neuroscience field.

## 6. Conclusion and future work

The goal of this article is to introduce the GDPKG-LLM model. GDPKG-LLM is a comprehensive graph of GO, DO and PharmKG that uses LLMs to create a single resource for cognitive neuroscience. In this approach, LMMs played a fundamental role in extracting information, nodes, and edges. GDPKG-LLM captures the complex relationships between genes, diseases and drugs. In different stages, the data went through different pre-processing processes to become data free of repetition and noise. Several different evaluation criteria were used to evaluate the proposed model. GDPKG-LLM was able to obtain acceptable results using these evaluation metrics. Clinical trials can be used as additional metadata for future work. This knowledge base has clinical information in text, which can provide a richer KG when integrated

with other knowledge bases. The genomics data set of patients can be used as another data set. GNNs and KG embeddings can also be explored as future works. Integration of deep learning and machine learning approaches can improve the model. Also, optimization-based approaches can better predict the optimal parameters of the model. Using better keywords and extracting more articles allows for presenting a better model. We can maneuver on these issues as future works. Also integrating GDPKG-LLM into personalized medicine frameworks could lead to a more precise diagnosis and therapy approach as the Representation can link genetic disease and pharmacological Information. These improvements pave the way for the development of smarter tools and broader applications in neuroscience research, contributing to future advancements in the field.

## References

[1] Abdin M., Jacobs S.A., Awan A.A., Aneja J., Awadallah A., Awadalla H., Bach N., Bahree A., Bakhtiari A., Behl H., *et al.*: Phi-3 technical report: A highly capable language model locally on your phone, *arXiv preprint arXiv:240414219*, 2024.

[2] Achiam J., Adler S., Agarwal S., Ahmad L., Akkaya I., Aleman F.L., Almeida D., Altenschmidt J., Altman S., Anadkat S., *et al.*: Gpt-4 technical report, *arXiv preprint arXiv:230308774*, 2023.

[3] Alharbi R., Ahmed U., Dobriy D., Łajewska W., Menotti L., Saeedizade M.J., Dumontier M.: Exploring the role of generative AI in constructing knowledge graphs for drug indications with medical context, *Proceedings http://ceur-ws org ISSN*, vol. 1613, p. 0073, 2023.

[4] Allen B.P., Groth P.T.: Evaluating Class Membership Relations in Knowledge Graphs using Large Language Models, *arXiv preprint arXiv:240417000*, 2024. doi: 10.1007/978-3-031-78952-6_2.

[5] Belleau F., Nolin M.A., Tourigny N., Rigault P., Morissette J.: Bio2RDF: towards a mashup to build bioinformatics knowledge systems, *Journal of biomedical informatics*, vol. 41(5), pp. 706–716, 2008. doi: 10.1016/j.jbi.2008.03.004.

[6] Bollacker K., Evans C., Paritosh P., Sturge T., Taylor J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008. doi: 10.1145/1376616.1376746.

[7] Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O.: Translating embeddings for modeling multi-relational data, *Advances in neural information processing systems*, vol. 26, 2013.

[8] Boylan J., Mangla S., Thorn D., Ghalandari D.G., Ghaffari P., Hokamp C.: KGValidator: A Framework for Automatic Validation of Knowledge Graph Construction, *arXiv preprint arXiv:240415923*, 2024.

[9] Canbek G., Taskaya Temizel T., Sagiroglu S.: PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics, *SN Computer Science*, vol. 4(1), p. 13, 2022. doi: 10.1007/s42979-022-01409-1.

[10] Chiang W.L., Liu X., Si S., Li Y., Bengio S., Hsieh C.J.: Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 257–266, 2019. doi: 10.1145/3292500.3330925.

[11] Choudhary N., Reddy C.K.: Complex logical reasoning over knowledge graphs using large language models, *arXiv preprint arXiv:230501157*, 2023.

[12] Chowdhery A., Narang S., Devlin J., Bosma M., Mishra G., Roberts A., Barham P., Chung H.W., Sutton C., Gehrmann S., *et al.*: Palm: Scaling language modeling with pathways, *Journal of Machine Learning Research*, vol. 24(240), pp. 1–113, 2023.

[13] Cm S., Prakash J., Singh P.K.: Question answering over knowledge graphs using BERT based relation mapping, *Expert Systems*, vol. 40(10), p. e13456, 2023. doi: 10.1111/exsy.13456.

[14] Colin R.: Exploring the limits of transfer learning with a unified text-to-text transformer, *J Mach Learn Res*, vol. 21, pp. 140–1, 2020.

[15] Dettmers T., Minervini P., Stenetorp P., Riedel S.: Convolutional 2d knowledge graph embeddings. In: *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018. doi: 10.1609/aaai.v32i1.11573.

[16] Gao Y., Li R., Croxford E., Tesch S., To D., Caskey J., W. Patterson B., M. Churpek M., Miller T., Dligach D., *et al.*: Large language models and medical knowledge grounding for diagnosis prediction, *medRxiv*, pp. 2023–11, 2023. doi: 10.1101/2023.11.24.23298641.

[17] Ge X., Wang Y.C., Wang B., Kuo C.C.J., *et al.*: Knowledge Graph Embedding: An Overview, *APSIPA Transactions on Signal and Information Processing*, vol. 13(1), 2024. doi: 10.1561/116.00000065.

[18] Godsil C., Royle G.F.: *Algebraic graph theory*, vol. 207, Springer Science & Business Media, 2001. doi: 10.1007/978-1-4613-0163-9.

[19] Guo Q., Cao S., Yi Z.: A medical question answering system using large language models and knowledge graphs, *International Journal of Intelligent Systems*, vol. 37(11), pp. 8548–8564, 2022. doi: 10.1002/int.22955.

[20] Himmelstein D.S., Lizee A., Hessler C., Brueggeman L., Chen S.L., Hadley D., Green A., Khankhanian P., Baranzini S.E.: Systematic integration of biomedical knowledge prioritizes drugs for repurposing, *Elife*, vol. 6, p. e26726, 2017. doi: 10.7554/elife.26726.

[21] Hu W., Fey M., Zitnik M., Dong Y., Ren H., Liu B., Catasta M., Leskovec J.: Open graph benchmark: Datasets for machine learning on graphs, *Advances in neural information processing systems*, vol. 33, pp. 22118–22133, 2020.

[22] Jiang P., Xiao C., Cross A., Sun J.: Graphcare: Enhancing healthcare predictions with personalized knowledge graphs, *arXiv preprint arXiv:230512788*, 2023.

[23] Jin Q., Wang Z., Floudas C.S., Chen F., Gong C., Bracken-Clarke D., Xue E., Yang Y., Sun J., Lu Z.: Matching patients to clinical trials with large language models, *ArXiv*, 2023. doi: 10.1038/s41467-024-53081-z.

[24] Kanithi P.K., Christophe C., Pimentel M.A., Raha T., Saadi N., Javed H., Maslenkova S., Hayat N., Rajan R., Khan S.: Medic: Towards a comprehensive framework for evaluating llms in clinical applications, *arXiv preprint arXiv:240907314*, 2024.

[25] Kenton J.D.M.W.C., Toutanova L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of naacL-HLT*, vol. 1, p. 2, Minneapolis, Minnesota, 2019.

[26] Khorashadizadeh H., Mihindukulasooriya N., Tiwari S., Groppe J., Groppe S.: Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text, *arXiv preprint arXiv:230508804*, 2023.

[27] Labrak Y., Bazoge A., Morin E., Gourraud P.A., Rouvier M., Dufour R.: Biomistral: A collection of open-source pretrained large language models for medical domains, *arXiv preprint arXiv:240210373*, 2024. doi: 10.18653/v1/2024.findings-acl.348.

[28] Li Q., Li X., Chen L., Wu D.: Distilling knowledge on text graph for social media attribute inference. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2024–2028, 2022. doi: 10.1145/3477495.3531968.

[29] Luo R., Sun L., Xia Y., Qin T., Zhang S., Poon H., Liu T.Y.: BioGPT: generative pre-trained transformer for biomedical text generation and mining, *Briefings in bioinformatics*, vol. 23(6), p. bbac409, 2022. doi: 10.1093/bib/bbac409.

[30] Ma Y., Tang J.: *Deep learning on graphs*, Cambridge University Press, 2021. doi: 10.1017/9781108924184.

[31] Miaschi A., DellOrletta F.: Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In: *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 110–119, 2020. doi: 10.18653/v1/2020.repl4nlp-1.15.

[32] Mikolov T.: Efficient estimation of word representations in vector space, *arXiv preprint arXiv:13013781*, 2013.

[33] Miller G.A.: WordNet: a lexical database for English, *Communications of the ACM*, vol. 38(11), pp. 39–41, 1995. doi: 10.1145/219717.219748.

[34] Nayyeri M., Cil G.M., Vahdati S., Osborne F., Rahman M., Angioni S., Salatino A., Recupero D.R., Vassilyeva N., Motta E., *et al.*: Trans4E: Link prediction on scholarly knowledge graphs, *Neurocomputing*, vol. 461, pp. 530–542, 2021. doi: 10.1016/j.neucom.2021.02.100.

[35] Nocedal J., Wright S.J.: *Numerical optimization*, Springer, 1999.

[36] Page L.: *The PageRank citation ranking: Bringing order to the web*, Tech. rep., Technical Report, 1999.

[37] Pan S., Luo L., Wang Y., Chen C., Wang J., Wu X.: Unifying large language models and knowledge graphs: A roadmap, *IEEE Transactions on Knowledge and Data Engineering*, 2024. doi: 10.1109/tkde.2024.3352100.

[38] Peng C., Yang X., Yu Z., Bian J., Hogan W.R., Wu Y.: Clinical concept and relation extraction using prompt-based machine reading comprehension, *Journal of the American Medical Informatics Association*, vol. 30(9), pp. 1486–1493, 2023. doi: 10.1093/jamia/ocad107.

[39] Qiu X., Sun T., Xu Y., Shao Y., Dai N., Huang X.: Pre-trained models for natural language processing: A survey, *Science China technological sciences*, vol. 63(10), pp. 1872–1897, 2020. doi: 10.1007/s11431-020-1647-3.

[40] Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I., *et al.*: Language models are unsupervised multitask learners, *OpenAI blog*, vol. 1(8), p. 9, 2019.

[41] Shao Y., Taylor S., Marshall N., Morioka C., Zeng-Treitler Q.: Clinical text classification with word embedding features vs. bag-of-words features. In: *2018 IEEE International conference on big data (big data)*, pp. 2874–2878, IEEE, 2018. doi: 10.1109/bigdata.2018.8622345.

[42] Soman K., Rose P.W., Morris J.H., Akbas R.E., Smith B., Peetoom B., Villouta-Reyes C., Cerono G., Shi Y., Rizk-Jackson A., *et al.*: Biomedical knowledge graph-enhanced prompt generation for large language models, *arXiv preprint arXiv:231117330*, 2023.

[43] Suchanek F.M., Kasneci G., Weikum G.: Yago: a core of semantic knowledge. In: *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, 2007. doi: 10.1145/1242572.1242667.

[44] Sun Z., Deng Z.H., Nie J.Y., Tang J.: Rotate: Knowledge graph embedding by relational rotation in complex space, *arXiv preprint arXiv:190210197*, 2019.

[45] Team G., Anil R., Borgeaud S., Wu Y., Alayrac J.B., Yu J., Soricut R., Schalkwyk J., Dai A.M., Hauth A., *et al.*: Gemini: a family of highly capable multimodal models, *arXiv preprint arXiv:231211805*, 2023.

[46] Touvron H., Lavril T., Izacard G., Martinet X., Lachaux M.A., Lacroix T., Rozière B., Goyal N., Hambro E., Azhar F., *et al.*: Llama: Open and efficient foundation language models, *arXiv preprint arXiv:230213971*, 2023.

[47] Trouillon T., Welbl J., Riedel S., Gaussier É., Bouchard G.: Complex embeddings for simple link prediction. In: *International conference on machine learning*, pp. 2071–2080, PMLR, 2016.

[48] Varshney D., Zafar A., Behera N.K., Ekbal A.: Knowledge grounded medical dialogue generation using augmented graphs, *Scientific Reports*, vol. 13(1), p. 3310, 2023. doi: 10.1038/s41598-023-29213-8.

[49] Wawrzik F., Rafique K.A., Rahman F., Grimm C.: Ontology learning applications of knowledge base construction for microelectronic systems information, *Information*, vol. 14(3), p. 176, 2023. doi: 10.3390/info14030176.

[50] Webb T., Holyoak K.J., Lu H.: Emergent analogical reasoning in large language models, *Nature Human Behaviour*, vol. 7(9), pp. 1526–1541, 2023. doi: 10.1038/s41562-023-01659-w.

[51] Wu Y., Hu N., Bi S., Qi G., Ren J., Xie A., Song W.: Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering, *arXiv preprint arXiv:230911206*, 2023.

[52] Xia F., Sun K., Yu S., Aziz A., Wan L., Pan S., Liu H.: Graph learning: A survey, *IEEE Transactions on Artificial Intelligence*, vol. 2(2), pp. 109–127, 2021. doi: 10.1109/tai.2021.3076021.

[53] Xiong H., Wang Z., Li X., Bian J., Xie Z., Mumtaz S., Barnes L.E.: Converging paradigms: The synergy of symbolic and connectionist ai in llm-empowered autonomous agents, *arXiv preprint arXiv:240708516*, 2024.

[54] Xu R., Shi W., Yu Y., Zhuang Y., Jin B., Wang M.D., Ho J.C., Yang C.: Ram-ehr: Retrieval augmentation meets clinical predictions on electronic health records, *arXiv preprint arXiv:240300815*, 2024. doi: 10.18653/v1/2024.acl-short.68.

[55] Yang B., Yih W.t., He X., Gao J., Deng L.: Embedding entities and relations for learning and inference in knowledge bases, *arXiv preprint arXiv:14126575*, 2014.

[56] Yuan H., Yuan Z., Gan R., Zhang J., Xie Y., Yu S.: BioBART: Pretraining and evaluation of a biomedical generative language model, *arXiv preprint arXiv:220403905*, 2022. doi: 10.18653/v1/2022.bionlp-1.9.

[57] Zhao W.X., Zhou K., Li J., Tang T., Wang X., Hou Y., Min Y., Zhang B., Zhang J., Dong Z., *et al.*: A survey of large language models, *arXiv preprint arXiv:230318223*, 2023.

[58] Zheng Z., Zhou B., Yang H., Tan Z., Waaler A., Kharlamov E., Soylu A.: Low-Dimensional Hyperbolic Knowledge Graph Embedding for Better Extrapolation to Under-Represented Data. In: *European Semantic Web Conference*, pp. 100–120, Springer, 2024. doi: 10.1007/978-3-031-60626-7_6.

[59] Zhu J., Cui Y., Liu Y., Sun H., Li X., Pelger M., Yang T., Zhang L., Zhang R., Zhao H.: Textgnn: Improving text encoder via graph neural network in sponsored search. In: *Proceedings of the Web Conference 2021*, pp. 2848–2857, 2021. doi: 10.1145/3442381.3449842.

## Affiliations

**Ali Sarabadani**
Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran, a.sarabadani@stu.qom.ac.ir

**Kheirolah Rahsepar Fard**
Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran, rahsepar@qom.ac.ir

**Hamid Dalvand**
Department of Occupational Therapy, School of Rehabilitation, Tehran University of Medical Sciences, Tehran, Iran, hamiddalvand@gmail.com

**Received**: 29.10.2024
**Revised**: 10.01.2025
**Accepted**: 23.09.2025