

Let Me Cypher That for You: A Multi-Graph Agent for Deciphering Biomedical Knowledge

Astha Anand

`astha.anand@scai.fraunhofer.de`

Fraunhofer Institute for Algorithms and Scientific Computing SCAI

Negin Babaiha

Fraunhofer Institute for Algorithms and Scientific Computing SCAI

Martin Hofmann-Apitius

Fraunhofer Institute for Algorithms and Scientific Computing SCAI

Marc Jacobs

Fraunhofer Institute for Algorithms and Scientific Computing SCAI

Research Article

Keywords: Cypher query generation, Large language models, Neurodegenerative diseases, Chatbots, Retrieval-augmented generation, COVID-19, Biomedical knowledge graphs, Natural language interfaces

Posted Date: September 30th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10988415/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Let Me Cypher That for You: A Multi-Graph Agent for Deciphering Biomedical Knowledge

Astha Anand^{1*}, Negin Babaiha¹, Martin Hofmann-Apitius^{1,2},
Marc Jacobs¹

^{1*}Bioinformatics Department, Fraunhofer Institute for Algorithms and
Scientific Computing SCAI, Schloss Birlinghoven, 1, Sankt Augustin,
53757, North Rhine-Westphalia, Germany.

²Life Science Informatics Department, Bonn-Aachen International
Center for Information Technology (b-it), University of Bonn, Bonn,
53115, North Rhine-Westphalia, Germany.

*Corresponding author(s). E-mail(s): astha.anand@scai.fraunhofer.de;
Contributing authors: negin.babaiha@scai.fraunhofer.de;
martin.hofmann-apitius@scai.fraunhofer.de;
marc.jacobs@scai.fraunhofer.de;

Abstract

Background: Crosstalk between infectious and non-infectious diseases may contribute to the initiation and progression of neurodegeneration. The COMMUTE project investigates post-COVID-19 multimorbidities and their potential links to an increased risk of neurodegenerative diseases. Hypothesis-driven biomedical knowledge graphs offer a powerful framework to represent and analyze complex relationships and mechanistic links between COVID-19 and neurodegeneration. However, interacting with these graphs typically requires proficiency in Cypher, a graph query language, which poses a technical barrier for many researchers and clinicians.

Results: We developed CGEx (Cypher Generating Expert), an interactive system that enables users to query multiple heterogeneous COVID-19 and neurodegenerative disease knowledge graphs using natural-language questions. CGEx translates user questions into Cypher queries, retrieves relevant sub-graphs, and generates explanatory natural-language responses. The system incorporates a user-feedback mechanism and a dynamically updated few-shot

example store to improve query generation. CGEx2.0 extends the original system by using GPT-5 and providing interactive solution-graph visualization with access to supporting evidence associated with individual relationships. We evaluated the original CGEx and CGEx2.0 using 52 manually curated biomedical questions covering biological pathways, molecular interactions, comorbidity relationships, and multi-hop reasoning. Accuracy was defined as generation of a syntactically and semantically correct Cypher query within three or fewer feedback iterations. CGEx2.0 achieved accuracies of 94.2% and 82.7% on the two graphs, compared with 82.7% and 55.8%, respectively, for the original system.

Conclusions: Our results show that CGEx streamlines the query formulation process by removing the need to learn or manually write Cypher, while uncovering meaningful comorbidity insights. Rather than harmonizing heterogeneous graphs into a single schema, CGEx illustrates how multi-graph agents can serve as a complementary paradigm to harmonization, enabling researchers to query diverse evidence sources directly.

Keywords: Cypher query generation, Large language models, Neurodegenerative diseases, Chatbots, Retrieval-augmented generation, COVID-19, Biomedical knowledge graphs, Natural language interfaces

1 Background

The COVID-19 pandemic has had a profound global impact, affecting millions of individuals across healthcare, economic, and societal dimensions. Beyond the acute phase, its long-term health consequences are now a growing area of concern. A significant number of individuals experience persistent symptoms, a condition recognized by the World Health Organization (WHO) as post-COVID-19 condition or long COVID [1]. Long COVID may be characterized by fatigue, cough, chest tightness, dyspnea, palpitations, myalgia, and arthralgia [2].

Long COVID also affects neurological and cognitive functions [3], including chronic malaise, olfactory and gustatory loss, sleep disorders, anxiety, depression, brain fog, and stroke [4, 5]. One proposed mechanism underlying these associations is chronic neuroinflammation, which activates glial cells and contributes to the progression of neurodegeneration. SARS-CoV-2 has been shown to cross the blood-brain barrier

(BBB), bind to ACE2 receptors, and disrupt BBB integrity, thereby exacerbating neuroinflammation [6]. The heightened cytokine response observed in SARS-CoV-2 infection, marked by elevated levels of pro-inflammatory mediators such as IL-6 and TNF, has also been implicated in neuronal damage [7]. Studies establish that cytokine associated neuroinflammation contributes to the onset of neurodegenerative diseases (NDD) such as multiple sclerosis (MS), Alzheimer’s disease (AD), Parkinson’s disease (PD), and amyotrophic lateral sclerosis (ALS) [8]. To systematically explore such potential mechanistic links between COVID-19 and NDDs, the EU-funded **COMMUTE** project (Co-Morbidity Mechanisms UTilized in Healthcare) [9] brings together artificial intelligence (AI), clinical, and experimental expertise.

However, effectively interacting with corresponding knowledge in graphs remains challenging. Biomedical knowledge graphs (KGs) are frequently implemented in graph database systems and require users to write Cypher queries [10]. Although Cypher provides a powerful mechanism for graph traversal, it creates a barrier to many biomedical researchers unfamiliar with graph database technologies. Even experienced users often face difficulties recalling graph schemas and composing multi-hop queries, particularly in unfamiliar domains.

Research on translating natural language into graph query languages has gained considerable traction. The SpCQL dataset introduced by Guo et al. [11] provides one of the first publicly available benchmarks for text-to-Cypher conversion, consisting of manually annotated natural-language questions paired with Cypher queries for a Neo4j graph. Using a sequence-to-sequence baseline, the study highlights the difficulties with Cypher’s graph-centric constructs and flexible schema, underscoring the need for schema-aware approaches. Litvin et al. [12] developed a tree-based semantic analysis method for converting natural language phrases into formal queries

(SPARQL/Cypher) over ontological graph databases. By first determining the semantic type of a phrase and then populating query templates via decision tree rules, their method offers greater interpretability and was implemented for Ukrainian and Norwegian languages.

More recently, large language models (LLMs) have been applied to graph-query generation. Feng et al. [13] addressed a knowledge graph reasoning question-answering task in the domain of foreign military unmanned systems. Their workflow applies named-entity recognition to identify and link entities in user questions, followed by a structured LLM prompt containing instructions, schema information, few-shot examples, and a verification step for Cypher generation. Another LLM-based Text-to-Cypher interface by Hornsteiner et al. [14] retrieves the schema from the selected database, and uses it as context for LLM-based Cypher generation. The system also uses an automated correction loop in which schema information and database error messages are returned to the LLM to refine unsuccessful queries. Q2Cypher [15] introduced a parameter-efficient fine-tuning strategy using low-rank adaptation with open-source LLMs. Evaluated on the SpCQL dataset, it showed improved performance over prompting-only baselines.

Despite these advances, applying natural-language-to-Cypher methods to heterogeneous, disease-focused biomedical KGs presents additional challenges. Biomedical concepts may be represented differently across graphs, relevant relationships may require multi-hop traversal, and incomplete graph coverage can make it difficult to distinguish query-generation errors from genuine knowledge gaps. Furthermore, generating a syntactically valid query alone does not provide biomedical users with an accessible interpretation of the retrieved evidence.

A central motivation for this work is to enable users to talk to a graph in natural language. This materializes into the system we call **CGEx** (Cypher **G**enerating **E**xpert), which transforms evidence bases into accessible, conversational resources. CGEx leverages large language models (LLMs) [16] within a Retrieval-Augmented Generation (RAG) [17] architecture to translate natural language questions into Cypher queries, execute them against a selected KG, retrieve relevant subgraphs, and generate explanatory responses. It additionally incorporates schema-aware prompt construction; a few-shot example store and user feedback mechanism for iterative refinement; dynamic few-shot example expansion; and transparent prompt display for explainability. Unlike prior work, it couples query generation with execution over the target KG to produce detailed, natural language answers, enabling an end-to-end question answering pipeline. This study describes the design and implementation of CGEx and evaluates its Cypher-generation performance across two heterogeneous COVID-19 and NDD knowledge graphs using a manually curated set of biomedical questions.

2 Methods

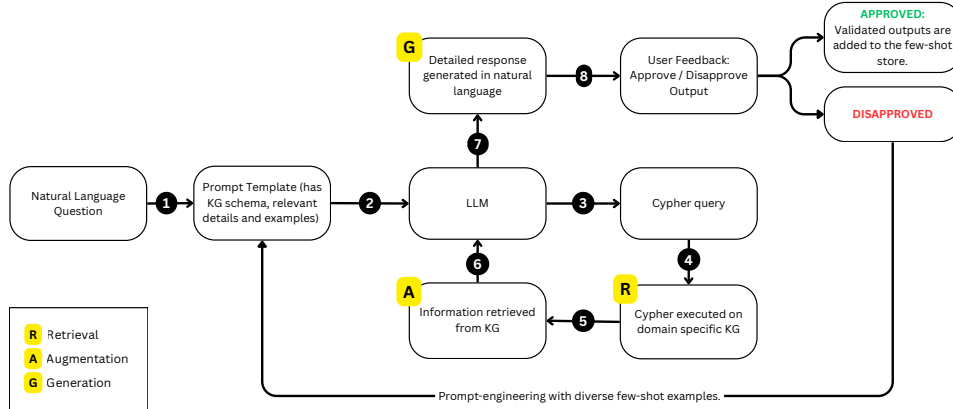


Fig. 1 Cypher Generating Expert System (CGEx) Workflow: A natural language question (1) is processed through a prompt template containing KG schema, relevant details, and examples (2). The LLM converts the question into a Cypher query (3), which is executed on the domain-specific KG (4, R-Retrieval) to obtain relevant information (5). This information is provided to the LLM (6, A-Augmentation) to generate a detailed natural language response (7, G-Generation). The user then provides feedback to approve or disapprove the output (8).

2.1 COMMUTE Knowledge Graphs

All experiments in this study were performed on COVID-NDD comorbidity knowledge graphs developed within the COMMUTE project [9]. The graphs integrate heterogeneous evidence to systematically capture pathways, molecular, and phenotypic relationships linking COVID-19 and neurodegenerative diseases such as Alzheimer’s and Parkinson’s disease.

2.1.1 COMMUTE KG-I

The KG-I (Causal Biological Mechanisms Knowledge Graph) [18] systematically models causal mechanisms underlying disease comorbidities by integrating molecular,

clinical, and mechanistic evidence into a unified, computable framework. It employs a comorbidity-centric approach, encoding directional cause-and-effect relationships between genes, proteins, molecular complexes, pathways, and clinical phenotypes using the Biological Expression Language (BEL) for semantic precision and evidence traceability. Each interaction is represented as a subject–predicate–object triple (e.g., “IL6 directly_increases TNF expression” or “SARS-CoV-2 infection increases blood–brain barrier permeability”), annotated with provenance and supporting literature references. Combining expert-guided text mining, manual curation, and integration of structured databases such as DisGeNET, DrugBank, OpenTargets, and KEGG, the graph captures thousands of mechanistic relationships spanning molecular to clinical levels. Implemented in Neo4j for interactive querying and visualization, KG-I provides a transparent, extensible resource to support mechanistic exploration, hypothesis generation, and biomarker and target discovery in complex comorbid disease contexts. There are a total number of 2,261 nodes and 4,646 edges in this KG.

2.1.2 COMMUTE KG-II

This Hypothesis KG [19] is extracted through the integration of both text-mining approaches and database-driven investigations, combining structured biomedical evidence with insights mined from scientific literature. The text-mining workflow employed PubTator3, an AI-based annotation system from the National Library of Medicine, and Sherpa, a machine-learning platform for relation extraction and BEL encoding, both optimized for identifying causal and associative relationships between biomedical entities such as genes, proteins, chemicals, phenotypes, and diseases. In parallel, curated datasets from major repositories—including DisGeNET, DrugBank, OpenTargets, KEGG, INDRA, and SCAI-curated disease maps, along with the CBM Knowledge Graph introduced above—were harmonized and integrated to ensure high coverage and biological reliability. The resulting graph comprises more than 1,400 nodes and 4,600 edges, structured as subject–predicate–object triples

(e.g., “SARS-CoV-2 infection increases neuroinflammation”, “IL6 activates TNF signaling”), and annotated with standardized ontologies such as MeSH, ChEBI, and Disease Ontology. Relationships are categorized by mechanistic predicates such as increase, decrease, association, `involved_in_pathway`, and positive correlation, representing diverse layers of molecular and phenotypic connectivity. Implemented in Neo4j for visualization and querying, the HKG integrates shared genetic risk factors (e.g., NLRP3, OAS1, TMPRSS2, CXCL10) and enriched biological pathways such as NF- κ B, nitric oxide, and chemokine signaling, collectively supporting the formulation of data-driven hypotheses on common mechanisms—particularly inflammation, immune dysregulation, and mitochondrial dysfunction—underlying complex disease comorbidities.

2.2 System Overview

CGEx (Cypher **G**enerating **E**xpert) is an interactive Retrieval-Augmented Generation (RAG) system that accepts a natural language question and produces both a Cypher query and a corresponding answer retrieved from COVID-NDD comorbidity KGs. Figure 1 illustrates the complete CGEx workflow. The user query is converted into a Cypher query using schema-aware prompting, executed on a domain-specific KG, and the results are passed back to the LLM to generate a detailed natural language explanation. The system follows the GraphRAG paradigm, implemented using LangChain components (e.g., LLM client, Neo4j connector, and prompt templates), and includes a user feedback loop to iteratively refine query accuracy and coverage.

2.3 Graph Connection and Schema Extraction

CGEx connects to Neo4j-hosted biomedical KGs using LangChain’s `Neo4jGraph` and the official Neo4j Python driver. To enable dynamic KG specific prompt generation,

the system first extracts the underlying KG schema-specifically node labels, relationship types, and associated properties-via Cypher-based introspection queries (Figure 1, prior to Step 2). Below are examples of such Cypher queries tailored to COMMUTE hypothesis KGs:

Listing 1 Query 1: Retrieve node labels and properties

```
CALL db.schema.nodeTypeProperties()

    YIELD nodeType, propertyName
    WITH nodeType,
        CASE
            WHEN nodeType CONTAINS 'ProteinModification' THEN
                COLLECT(CASE WHEN propertyName IN ['name', 'amino_acid', 'position'] THEN propertyName END)
            ELSE
                COLLECT(CASE WHEN propertyName = 'name' THEN
propertyName END)
        END AS Properties
    RETURN nodeType AS NodeLabel, Properties
    ORDER BY NodeLabel;
```

Listing 2 Query 2: Retrieve Relations and respective Properties

```
MATCH ()-[r]->()

    WITH DISTINCT type(r) AS relType, keys(r) AS properties
    UNWIND properties AS property
    WITH relType, COLLECT(DISTINCT property) AS uniqueProps
    WITH relType, [p IN uniqueProps WHERE p IN ["source", "citationType", "pmid", "citationRef", "evidence"]] AS filteredProps
    RETURN relType, filteredProps
    ORDER BY relType;
```

415 The extracted schema is formatted and injected into the LLM prompt as
416 a list of node types (for instance, ":'BioConcept':'BiologicalProcess'"; ":'BioCon-
417 cept':'Pathology'"; ":'ProteinModification'") and relationship types (for instance,
418 "ASSOCIATION"; "HAS_COMPONENT"; "REGULATES") alongside properties.
419
420 Populating the prompt with this context ensures the LLM generates Cypher based
421 on actual KG structure.
422
423
424

426 2.4 Prompt Engineering via Iterative Evolution

427
428 Throughout CGEx development, we adopted an iterative strategy to refine prompt
429 structure, guided by empirical evaluation across multiple prompt variants.
430
431

432
433 Initially, our base prompt used static instructions with minimal schema awareness.
434
435 However, to avoid hallucinated Cypher elements and mismatches with the underly-
436 ing KG, we systematically incorporated several key enhancements through prompt
437 variants:
438
439

- 440 • **Schema-aware Prompting:** Dynamic schema extraction (Listing 1) ensures that
441 the model aligns with the evolving structure of COMMUTE KGs, which are
442 continuously updated with new data sources.
443
- 444 • **Explicit Syntactical Instructions:** Enforce Cypher syntax constraints, apply
445 relationship directionality and multi-hop logic, flexible node matching heuristics
446 (e.g., case-insensitive substring filters). These rules were added to help the LLM
447 with the complicated Cypher syntax, and to avoid overly broad multi-hop traversals.
448
449
450
451
452

453 **Listing 3** PROMPT I: Syntactical instructions for the LLM to construct Cypher Queries

```
454 ### **Instructions for Query Generation**  
455 - **Refer to the Graph schema for correct node labels, relationship types,  
456 and property names.**  
457  
458  
459  
460
```

```

- **Use direct relationships (`MATCH (a)-[r]->(b)`) whenever possible.**
  Avoid arbitrary multi-hop traversal (`[*..3]`) unless indirect links are
  explicitly required.
- **Do NOT assume specific node labels unless explicitly required.** If
  unsure, match based on `name` rather than restricting labels (`MATCH (n)
  WHERE toLower(n.name) CONTAINS "covid"` instead of `MATCH (n:BioConcept
  ) WHERE n.name = "COVID")`.
- **Do NOT assume specific relationship types unless needed.** If uncertain,
  allow general connections (`MATCH (a)-[r]-(b)`) instead of forcing any
  relationship labels.
- Expand queries to consider multiple terminologies (e.g., "nervous system"
  = "neurological impact" = "neurodegeneration"). Instead of `WHERE
  toLower(n.name) CONTAINS "nervous system"`, use:
  WHERE (toLower(n.name) CONTAINS "nervous system" OR toLower(n.name)
  CONTAINS "neurological impact" OR toLower(n.name) CONTAINS "
  neurodegeneration")

```

- **Matching Rules:** We integrated detailed custom rules for node matching (e.g., using `toLower(name) CONTAINS "<keyword>"`) and relationship traversal. By enforcing case-insensitive substring filters and using logical OR for synonyms, the prompt could match domain-relevant entities even when exact labels differed.

Listing 4 PROMPT II: Meticulously organized instructions for identifying entities and relationships

```

### **Instructions for Query Generation**

**MANDATORY: USE THE GRAPH SCHEMA PROVIDED**

- You must refer to the graph schema when constructing Cypher queries.
- STRICTLY follow the node labels, relationship types, and property names as
  provided in the schemas.

```

```

507 - DO NOT invent node labels, properties, or relationships not present in the
508     schema.
509
510
511
512
513 **Node Matching Rules:**
514
515 - Do NOT specify any node labels, use name filtering:
516 For example, use `MATCH (n) WHERE toLower(n.name) CONTAINS "covid"`
517
518
519 - Do NOT apply conflicting conditions on the same entity. Use `OR` instead
520     of `AND` for multiple concepts.
521
522
523
524 - If question mentions "nervous system", "neurological impact", or "
525     neurodegeneration", so on:
526     then match nodes using `"neuro"`.
527
528
529
530
531 Relationship Matching Rules:
532
533 - First, check for a direct relationship (MATCH (a)-[r]-(b)).
534
535 - If no direct relationship exists, use multi-hop traversal (MATCH (a)
536     -[*..3]-(b)).
537
538
539 - Do NOT specify any relationship types, allow general connections:
540 Use `MATCH (a)-[r]-(b)` instead of forcing any relationship labels.
541
542 - DO NOT filter relationships using r.name in WHERE clauses. Do NOT specify
543     r.name.
544
545
546
547 - EVERY query MUST include LIMIT 5 to prevent excessive output!
548 In Cypher, use `LIMIT 5` with `RETURN`
549
550
551
552

```

- CoT-Few-Shot Prompting: Ultimately, we included *step-by-step* directives (chain-of-thought (CoT)) with a curated few-shot example from the COVID-NDD domain, guiding the model to reason through entity identification and query strategy.

Listing 5 PROMPT III: Includes step-by-step thinking instructions with an example

```
## **EXAMPLES (Few-Shot Reasoning + Query)**

### Example 1:

**User Question**:

*How do cytokine levels correlate with neurodegenerative outcomes in COVID-19 patients?*

**Step-by-Step Reasoning**:

1. Identify key entities: "cytokine", "COVID", and "neurodegenerative".
2. For "cytokine", match nodes where the entity name includes "cytokine".
3. "COVID" can be matched using `toLower(d.name) CONTAINS "covid"`.
4. Neurodegenerative examples include Alzheimers, Parkinsons, or general neurodegeneration; use:
   toLower(d.name) CONTAINS "alzheimer" OR toLower(d.name) CONTAINS "parkinson"
   OR toLower(d.name)
   CONTAINS "neuro"
5. No specific relationship types are required. Use `[r]`.
6. Add `LIMIT 5`.

**Cypher Query**:

```cypher
MATCH (c)-[r]-(d)-[r2]-(e)
WHERE toLower(c.name) CONTAINS "cytokine" AND
 (toLower(d.name) CONTAINS "parkinson" OR
```

```

 toLower(d.name) CONTAINS "alzheimer" OR
 toLower(d.name) CONTAINS "neuro") AND
 toLower(e.name) CONTAINS "covid"
RETURN c, r, d, r2, e LIMIT 5

```

## 2.5 Dynamic Prompt Construction and Multi-KG Query

### Execution

The final prompt combines advances from Listings 4 and 5 and supports multi-KG switching with schema-specific prompt templates generated dynamically based on the selected KG. To ensure accurate Cypher generation aligned with the evolving KG structure, CGEx constructs the LLM prompt dynamically at runtime. At step 1 of Figure 1, the user submits a natural language question. In step 2, CGEx determines the selected COMMUTE KG, retrieves its schema (see Subsection 2.3), and passes this information into the prompt template along with relevant instructions and chain-of-thought (CoT) reasoning example.

## 2.6 Cypher Query Generation

In steps 2-3 of Figure 1, an LLM (OpenAI's GPT series) receives the dynamically assembled prompt-including KG elements, syntactic instructions (Listing 4), and CoT-based few-shot example (Listing 5). Using the factual context embedded in the prompt template, the LLM transforms the user's submitted question into a Cypher query executable on the selected KG.

## 2.7 Cypher Execution and KG Retrieval

The generated Cypher query is sent to Neo4j backend for execution on the target KG, and the subgraphs are returned as raw results (steps 4 and 5 in Figure 1).

## 2.8 Detailed Response Generation

Each query’s Cypher output is executed on the selected Neo4j-hosted KG, and the raw results are fed into the LLM again for natural language generation (steps 6 and 7 in Figure 1) using a dedicated response prompt.

**Listing 6** Detailed Response Generation Prompt

```
response_prompt =
You are a medical expert in COVID-19 and NDD (Neurodegenerative Diseases)
knowledge.
Make sure you give complete response. It can be concise but should not be
incomplete.
Ensure that your response ends with a complete thought and does not stop
abruptly.
Based on the following knowledge graph results, provide a detailed and
structured response:

{json.dumps(kg_results, indent=2)}

Detailed Response:
```

This creates an interpretable explanation grounded in the KG results.

## 2.9 User Feedback and Real-time Example Updating

To enhance system performance and learning, CGEx incorporates a feedback mechanism (last step in Figure 1) that allows users to either approve or disapprove the generated Cypher queries and their corresponding natural language responses. In cases of disapproval, CGEx enters a few-shot mode, re-attempting Cypher query generation by referencing a curated example store containing previously validated (question, Cypher) pairs. This example store is dynamically updated-user approved cases are

691 appended in real-time and used to guide future translations. Through this itera-  
692 tive process, CGEx continuously refines its output by leveraging user feedback and  
693 contextually relevant few-shot examples tailored to biomedical query patterns.  
694

695

## 697 2.10 Implementation Details

698

699 The entire pipeline is deployed as a web application using Dash and Dash Bootstrap  
700 Components. It is hosted locally and offers secure access to both the OpenAI API and  
701 two Neo4j-based biomedical graph instances. All credentials, including API keys and  
702 KG access parameters, are managed securely through a `.env` configuration file. The  
703 frontend comprises:  
704

705

- 706 • A text input box for entering natural language questions.
- 707 • A drop-down menu to select the target KG.
- 708 • A submit button to initiate query generation.
- 709 • An explainability section that displays the dynamically constructed LLM prompt.  
710 As part of the explainable AI (XAI) design, CGEx displays the full prompt sent  
711 to the LLM on the web interface, ensuring user transparency, interpretability, and  
712 ethical accountability in the generation process.
- 713 • Output panes that show the generated Cypher query, KG results, and an LLM-  
714 generated detailed explanation.
- 715 • An interactive solution graph view where users can select individual edges to inspect  
716 associated evidence statements retrieved from the KG, providing fine-grained  
717 explainability beyond structural visualization.
- 718 • Crucially, the UI includes feedback buttons ("Approve" and "Disapprove") for each  
719 Cypher output. Upon approval, the question-query pair is added to a persistent  
720 example store. If disapproved, CGEx automatically re-runs the query in few-shot  
721 mode, retrieving relevant examples from the store to guide a second attempt,  
722 enabling real-time learning and adaptive performance tuning.

723

## 2.11 CGEx2.0

The upgraded version, **CGEx2.0**, introduces two major enhancements while preserving the overall system architecture. First, the LLM backend was upgraded from earlier GPT models to **GPT-5**. The prompting strategy remains unchanged, schema-aware instructions combined with chain-of-thought (CoT) reasoning and domain-specific few-shot example (Listings 4 and 5). The feedback loop and example store are also retained. To improve the reliability of generated natural language answers, minor guardrails were added in the response prompt, such as quoting relation types verbatim, avoiding invented directionality, and enforcing complete responses.

Second, **CGEx2.0** integrates solution graph visualization into the Dash interface using Cytoscape. The retrieved subgraph from Neo4j is normalized into nodes and edges, styled by label families, and rendered alongside the textual explanation. In addition to structural visualization, **CGEx2.0** exposes evidence annotations associated with selected edges in the solution subgraph. For each retrieved relationship, supporting evidence properties available in the knowledge graph (e.g., evidence statements, provenance, or citation identifiers) are extracted and displayed upon edge selection in the interface. This enables users to directly inspect the textual or bibliographic justification underlying specific mechanistic links, strengthening transparency and trust in the retrieved pathways.

## 3 Results

We evaluated CGEx on a manually curated benchmark of 52 biomedical questions designed to probe complex, multi-hop reasoning patterns across two COMMUTE comorbidity KGs. Both the original prototype and the upgraded **CGEx2.0** (GPT-5 with solution graph visualization) were tested on this set.

### 783 3.0.1 Prompt Evolution

784  
785 The system was tested iteratively across several prompt variants during the devel-  
786  
787 opment phase, culminating in a final prompt that significantly reduced query  
788 hallucinations and improved generation fidelity (Listing 7).  
789

790  
791 **Listing 7** Finalized Prompt: Improved overall quality of generated Cypher queries

```
792 You are an expert at translating user questions into Cypher Queries.
793
794 These Cypher queries are then run on a knowledge graph about COVID-19 and NDDs
795 (Neurodegenerative diseases).
796
797
798 Your task is to **think step-by-step** before generating the Cypher query.
799
800
801 ---
802
803
804 ## **Graph Schema**
805
806 - Nodes:
807 {nodes_schema_str}
808
809
810 - Relationships:
811 {relationships_schema_str}
812
813
814 ---
815
816
817
818 ## **Instructions for Query Generation**
819
820
821 ### **MANDATORY: USE THE GRAPH SCHEMA PROVIDED**
822
823 - You must refer to the graph schema when constructing Cypher queries.
824
825 - STRICTLY follow the node labels, relationship types, and property names as
826 provided in the schemas.
```

827  
828

- DO NOT invent node labels, properties, or relationships not present in the	829
schema.	830
	831
	832
---	833
	834
	835
	836
## **Node Matching Rules**	837
- Do NOT specify any node labels, use name filtering:	838
MATCH (n) WHERE toLower(n.name) CONTAINS "covid"	839
	840
	841
	842
- Do NOT apply conflicting conditions on the same entity. Use `OR` instead of	843
`AND` for multiple concepts.	844
	845
- If the question mentions "neurological impact", "nervous system", "	846
neurodegeneration", etc.,	847
match nodes using `"neuro"`.	848
	849
	850
---	851
	852
	853
	854
## **Relationship Matching Rules**	855
- First check for a direct relationship (`MATCH (a)-[r]-(b)`).	856
	857
- If no direct relationship exists, use multi-hop traversal (`MATCH (a)	858
-[*..3]-(b)`).	859
	860
- Do NOT specify any relationship types. Allow general connections.	861
	862
- DO NOT filter relationships using `r.name`.	863
	864
	865
---	866
	867
	868
## **ALWAYS Include `LIMIT 5` in RETURN.**	869
	870
	871
---	872
	873
	874

```

875 ## **EXAMPLES (Few-Shot Reasoning + Query)**
876
877
878 ### Example 1:
879
880 **User Question**:
881
882 *How do cytokine levels correlate with neurodegenerative outcomes in COVID-19
883 patients?*
884
885
886 **Step-by-Step Reasoning**:
887
888 1. Identify key entities: "cytokine", "COVID", and "neurodegenerative".
889
890 2. For "cytokine", match nodes where the entity name includes "cytokine".
891
892 3. "COVID" can be matched using `toLower(d.name) CONTAINS "covid"`.
893
894 4. Neurodegenerative examples include Alzheimers, Parkinsons, or general
895 neurodegeneration; use:
896 toLower(d.name) CONTAINS "alzheimer" OR toLower(d.name) CONTAINS "parkinson"
897 OR toLower(d.name)
898 CONTAINS "neuro"
899
900 5. No specific relationship types are required. Use `-[r]-`.
901
902 6. Add `LIMIT 5`.
903
904 **Cypher Query**:
905
906 ```cypher
907 MATCH (c)-[r]-(d)-[r2]-(e)
908 WHERE toLower(c.name) CONTAINS "cytokine" AND
909 (toLower(d.name) CONTAINS "parkinson" OR
910 toLower(d.name) CONTAINS "alzheimer" OR
911 toLower(d.name) CONTAINS "neuro") AND
912 toLower(e.name) CONTAINS "covid"
913
914 RETURN c, r, d, r2, e LIMIT 5
915
916 ```
917
918
919
920

```

```
Do NOT add extra text like "Example:", "Cypher:", "---", etc., to the actual
 Cypher query.
```

```
Now generate a Cypher query for this question:
{{question}}
```

Certain complex questions, such as “*How do ACE2 and TMPRSS2 regulate the effects of COVID-19 on the nervous system?*” and “*How does the presence of dementia influence the severity of COVID-19 outcomes?*” resulted in unstable outputs when using the prompt without chain-of-thought (CoT) guidance (see Listing 3). The system frequently failed or fluctuated across multiple feedback loops, generating inconsistent or invalid Cypher queries. Without CoT’s step-by-step reasoning and our operator hints (case-insensitive CONTAINS, alias-aware matching, and OR disjunctions), the model defaulted to brittle exact matches and over-constrained conjunctions, making it highly sensitive to small label or casing differences and prone to empty or invalid Cypher. In contrast, the CoT-enabled few-shot prompt (see Listing 7) first decomposed the question into intermediate reasoning steps before query generation, applying flexible matching rules that improved recall without sacrificing precision. This combination consistently yielded valid queries within a stable feedback loop (typically resolving within one or two iterations), highlighting the importance of both prompt structure and reasoning guidance in stabilizing model performance on semantically dense biomedical queries.

### 3.0.2 CGEx Accuracy in Cypher Generation

To evaluate system performance, we extended testing to a benchmark of 52 curated biomedical questions across two COMMUTE comorbidity KGs. Both the original

CGEx prototype and the upgraded **CGEx2.0** (GPT-5 with solution graph visualization) were assessed, yielding four evaluation conditions in total.

**Evaluation criteria:** Accuracy was defined as the syntactic and semantic correctness of the generated Cypher query. A Cypher query was considered correct if it was successfully generated within **three or fewer feedback loops**. Queries exceeding this threshold or failing outright were counted as incorrect.

The comparative accuracy results are summarized in Figure 2. The accuracies for each condition were as follows:

- KG-I (original CGEx): **82.7%**
- KG-II (original CGEx): **55.8%**
- KG-I (CGEx2.0, GPT-5): **94.2%**
- KG-II (CGEx2.0, GPT-5): **82.7%**

The upgraded GPT-5 version clearly outperformed the earlier prototype on both KGs, achieving a particularly strong gain on KG-II (55.8% → 82.7%). Importantly, these improvements were achieved while maintaining the same prompting strategy, highlighting the robustness of the design and the benefits of a stronger LLM backend. Detailed per-question results for all 52 test cases, including the number of feedback loops per run, are provided in the Supplementary Material.

### 3.0.3 Edge Cases Analysis

To further evaluate CGEx’s robustness, we conducted an edge case analysis targeting questions that the earlier version of the system struggled to resolve. These challenges typically stemmed from schema limitations, semantic ambiguity, missing or inconsistently represented entities, and relationship directionality not captured by the schema. To illustrate these recurring error categories, we showcase a selection of

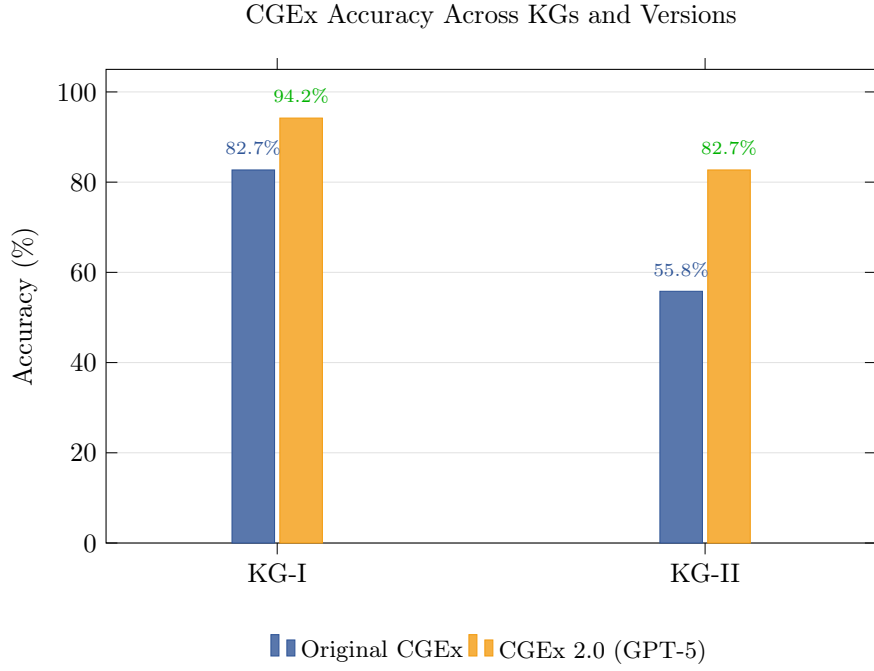


**Fig. 2** Accuracy (success within  $\leq 3$  feedback loops) on 52 questions across two COMMUTE KGs.

representative cases. We then re-tested the same cases using *CGEx2.0*, which successfully resolved several of the previously unsolved edge cases, particularly handling ambiguous entity representations and multi-hop relationships, while a subset of issues (e.g., incomplete KG coverage) persisted.

### *Representative Edge Cases on COMMUTE KG-I*

#### **Edge Case I**

How does COVID-19 affect sensory perception, and is this linked to neurodegenerative diseases?

**Listing 8** Cypher by CGEx (original)

```

1059 MATCH (c)-[r]-(d)-[r2]-(e) WHERE toLower(c.name) CONTAINS "neuro" AND toLower(
1060 e.name) CONTAINS "covid" AND (toLower(d.name) CONTAINS "sensory perception
1061 " OR toLower(d.name) CONTAINS "nervous system") RETURN c, r, d, r2, e
1062
1063 LIMIT 5
1064

```

1066 **Identified Issue:** When more than three nodes are involved, the model may face  
1067 confusions with ordering and connectivity, leading to unstable traversal.

1072 **Listing 9** Cypher by Expert

```

1073 MATCH (c)-[r]-(e)-[r2]-(d) WHERE toLower(c.name) CONTAINS "neuro" AND toLower(
1074 e.name) CONTAINS "covid" AND (toLower(d.name) CONTAINS "sensory perception
1075 " OR toLower(d.name) CONTAINS "nervous system") RETURN c, r, d, r2, e
1076
1077 LIMIT 5
1078

```

1081 **Listing 10** Cypher by CGEx2.0

```

1082 MATCH path = (c)-[*..3]-(s)-[*..3]-(n) WHERE toLower(c.name) CONTAINS "covid"
1083
1084 AND (toLower(s.name) CONTAINS "sensory" OR toLower(s.name) CONTAINS "
1085 perception" OR toLower(s.name) CONTAINS "olfactory" OR toLower(s.name)
1086 CONTAINS "smell" OR toLower(s.name) CONTAINS "anosmia" OR toLower(s.name)
1087 CONTAINS "taste" OR toLower(s.name) CONTAINS "ageusia" OR toLower(s.name)
1088 CONTAINS "hearing" OR toLower(s.name) CONTAINS "vision") AND (toLower(n.
1089 name) CONTAINS "alzheimer" OR toLower(n.name) CONTAINS "parkinson" OR
1090 toLower(n.name) CONTAINS "neuro" OR toLower(n.name) CONTAINS "dementia")
1091
1092 RETURN path LIMIT 5
1093

```

1097 **Improvement (CGEx2.0):** The GPT-5 version generated a broader, more  
1098 semantically aligned query, expanding synonyms for chemosensory dysfunction (e.g.,  
1099 *anosmia, ageusia, olfactory, smell, taste, dysgeusia, hyposmia*) in line with expert  
1100 expectations. It also integrated interpretation across the knowledge graph (COVID-19  
1101 ↔ chemosensory dysfunction ↔ ACE2/TMPRSS2 ↔ dementia), thereby resolving

the earlier connectivity issue. The resulting solution subgraph is shown in Figure 3.

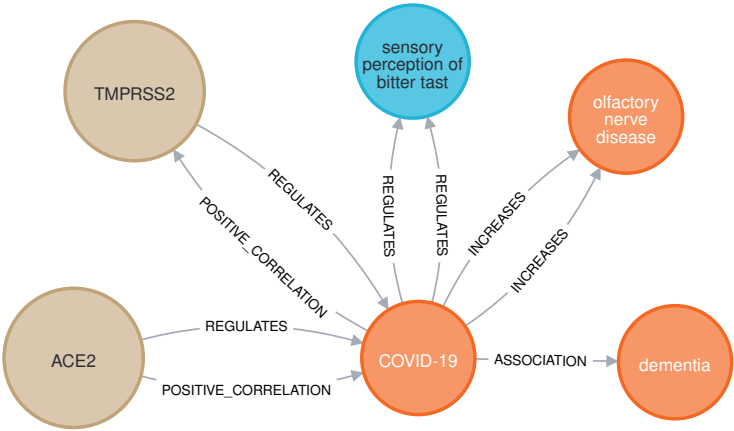


**Fig. 3** Subgraph generated from CGEx2.0 query, demonstrating successful resolution of Edge Case I

As shown in Figure 4, selecting the “Association” edge between COVID-19 and dementia displays the supporting evidence and provenance.

Selected Edge Evidence:

Evidence:  
Coexisting conditions included diabetes mellitus (n=4), hypertension (n=21), cardiovascular disease (n=9), hyperlipidemia (n=2), chronic kidney diseasechronic kidney disease (n=2), prior stroke (n=6) and dementia (n=5) (Supplementary Table 1).

PMID: 33257876  
Source: PubMed

**Fig. 4** Example of the evidence panel shown when a user selects an edge in the solution subgraph, presenting the supporting textual evidence, associated coexisting conditions with counts, and bibliographic provenance (PubMed source and PMID), thereby enabling transparent inspection of the knowledge graph relationships.

**Edge Case II**

Can ACE2-targeted therapies mitigate the neurological impacts of COVID-19?

1151 **Listing 11** Cypher by CGEx (original)

```
1152 MATCH (c)-[r]-(d)-[r2]-(e) WHERE toLower(c.name) CONTAINS "ACE2-targeted
1153 therapies" AND (toLower(d.name) CONTAINS "neuro" OR toLower(d.name)
1154 CONTAINS "nervous system") AND toLower(e.name) CONTAINS "covid" RETURN c,
1155 r, d, r2, e LIMIT 5
```

1159  
1160 **Identified Issue:** Model misinterprets "ACE2-targeted therapies" to be present  
1161 in KG and spirals into multiple hallucinated hops. Queries failed when terms such as  
1162 therapy or treatment did not directly map to KG nodes, leading to missed connec-  
1163 tions to inhibitors or drug entities.

1166

1167

1168

1169 **Listing 12** Cypher by Expert

```
1170 MATCH path = (a)-[*..2]-(c)-[r3]-(d)
1171 WHERE a.name CONTAINS "Alzheimer"
1172 AND c.name = "ACE2"
1173 AND d.name = "COVID-19"
1174 RETURN path;
```

1177

1178

1179 **Listing 13** Cypher by CGEx2.0

```
1180 MATCH path = (t)-[*..3]-(a)-[*..3]-(c)-[*..3]-(n) WHERE toLower(a.name)
1181 CONTAINS "ace2" AND toLower(c.name) CONTAINS "covid" AND (toLower(n.name)
1182 CONTAINS "neuro" OR toLower(n.name) CONTAINS "alzheimer" OR toLower(n.name)
1183 CONTAINS "parkinson" OR toLower(n.name) CONTAINS "dementia") AND (
1184 toLower(t.name) CONTAINS "therap" OR toLower(t.name) CONTAINS "treatment"
1185 OR toLower(t.name) CONTAINS "inhibitor" OR toLower(t.name) CONTAINS "
1186 antagonist" OR toLower(t.name) CONTAINS "blocker" OR toLower(t.name)
1187 CONTAINS "antibody") RETURN path LIMIT 5
```

1193

1194 **Improvement (CGEx2.0):** The upgraded CGEx2.0 introduced synonym  
1195 expansion (*therapy, treatment, inhibitor, antagonist, antibody, blocker*), which aligned  
1196

with how therapeutic agents were represented in the KG (Figure 5). This resolved the earlier mismatch and retrieved clinically meaningful chains, including:

- Histone deacetylase inhibitors (HDAC inhibitors) → increase regulatory T cells (Tregs) → decrease inflammatory response.
- HDAC inhibitors → increase ACE2 expression.
- SARS-CoV-2 uses ACE2 to enter cells → drives COVID-19 pathogenesis.
- COVID-19 → sensory perception of bitter taste.
- COVID-19 → increased neurological signs and symptoms; increased neurocognitive disorders; association with dementia and Parkinson's disease; decreased neuromelanin.

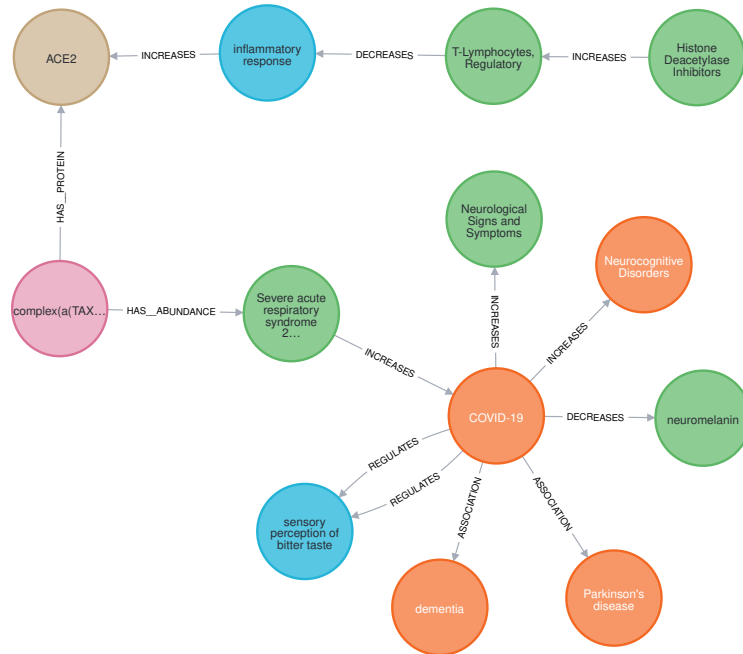


**Fig. 5** Solution subgraph generated by CGEx2.0, correctly resolving Edge Case II by mapping therapeutic terms to inhibitors represented in the KG.

### 1243 Edge Case III

1244 Is the phosphorylation state of MAPT (e.g., pmod(Ph)) a critical factor in the dis-  
1245 ruption of neuronal function?  
1246

1247

1248

1249

1250 **Listing 14** Cypher by CGEx (original)

```
1251 MATCH path = (p:ProteinModification)-[r]-(g:Gene)-[r2]-(p2:Protein) WHERE
1252 toLower(p.name) CONTAINS "pmod" AND toLower(p.name) CONTAINS "ph" AND
1253 toLower(g.name) CONTAINS "mapt" AND toLower(p2.name) CONTAINS "protein"
1254
1255 RETURN path LIMIT 5
```

1256

1257  
1258  
1259 **Identified Issue:** This case illustrates a schema mismatch: functional features  
1260 such as phosphorylation are encoded within BEL (e.g., pmod annotations) rather  
1261 than as standard node properties or explicit relationships. As a result, the earlier  
1262 CGEx versions generated incomplete queries that failed to retrieve phosphorylation-  
1263 specific modifications of MAPT.  
1264

1265

1266

1267

1268 **Listing 15** Cypher by Expert

```
1269 match (n)-[r]-(m) where n.name contains "MAPT" OR n.bel contains "pmod(Ph" AND
1270 m.name contains "neuron" return n,r,m
```

1271

1272

1273 **Listing 16** Cypher by CGEx2.0

```
1274 MATCH path = (m)-[*..3]-(pm)-[*..3]-(n) WHERE toLower(m.name) CONTAINS "mapt"
1275
1276 AND (toLower(pm.name) CONTAINS "pmod" OR toLower(pm.name) CONTAINS "
1277 phospho" OR toLower(pm.name) CONTAINS "phosphoryl") AND (toLower(n.name)
1278 CONTAINS "neuro" OR toLower(n.name) CONTAINS "neuron" OR toLower(n.name)
1279 CONTAINS "neuronal" OR toLower(n.name) CONTAINS "dysfunction" OR toLower(n
1280 .name) CONTAINS "disrupt") RETURN path LIMIT 5
```

1281

1282

1283

1284

**Improvement (CGEx2.0):** The GPT-5 upgrade resolved the mismatch by expanding synonyms (*pmod*, *phospho*, *phosphoryl*) and incorporating cases for phosphorylated states. This enabled the retrieval of phosphorylation-specific MAPT variants and their downstream associations. The resulting subgraph (Figure 6) captured key causal links:

- MAPT (tau) is positively correlated with Parkinson’s disease (PD).
- PD is linked to increased cytokine production, driving inflammatory responses and neuroinflammation.
- The graph encodes that PD is connected to oxidative phosphorylation (OxPhos).
- Tau states are explicitly modeled, including MAPT variants and phosphorylated forms, tying post-translationally modified tau to the PD-inflammation-mitochondrial axis.

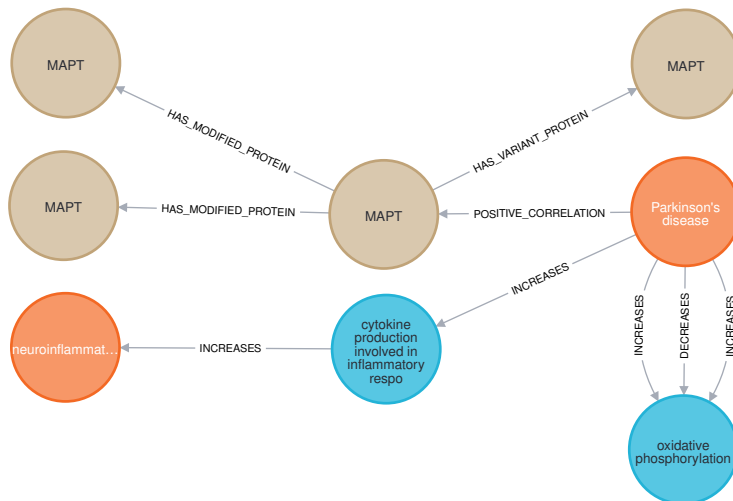


**Fig. 6** Subgraph generated by CGEx2.0 for Edge Case III, correctly resolving BEL-encoded phosphorylation features of MAPT

*Representative Edge Cases on COMMUTE KG-II*

1335 **Edge Case I**

1336  
1337 How do ACE2 and TMPRSS2 regulate the effects of COVID-19 on the nervous  
1338 system?

1339  
1340  
1341

1342 **Listing 17** Cypher by CGEx (original)

```
1343 MATCH (a)-[r]-(b)-[r2]-(c)-[r3]-(d) WHERE toLower(a.name) CONTAINS "ace2" AND
1344 toLower(b.name) CONTAINS "tmprss2" AND (toLower(c.name) CONTAINS "neuro"
1345 OR toLower(c.name) CONTAINS "nervous") AND toLower(d.name) CONTAINS "covid"
1346 " RETURN a, r, b, r2, c, r3, d LIMIT 5
```

1350

1351 **Identified Issue:** The model incorrectly assumes ACE2 and TMPRSS2 are  
1352 separate node labels and attempts to chain them via an edge, i.e., (a)-[r]-(b) where  
1353 a.name CONTAINS 'ace2' and b.name CONTAINS 'tmprss2', instead of treating  
1354 them as alternative string matches in a single node using a.name CONTAINS 'ace2'  
1355 OR 'tmprss2'. This leads to an extra hop and incorrect node chaining.

1358  
1359  
1360

1361 **Listing 18** Cypher by Expert

```
1362 MATCH (a)-[r]-(b)-[r2]-(c) WHERE (toLower(a.name) CONTAINS "ace2" OR toLower(a
1363 .name) CONTAINS "tmprss2") AND toLower(b.name) CONTAINS "covid" AND (
1364 toLower(c.name) CONTAINS "neuro" OR toLower(c.name) CONTAINS "nervous")
1365 RETURN a, r, b, r2, c
```

1369

1370 **Listing 19** Cypher by CGEx2.0

```
1371 MATCH path = (g)-[*..3]-(c)-[*..3]-(n)
1372 WHERE (toLower(g.name) CONTAINS "ace2"
1373 OR toLower(g.name) CONTAINS "tmprss2")
1374 AND toLower(c.name) CONTAINS "covid"
1375 AND toLower(n.name) CONTAINS "neuro"
1376 RETURN path LIMIT 5
```

1380

**Improvement (CGEx2.0):** The upgraded query resolves the original chaining error by grouping ACE2 and TMPRSS2 as alternative string matches within a single node variable rather than forcing an edge between them. It also introduces variable-length traversals ([\*..3]) between gene-COVID and COVID-neuro nodes, allowing the model to capture both direct and indirect regulatory paths without relying on rigid schema constraints or edge directionality. This more flexible and semantically aligned structure retrieves the intended subgraph, correctly linking ACE2/TMPRSS2 to COVID-19 and neuro-related nodes, as shown in Figure 7.

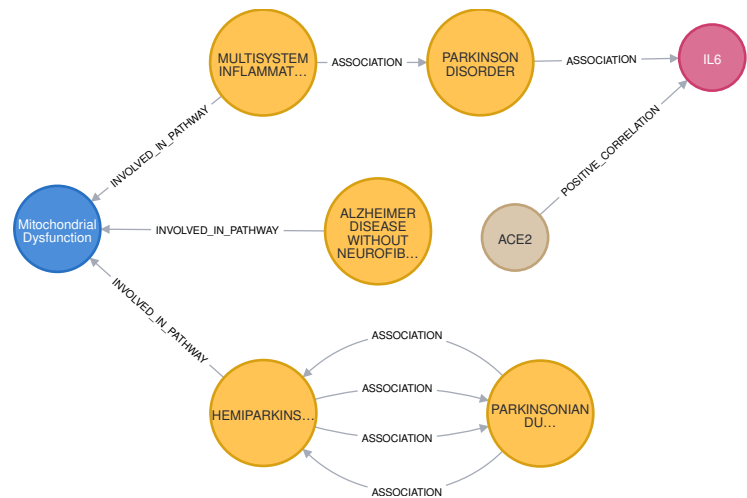


**Fig. 7** Solution subgraph generated by CGEx2.0 for the ACE2/TMPRSS2-COVID-neuro query. The upgraded query retrieves multiple variable-length paths, uncovering intermediary inflammatory (e.g., IL6) and mitochondrial processes that bridge ACE2 to downstream neurological disease nodes. This richer multi-hop structure reflects the system’s improved ability to traverse indirect mechanistic links and capture higher-order biological context beyond simple pairwise associations.

## Edge Case II

What are the impacts of COVID-19 on psychosis?

**Listing 20** Cypher by CGEx (original)

```

1427 MATCH (d)-[r]-(p)-[r2]-(c) WHERE toLower(d.name) CONTAINS "covid" AND (toLower
1428 (p.name) CONTAINS "psychosis" OR toLower(p.name) CONTAINS "neuro") AND
1429 toLower(c.name) CONTAINS "covid" RETURN d, r, p, r2, c LIMIT 5
1430
1431

```

1432  
1433 **Identified Issue:** The query redundantly anchors COVID at both ends ( d ...  
1434 c both filtered to “covid”), forcing a fixed two-hop motif COVID-psychosis/neuro-  
1435 COVID. This brittleness

- 1437 1. dilutes specificity by mixing psychosis with the generic neuro term, and
- 1438 2. couples the answer to a rigid middle node pattern rather than allowing alternate
- 1439 mediators.

1440  
1441  
1442  
1443  
1444 **Listing 21** Cypher by Expert

```

1445
1446 MATCH p = (cov)-[*..3]-(psy)
1447 WHERE toLower(cov.name) CONTAINS "covid"
1448 AND (toLower(psy.name) CONTAINS "psychosis" OR toLower(psy.name) CONTAINS "
1449 psychotic")
1450 RETURN p LIMIT 5
1451
1452
1453

```

1454  
1455 **Listing 22** Cypher by CGEx2.0

```

1456
1457 MATCH p = (a)-[*1..1]-(b)
1458 WHERE toLower(a.name) CONTAINS "covid" AND toLower(b.name) CONTAINS "psychosis
1459 "
1460 RETURN p
1461
1462 UNION
1463 MATCH p = (a)-[*2..3]-(b)
1464 WHERE toLower(a.name) CONTAINS "covid" AND toLower(b.name) CONTAINS "psychosis
1465 "
1466 RETURN p
1467 LIMIT 5
1468
1469
1470
1471

```

**Improvement (CGEx2.0):** CGEx2.0 decomposes the task into direct ([\*1..1]) and indirect ([\*2..3]) searches and combines them using UNION, providing a systematic fallback from single-edge to multi-hop evidence. The undirected patterns keep the query directionality-flexible, while bounded hops control path length and reduce noise. The resulting solution subgraph and its converging mechanistic pathways are shown in Figure 8.

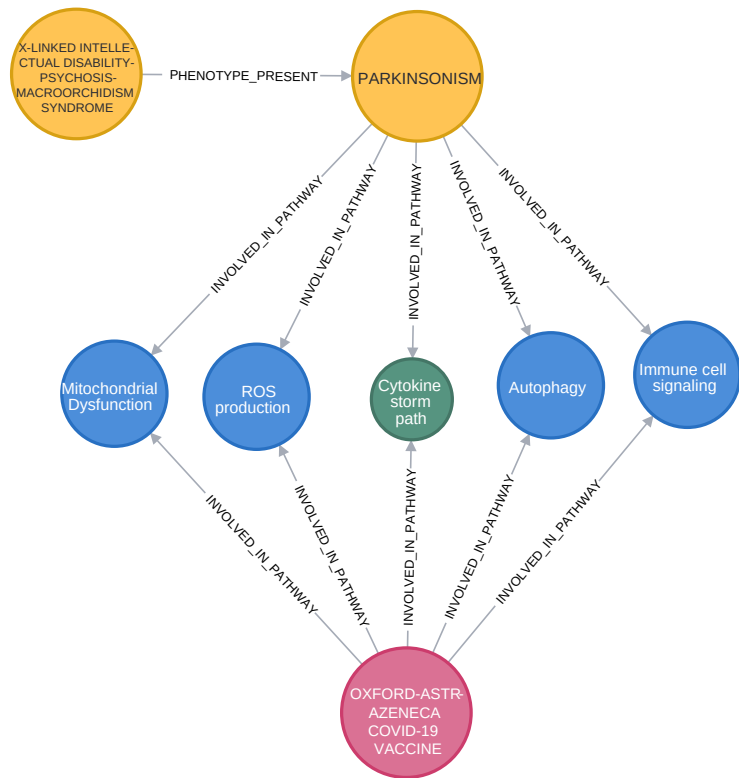


**Fig. 8** Solution subgraph generated by CGEx2.0 for the COVID-psychosis query. The graph reveals multiple converging and branching mechanistic pathways linking the disease to psychosis-related nodes, spanning immune, mitochondrial, oxidative stress, and autophagy processes, as well as neurodegenerative intermediates such as parkinsonism.

### 1519 3.0.4 Unresolved Edge Cases: KG Gaps

1520

1521 Some queries remain unsolved in CGEx2.0 not due to model shortcomings, but  
1522 because of incomplete or missing information in the knowledge graphs. Two repre-  
1523 sentative cases illustrate this:

1526

1527

1528 (A) What are the neurological symptoms associated with long COVID, and how  
1529 do they relate to conditions like stroke or Alzheimer's?

1531

1532

1533 **Listing 23** Cypher by CGEx

1534

```
1535 MATCH (n)-[r]-(d)-[r2]-(e) WHERE toLower(n.name) CONTAINS "neuro" AND (
1536 toLower(d.name) CONTAINS "stroke" OR toLower(d.name) CONTAINS "alzheimer")
1537 AND toLower(e.name) CONTAINS "covid" RETURN n, r, d, r2, e LIMIT 5
```

1539

1540

1541 **Identified Issue:** The question asks for neurological aspects associated with long  
1542 COVID, via stroke or Alzheimer's. However, the KG has no "neurological aspect"  
1543 connected to stroke or AD. The KG consists of only relations between COVID and  
1544 stroke or Alzheimer's disease.

1547

1548

1549 (B) Does hypertension increase the susceptibility of COVID-19 patients to neu-  
1550 rodegenerative diseases?

1552

1553

1554

1555 **Listing 24** Cypher by CGEx

1556

```
1556 MATCH (a)-[r]-(b)-[r2]-(c) WHERE toLower(a.name) CONTAINS "hypertension" AND
1557 toLower(b.name) CONTAINS "covid" AND (toLower(c.name) CONTAINS "neuro" OR
1558 toLower(c.name) CONTAINS "alzheimer" OR toLower(c.name) CONTAINS "
1559 parkinson") RETURN a, r, b, r2, c LIMIT 5
```

1562

1563

1564

**Identified Issue:** COMMUTE KG-II does not consist of explicit "hypertension" nodes.

## 4 Discussion

Prompt design played a pivotal role in CGEx’s overall performance. We observed that adding chain-of-thought (CoT) reasoning to the prompt, especially in combination with few-shot examples substantially improved the quality and stability of generated Cypher queries. CoT reasoning introduces structured thinking that aligns the LLM’s output with logical execution paths, which is crucial for multi-hop query generation. Few-shot examples act as cognitive anchors, guiding the model toward schema-conformant and semantically precise Cypher generation.

With the upgrade to GPT-5 (CGEx2.0), we observed a marked increase in both accuracy and robustness across COMMUTE KG-I and KG-II. The system overcame several challenging edge cases and achieved more stable outputs with fewer user feedback loops.

At the Cypher generation level, CGEx2.0 exhibited qualitatively more systematic reasoning strategies. The model generated Cypher queries combining direct and indirect paths (**UNION**), introducing **fallbacks** to multi-hop edges. It also fostered direction-flexible **undirected patterns** to tolerate edge orientation differences, expanded lexical coverage through **synonym diversification** (e.g., chemosensory terms, therapeutic categories), and leveraged **bounded variable-length traversals** to reveal indirect mechanisms while minimizing spurious connections. With version 2.0, generated queries were directly rendered as solution graphs, offering visual interpretation for non-Cypher users.

1611 It is crucial to acknowledge that LLM outputs are inherently probabilistic  
1612 rather than deterministic. Minor fluctuations across runs can occur, particularly on  
1613 semantically ambiguous questions. This underlines the value of human-in-the-loop  
1614 verification during deployment, especially for biomedical use cases where repro-  
1615 ducibility is paramount.

1619  
1620  
1621 Notably, the natural language responses generated by CGEx2.0 (see Supplemen-  
1622 tary Material) were elaborate, blending evidence retrieved from the KG with the  
1623 model’s broader parametric knowledge drawn from its training corpus. Thus, the  
1624 responses often contextualized KG-derived mechanisms with supporting information  
1625 from literature and clinical understanding, producing well-rounded answers that were  
1626 both mechanistically grounded and narratively informative. This hybrid reasoning  
1627 allowed the system to provide meaningful insights even when the KG partially cov-  
1628 ered or completely lacked information on the queried topic. The system additionally  
1629 identified the likely cause of failure, such as missing nodes, or absent intermediate  
1630 relations, effectively surfacing gaps in the KG. This diagnostic behavior is critical for  
1631 downstream KG refinement, as it highlights where additional entities, synonyms, or  
1632 schema alignment are needed (see Supplementary Material).

1641  
1642  
1643 Some of the failures arose not from the system itself but from **incomplete KG**  
1644 **coverage**. Interestingly, the two KGs tended to fail on different questions, indicating  
1645 a complementary distribution of knowledge. This behavior highlights a key insight:  
1646 rather than attempting costly schema harmonization, a more pragmatic approach  
1647 is to harness **multi-KG complementarity** (as shown in Supplementary Material)  
1648 through a unified Cypher-generating agent. In practice, CGEx2.0 frequently suc-  
1649 ceeded by retrieving answers from one KG when the other lacked relevant nodes or  
1650 relationships, demonstrating the viability of parallel, complementary querying as an  
1651  
1652  
1653  
1654  
1655  
1656

alternative to full harmonization.

Even domain experts with full access to the KG structure required 5-10 minutes to formulate correct Cypher queries for several edge cases. In contrast, participants unfamiliar with the KG schema often produced queries that closely resembled CGEx’s outputs, underscoring the model’s ability to approximate plausible paths even under cryptic or under-specified inputs. This not only highlights CGEx’s practical utility as a time-saving assistant, but also reinforces its value in guiding non-expert users towards syntactically as well as semantically relevant query formulations. By removing the need for SQL or Cypher expertise, CGEx lowers the technical barrier for interacting with hypothesis-driven KGs and supports a wider community of users. Natural-language interaction enables versatile users to explore hypotheses, pathways, guidelines, and mechanistic evidence in a transparent manner, making structured evidence bases more accessible and interpretable.

## 5 Limitations

While CGEx2.0 represents a significant advance in Cypher generation and NL interpretation, several limitations remain:

- **Schema blind spot:** Certain schema blind spots persist with BEL-encoded information. COMMUTE KGs store some details, like post-translational modifications using BEL annotations embedded within node properties, rather than as explicit graph structures. Since GPT-based models have very limited exposure to BEL syntax, CGEx does not attempt to parse or translate BEL expressions directly. A BEL-to-graph conversion step is needed upstream to expose this information in the KG, enabling Cypher-based retrieval.
- **Knowledge Graph Coverage:** CGEx2.0 remains constrained by the coverage and granularity of the underlying KGs. Missing entities and relationships lead to

1703 predictable retrieval failures. For example, the absence of intermediate nodes (e.g.,  
1704 *neurological symptoms* linking long COVID and stroke or Alzheimer’s) limits the  
1705 system’s ability to resolve multi-hop questions.  
1706

1707  
1708 • **Incorrect node chaining:** Although CGEx2.0 *generally maintains correct undi-*  
1709 *rected Cypher patterns*, it may occasionally hallucinate to misplace node connec-  
1710 tions, leading to spurious chaining. This can be mitigated by implementing a  
1711 post-generation validation step that checks traversal direction and connectivity  
1712 against the schema.  
1713

1714  
1715  
1716 • **Ontology grounding limitations:** Without integrated ontology-based reasoning,  
1717 CGEx may overlook medically valid connections that are not explicitly encoded in  
1718 the KG. For instance, while the graph may contain the entity “blood pressure,” it  
1719 lacks a corresponding “hypertension” node. As a result, queries related to hyperten-  
1720 sion return empty outputs despite being semantically meaningful. Although “BP”  
1721 (blood pressure) is clinically understood as a marker or symptom of hypertension,  
1722 this relationship is not represented in the KG, preventing the model from inferring  
1723 such links. Incorporating lightweight ontology reasoning or precomputed mappings  
1724 could help bridge these implicit relationships.  
1725  
1726  
1727  
1728  
1729  
1730

1731  
1732

## 1733 6 Conclusion

1734

1735 We presented CGEx2.0, an explainable NL-to-Cypher generation framework to query  
1736 heterogeneous biomedical KGs. It lowers the barrier for clinicians, biologists, and  
1737 researchers to ask natural language questions and receive Cypher queries and struc-  
1738 tured answers in return. As an end-to-end QA agent that combines KG retrieval with  
1739 LLM-based reasoning, the system delivers the best of both worlds: structured evi-  
1740 dence grounding (RAG) and rich explanatory context. In doing so, it transforms KGs  
1741 into knowledge discovery engines for biologists to uncover COVID-NDD links, support  
1742 hypothesis generation, pathway exploration, and potentially informing therapeutic  
1743  
1744  
1745  
1746  
1747  
1748

strategies. Furthermore, CGEx2.0 demonstrates that querying multiple heterogeneous KGs in parallel can be more practical than harmonizing them, offering a flexible and scalable strategy for uncovering COVID-NDD insights. Future work will focus on ontology grounding, and multi-modal evidence integration to support richer COVID-associated neurodegeneration reasoning.

## 7 Declarations

### 7.1 Ethics approval and consent to participate

Not applicable. This study did not involve human participants, human data, human tissue, or animals.

### 7.2 Consent for publication

Not applicable.

### 7.3 Availability of data and materials

The knowledge graphs used in this study are publicly available through the CBM Comorbidity Knowledge Graph repository [20] and the COVID-19 and neurodegenerative disease comorbidity repository [21]. The CGEx (Cypher Generating Expert) source code and documentation are publicly available through the CGEx repository [22].

### 7.4 Competing interests

The authors declare that they have no competing interests.

## 1795 **7.5 Funding**

1796

1797 This work was supported by the Fraunhofer Institute for Algorithms and Scientific  
1798 Computing (SCAI). Funding came from the COMMUTE project, which is funded by  
1799 the European Union under grant agreement number 101136957.  
1800

1801

## 1802 **7.6 Authors' contributions**

1803

1804 AA contributed to conceptualization, methodology, software development and imple-  
1805 mentation, formal analysis, system evaluation and validation, visualization, and  
1806 writing the original draft. NB prepared the expert Cypher queries used as references  
1807 for comparison. MHA contributed to funding acquisition. MJ supervised the work and  
1808 contributed to conceptualization and methodological development and improvement.  
1809 All authors contributed to reviewing and editing the manuscript. All authors read and  
1810 approved the final manuscript.  
1811

## 1812 **7.7 Acknowledgements**

1813

1814 Not applicable.

1815

## 1816 **8 Disclaimer**

1817

1818 The findings and recommendations in this paper are presented to the best of the  
1819 authors' knowledge and understanding. While the systems and methods have been  
1820 rigorously tested, there may be limitations or errors arising from incomplete datasets  
1821 or unforeseen scenarios.  
1822

1823

## 1824 **References**

1825

- 1826 [1] World Health Organization.: A clinical case definition of post COVID-  
1827 19 condition by a Delphi consensus. Accessed: 2025-07-08. Avail-  
1828 able from: [https://www.who.int/publications/i/item/WHO-2019-nCoV-Post\\_](https://www.who.int/publications/i/item/WHO-2019-nCoV-Post_)  
1829  
1830  
1831  
1832  
1833

COVID-19\_condition-Clinical\_case\_definition-2021.1. 1841  
1842

[2] Raveendran AV, Jayadevan R, Sashidharan S. Long COVID: An overview. Dia- 1843  
betes & Metabolic Syndrome: Clinical Research & Reviews. 2021;15(3):869–875. 1844  
<https://doi.org/10.1016/j.dsx.2021.04.007>. 1845  
1846  
1847  
1848

[3] Fu YW, Xu HS, Liu SJ. COVID-19 and neurodegenerative diseases. European 1849  
Review for Medical & Pharmacological Sciences. 2022;26(12):4535–4544. [https://doi.org/10.26355/eurrev\\_202206\\_29093](https://doi.org/10.26355/eurrev_202206_29093). 1850  
1851  
1852  
1853  
1854

[4] Davis HE, Assaf GS, McCorkell L, Wei H, Low RJ, Re'em Y, et al. Character- 1855  
izing long COVID in an international cohort: 7 months of symptoms and their 1856  
impact. EClinicalMedicine. 2021;38:101019. <https://doi.org/10.1016/j.eclinm.2021.101019>. 1857  
1858  
1859  
1860  
1861  
1862

[5] Koc HC, Xiao J, Liu W, Li Y, Chen G. Long COVID and its Management. 1863  
International journal of biological sciences. 2022;18(12):4768–4780. <https://doi.org/10.7150/ijbs.75056>. 1864  
1865  
1866  
1867  
1868

[6] Baazaoui N, Iqbal K. COVID-19 and neurodegenerative diseases: prion- 1869  
like spread and long-term consequences. Journal of Alzheimer's Disease. 1870  
2022;88(2):399–416. <https://doi.org/10.3233/JAD-220105>. 1871  
1872  
1873  
1874

[7] Channappanavar R, Perlman S. Pathogenic human coronavirus infections: 1875  
causes and consequences of cytokine storm and immunopathology. Sem- 1876  
inars in Immunopathology. 2017;39(5):529–539. <https://doi.org/10.1007/s00281-017-0629-x>. 1877  
1878  
1879  
1880  
1881

[8] Frank-Cannon TC, Alto LT, McAlpine FE, Tansey MG. Does neuroinflamma- 1882  
tion fan the flame in neurodegenerative diseases? Molecular neurodegeneration. 1883  
2009;4:47. <https://doi.org/10.1186/1750-1326-4-47>. 1884  
1885  
1886

1887 [9] Fraunhofer Institute for Algorithms and Scientific Computing SCAL.: COM-  
1888 MUTE Project. Accessed: 2025-12-04. [https://www.commute-project.eu/en/](https://www.commute-project.eu/en/about.html)  
1889 [about.html](https://www.commute-project.eu/en/about.html).  
1890  
1891  
1892  
1893 [10] Francis N, Green A, Guagliardo P, Libkin L, Lindaaker T, Marsault V, et al.  
1894 Cypher: An evolving query language for property graphs. In: Proceedings of the  
1895 2018 international conference on management of data. Association for Computing  
1896 Machinery; 2018. p. 1433–1445. Available from: [https://doi.org/10.1145/3183713.](https://doi.org/10.1145/3183713.3190657)  
1897 [3190657](https://doi.org/10.1145/3183713.3190657).  
1898  
1899  
1900  
1901  
1902 [11] Guo A, Li X, Xiao G, Tan Z, Zhao X. SpCQL: A Semantic Parsing Dataset  
1903 for Converting Natural Language into Cypher. In: Proceedings of the 31st ACM  
1904 International Conference on Information & Knowledge Management. CIKM '22.  
1905 New York, NY, USA: Association for Computing Machinery; 2022. p. 3973–3977.  
1906 Available from: <https://doi.org/10.1145/3511808.3557703>.  
1907  
1908  
1909  
1910  
1911 [12] Litvin AA, Velychko VY, Kaverynskyi VV. TREE-BASED SEMANTIC  
1912 ANALYSIS METHOD FOR NATURAL LANGUAGE PHRASE TO FORMAL  
1913 QUERY CONVERSION. Radio Electronics, Computer Science, Control. 2021  
1914 Jul;(2):105–113. <https://doi.org/10.15588/1607-3274-2021-2-11>.  
1915  
1916  
1917  
1918  
1919 [13] Feng G, Zhu G, Shi S, Sun Y, Fan Z, Gao S, et al. Robust NL-to-Cypher Trans-  
1920 lation for KBQA: Harnessing Large Language Model with Chain of Prompts.  
1921 In: Wang H, Han X, Liu M, Cheng G, Liu Y, Zhang N, editors. Knowledge  
1922 Graph and Semantic Computing: Knowledge Graph Empowers Artificial General  
1923 Intelligence. vol. 12923 of Communications in Computer and Information Sci-  
1924 ence. Singapore: Springer Nature Singapore; 2023. p. 317–326. Available from:  
1925 [https://doi.org/10.1007/978-981-99-7224-1\\_25](https://doi.org/10.1007/978-981-99-7224-1_25).  
1926  
1927  
1928  
1929  
1930  
1931  
1932

- [14] Hornsteiner M, Kreussel M, Steindl C, Ebner F, Empl P, Schöning S. Real-Time Text-to-Cypher Query Generation with Large Language Models for Graph Databases. *Future Internet*. 2024;16(12):438. <https://doi.org/10.3390/fi16120438>. 1933  
1934  
1935  
1936  
1937  
1938  
1939
- [15] Zou Y, Wang Y, Liu D. Q2Cypher: Converting Natural Language Questions to Cypher with Fine-Tuned Large Language Models. In: 2024 5th International Conference on Artificial Intelligence and Computer Engineering (ICAICE); 2024. p. 783–788. Available from: <https://doi.org/10.1109/ICAICE63571.2024.10863984>. 1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947
- [16] Ding Y, Fan W, Ning L, Wang S, Li H, Yin D, et al.: A Survey on RAG Meets LLMs: Towards Retrieval-Augmented Large Language Models. *ArXiv preprint arXiv:2405.06211*. Available from: <https://doi.org/10.48550/arXiv.2405.06211>. 1948  
1949  
1950  
1951  
1952  
1953
- [17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*. 2020;33:9459–9474. 1954  
1955  
1956  
1957  
1958  
1959
- [18] Guvenilir HA, Sethumadhavan P, Kaladharan A, de Sa RV, Sridhar A, Haferkamp U, et al. CBM KG: A Comorbidity-Centric Knowledge Graph Uncovering Causal Pathomechanisms Between COVID-19 and Neurodegenerative Diseases. *bioRxiv*. 2025;p. 2025.08.18.670790. <https://doi.org/10.1101/2025.08.18.670790>. 1960  
1961  
1962  
1963  
1964  
1965  
1966
- [19] Babaiha NS, Geissler S, Nibart V, Atas Güvenilir H, Bharadhwaj VS, Tom Kodamullil A, et al. Unraveling the co-morbidity between COVID-19 and neurodegenerative diseases through multi-scale graph analysis: A systematic investigation of biological databases and text mining. *Artificial Intelligence in the Life Sciences*. 2025;8:100138. <https://doi.org/10.1016/j.ailsci.2025.100138>. 1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978

1979 [20] SCAI-BIO.: CBM Comorbidity Knowledge Graph. Accessed 8 September 2026.  
1980  
1981 <https://github.com/SCAI-BIO/CBM-Comorbidity-KG>.  
1982  
1983 [21] SCAI-BIO.: COVID-19 and Neurodegenerative Disease Comorbidity Knowl-  
1984 edge Graph. Accessed 8 September 2026. [https://github.com/SCAI-BIO/](https://github.com/SCAI-BIO/covid-NDD-comorbidity-NLP)  
1985 [covid-NDD-comorbidity-NLP](https://github.com/SCAI-BIO/covid-NDD-comorbidity-NLP).  
1986  
1987  
1988  
1989 [22] SCAI-BIO.: CGEx: Cypher Generating Expert. Accessed 8 September 2026.  
1990 <https://github.com/SCAI-BIO/CGEx>.  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryMaterialCGExevaluationresults.zip](#)