# BioVerge: A Comprehensive Benchmark and Study of Self-Evaluating Agents for Biomedical Hypothesis Generation

**Anonymous ACL submission**


## Abstract

Hypothesis generation in biomedical research has traditionally centered on uncovering hidden relationships within vast scientific literature, often using methods like Literature-Based Discovery (LBD). Despite progress, current approaches typically depend on single data types or predefined extraction patterns, which restricts the discovery of novel and complex connections. Recent advances in Large Language Model (LLM) agents show significant potential, with capabilities in information retrieval, reasoning, and generation. However, their application to biomedical hypothesis generation has been limited by the absence of standardized datasets and execution environments. To address this, we introduce BIOVERGE, a comprehensive benchmark, and BIOVERGE AGENT, an LLM-based agent framework, to create a standardized environment for exploring biomedical hypothesis generation at the frontier of existing scientific knowledge. Our dataset includes structured and textual data derived from historical biomedical hypotheses and PubMed literature, organized to support exploration by LLM agents. BIOVERGE AGENT utilizes a ReAct-based approach with distinct Generation and Evaluation modules that iteratively produce and self-assess hypothesis proposals. Through extensive experimentation, we uncover key insights: 1) different architectures of BIOVERGE AGENT influence exploration diversity and reasoning strategies; 2) structured and textual information sources each provide unique, critical contexts that enhance hypothesis generation; and 3) self-evaluation significantly improves the novelty and relevance of proposed hypotheses. Our dataset and code are accessible here: `https://anonymous.4open.science/r/BioVerge/`


## 1 Introduction

Traditional hypothesis generation tasks centered around finding hidden relationships among concepts within the scientific domain. A widely recognized method was **Literature Based Discovery** (LBD), which highlights that undiscovered knowledge exist within scientific literature and are obtainable through automated extraction systems Bhasuran et al. (2023). Importantly, the ABC principle depicted in Figure 1 tackles the LBD task Swanson (1986), stating that if distinct literature connects concepts $A/B$ and concepts $B/C$, then the intersections of concept $B$ infers potential links between concepts $A/C$. This technique harnessed many novel relationships that were proven experimentally, including connections between magnesium deficiency and migraine, Raynaud's disease and fish oil, significantly contributing to scientific research Swanson (1988); Bhasuran et al. (2023). Despite earlier successes, these traditional co-occurrence, semantic, and graph-based methods often relied on a single data type, limiting their ability to discover subtler connections Smalheiser et al. (2009); Fleuren and Alkema (2015); Jha et al. (2018); Wilson et al. (2018); Tshitoyan et al. (2019); Sybrandt et al. (2020). Furthermore, with little reasoning or refinement, these approaches fail to develop intricate reasoning paths that lead to more discoveries Hristovski et al. (2006).

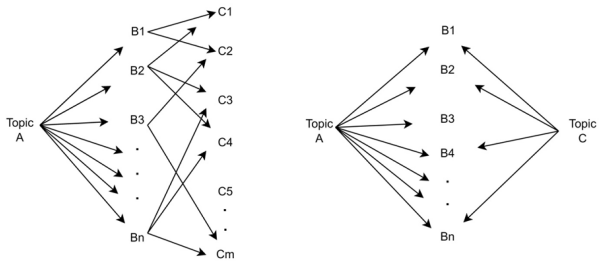


Figure 1: The ABC principle as a method to explore Literature Based Discovery (LBD).

Recently, Large Language Models (LLM) agents have garnered significant interest from its performance on exploration, reasoning, and generation tasks Tong et al. (2023); Wang et al. (2024); Ye et al. (2024). LLM-based agents can utilize tools to gather information, perform thoughtful reasoning,

and create novel proposals Qi et al. (2023); Baek et al. (2024); Zhang et al. (2024); Kumbhar et al. (2025); Su et al. (2025). However, despite its potential, LLM agents have limited applications in the biomedical hypothesis generation, due to a lack of standardized benchmarks and databases.

To address this gap, we introduce BIOVERGE, a benchmark that stores biomedical knowledge as structured hypotheses triplets from PubTator3 and textual PubMed literature Wei et al. (2024); PubMed (2025). The knowledge base contains data published before Jan. 1, 2024 to prevent test set data contamination, and both datasets perform cleaning and filtering to ensure experiment and evaluation fairness. We rank candidate test hypotheses by the impact factor (IF) of their publishing journals to select a credited, reliable test set SCImago (2025).

We also propose BIOVERGE AGENT, an LLM agent framework leveraging BIOVERGE to perform biomedical hypothesis generation. We construct an API interface to support agent tool calling to extensively explore and retrieve context in natural language. By alternating between *Generation* and *Evaluation* modules, BIOVERGE AGENT can propose, self-evaluate, and iteratively refine hypothesis. We further compare Single and Double Agent architectures, which differ in memory sharing between the modules, to explore their impacts on hypothesis generation reasoning and performance.

Finally, we perform extensive experiments on our constructed test dataset. Our results reveal that Single and Double Agent clearly present different reasoning strategies and suggests a trade-off between exploration and accuracy performance. Additionally, the access to diverse structural and textual sources, and self-evaluation of past proposals, all encourage BIOVERGE AGENT to refine its reasoning trajectory and notably improve the novelty and relevance of hypothesis proposals.

Overall, our main contributions are three-fold:

1. We introduce the BIOVERGE benchmark, which enables tool-augmented reasoning over both structured and unstructured biomedical data.
2. We demonstrate the capability of BIOVERGE AGENT to generate hypotheses that push the boundaries of existing biomedical knowledge.
3. We present a systematic study of self-evaluating agent architectures and information sources, identifying key design choices that enhance biomedical hypothesis generation.

Table 1: Dataset stats for knowledge base and test set.

| Dataset | Triplets | Publications | Unique Entities |
|---|---|---|---|
| Knowledge Base | 10,566,798 | 8,896,514 | 558,006 |
| Test Set | 177 | 135 | 174 |

## 2 BIOVERGE Benchmark

We define **hypothesis** as a triplet $(s, r, o)$, where $s, o \in \mathcal{E}$ are biomedical entities and $r \in \mathcal{R}$ is one of twelve relation types in Table 5 of Appendix section B.1, all defined in the PubTator3 ontology (Wei et al., 2024). We denote **hypothesis description** $d$ as a natural language explanation for $(s, r, o)$.

**Biomedical hypothesis generation task.** Given entities $s, o \in \mathcal{E}$, propose relation $r \in \mathcal{R}$ and description $d$ such that $(s, r, o) \notin H$ (knowledge base), $(s, r, o) \in T$ (test set), $d$ is distinct from historical articles with hypotheses of the same $s, o$ entities and aligns with ground truth articles.

### 2.1 Dataset construction

In following sections we describe the construction and cleaning of the historical knowledge base and test set, along with evaluation metrics for experiment studies. Dataset statistics are in Table 1.

#### 2.1.1 Knowledge base

BIOVERGE contains structured triplets, literature articles, and knowledge graph, shown in Figure 2. Specifically, the knowledge base contains hypotheses and articles published before Jan. 1, 2024.

**Structured sources.** PubTator3 provides abundant article-extracted hypothesis triplets, but limited natural language entity names. Therefore, we standardize triplet entities from the official databases: Medical Subject Headings or Mesh (chemical and disease), NCBI Gene (gene), NCBI Taxonomy (species), and Cellosaurus (cellline) (National Library of Medicine, 2025; National Center for Biotechnology Information, 2025a,b; Bairoch, 2018). All entities are appended with entity names, except "mutation" types due to lack of interpretability. We further construct a searchable knowledge graph with entity nodes and relation edges. Each unique triplet is a transition $(s \xrightarrow{r} o)$, where $(s \xrightarrow{r} o)$ and $(o \xrightarrow{r} s)$ are distinct transitions, possibly with different relations.

**Textual sources.** PubMed articles identified with titles and abstracts map to all triplets in the knowledge base, providing textual historical contexts. Articles missing publication date or its title and abstract are discarded from knowledge base.

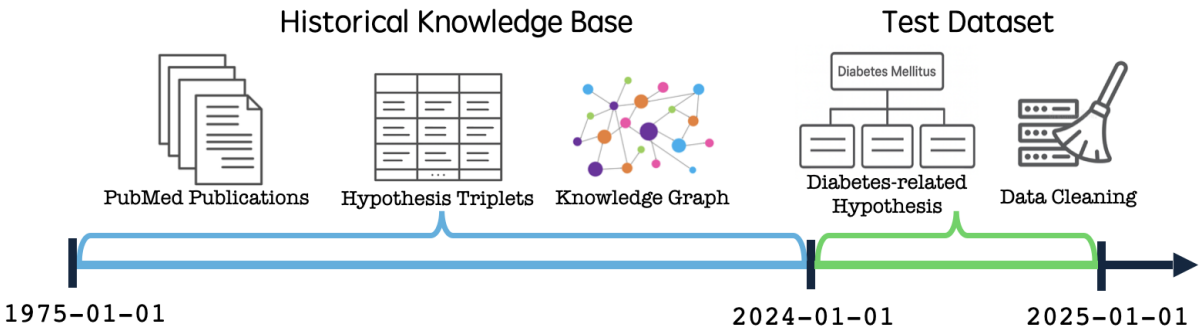


Figure 2: The cutoff date separation between the historical knowledge base and test dataset candidates for dataset construction. The knowledge base contains article corpus, knowledge graph, and triplet representations. The test dataset is cleaned to ensure all test queries are proposed after Jan. 1, 2024, related to Diabetes Mellitus, and ranked by the impact factor (IF) score of its publishing journal.

**Data cleaning and processing.** We remove triplets that either have missing entity names or is not a valid triplet $(s, r, o)$ combination defined in the PubTator3 report, shown in Table 5 in Appendix section B.1. We then merge triplets that appear in multiple articles, append all unique PMIDs, and assign the earliest publication date as the discovery date. Finally, we discard any triplets with no associated articles to ensure every triplet links to verifiable literature.

#### 2.1.2 Test set

The test dataset contains hypotheses discovered between Jan. 1 and Dec. 31, 2024 involving "Diabetes Mellitus" (MeSH: $D003920$), thereby developing a **diabetes-related** test set. We then apply similar data cleaning as the knowledge base to filter suboptimal triplets and ensure they are absent prior to Jan. 1, 2024. Finally, we rank the remaining hypotheses with the **journal impact factor (IF)** score documented in Scimago's Scientific Journal Rank (SJR) metrics SCImago (2025). Taking the top 50 highest ranked journals, we curate a test set of 177 hypotheses spanning diverse relation types, triplet combinations, and diabetes entities under "Diabetes Mellitus."

### 2.2 Answer evaluation metric

We evaluate each query's hypothesis proposal by comparing its relation $r$ and description $d$ against the ground truth using two metric categories:

1) **Novelty**: whether the hypothesis is absent from knowledge base. We define $\texttt{novelty}_r$ to check if $(s, r, o) \notin H$, where $H$ is the historical knowledge base, and $\texttt{novelty}_d$ to assesses whether the generated hypothesis description $d$ is semantically different from historical articles that have proposed hypothesis of the same entities $s$ or $o$.

2) **Alignment**: whether the hypothesis exists in the test set. We define $\texttt{alignment}_r$ to check if $(s, r, o) \in T$, where $T$ is the test set, and $\texttt{alignment}_d$ to evaluate the similarity of the generated hypothesis description $d$ to the ground truth articles that support the test hypothesis.

Evaluation metrics for relations, $\texttt{novelty}_r$ and $\texttt{alignment}_r$, are computed as binary scores (1 = true, 0 = false) and expressed as percentages over the test set. Description-based metrics $\texttt{novelty}_d$ and $\texttt{alignment}_d$ are scored by an LLM evaluator out of 100 and averaged over the test set.

## 3 BIOVERGE AGENT

Considering the individual benefits of co-occurrence, semantic, and graph-based methods, we introduce BIOVERGE AGENT, an LLM agent framework that combines these approaches for hypothesis generation. We develop an API interface for agent tool calling to retrieve diverse historical information and introduce two modules, *Generation* and *Evaluation*, to iteratively generate, self-evaluate, and refine hypothesis proposals. Furthermore, we examine the effects on hypothesis generation outcomes when applying these modules in distinct Single and Double agent architectures.

### 3.1 API usage

We construct a comprehensive API interface for agent tool calling to access our knowledge base. The interface standardizes historical knowledge retrieval into *data classes* (Entity, Relation, PMID, Triplet, Article) and exposes a set of *functions* for the agent to call and retrieve relevant contexts.

**API functions.** The API interface covers all data types, providing both structural and textual historical contexts. Core APIs such as `get_entities`, `get_relations`, and `get_articles` return the most relevant entities, relations, and articles for a given query, while auxiliary functions like `get_relation_description`

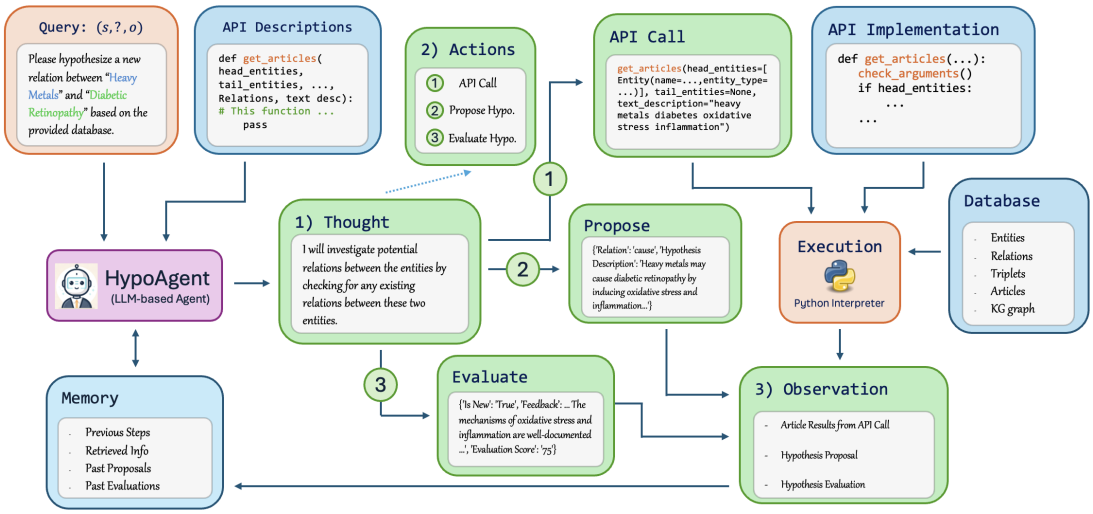


Figure 3: *Generation* and *Evaluation* modules' ReAct workflow. Given a query, the agent iteratively reflects on its current state, executes an action, and stores the observed output until termination.

and `get_entity_description` provide additional semantic detail. Together, these APIs enable the agent to flexibly combine structured triplets with article context. The complete API list is in Table 6 of Appendix section B.2.

**Input parameters.** Each API accepts optional parameters that guide query results. Common parameters include `head_entities`, `tail_entities`, `relations`, `PMIDs`, and `text_description`. Varying these parameters enable BIOVERGE AGENT to specialize the query criteria, constrain the search space, and return results that best match the provided inputs.

### 3.2 Generation and evaluation agent architectures

Proposing a new relation between two entities is inherently difficult because no prior literature or triplets exist to document this connection. Even when past hypotheses mention the same entities, they rarely provide direct support for new proposals. Thus, hypothesis generation requires extensive exploration and reasoning over historical data, with no guarantee of correctness.

**Modules.** To address these challenges, BIOVERGE AGENT introduces *Generation* and *Evaluation* modules (system prompts in Appendix section E). We also propose two agent architectures, shown in Figure 4, and explain a practical execution example in Figure 9 of Appendix section C.

The ***Generation*** module proposes new hypotheses, including the relation and hypothesis description, and refines previously proposed hypotheses. The ***Evaluation*** module acts as a critic that accesses the most recently proposed hypothesis. The evaluation specifically addresses 3 metrics: 1) **Is New**, a boolean value stating the novelty of the hypothesis from the historical dataset; 2) **Feedback**, a natural language explanation that justifies the reasonability of the proposed hypothesis and offers suggestions for improvement; and 3) **Evaluation Score**, a numeric score from 0 to 100 assessing the hypothesis reasonableness based on **Feedback**.

**ReAct Framework.** Both modules use ReAct framework shown in Figure 3 to iterate over 3 steps:

1. **Think**: Use past memory to choose the next action. *Generation* module may call APIs or propose a hypothesis, while *Evaluation* module can call APIs or evaluate a proposed hypothesis.
2. **Act**: Execute API queries via tool calling or propose/evaluate a hypothesis in JSON format.
3. **Observe**: Record results in memory. Hypothesis proposals pass to *Evaluation* module for assessment while feedback returns to *Generation* module for hypothesis refinement.

**Agent architecture.** 1) **Single Agent**: The two modules share a common memory log, allowing the *Evaluation* module to access *Generation* module's historical information, reasoning steps, and hypothesis proposals, and vice versa. 2) **Double Agent**: The two modules have separate memory logs with no access to each other's past actions or observations. This setup creates two separate agents: a *Generator* agent as the *Generation* module and an *Evaluator* agent as the *Evaluation* module.

**Design Rationale.** We study both Single and Double Agent to understand their differences in tool calling, reasoning, and self-evaluation for hypothesis generation. In Single Agent, shared memory between modules reduces overall number of API queries and computation cost, while still allowing the *Evaluation* module to provide targeted feedback for *Generation* module to refine proposals. On the other hand, separating memory logs

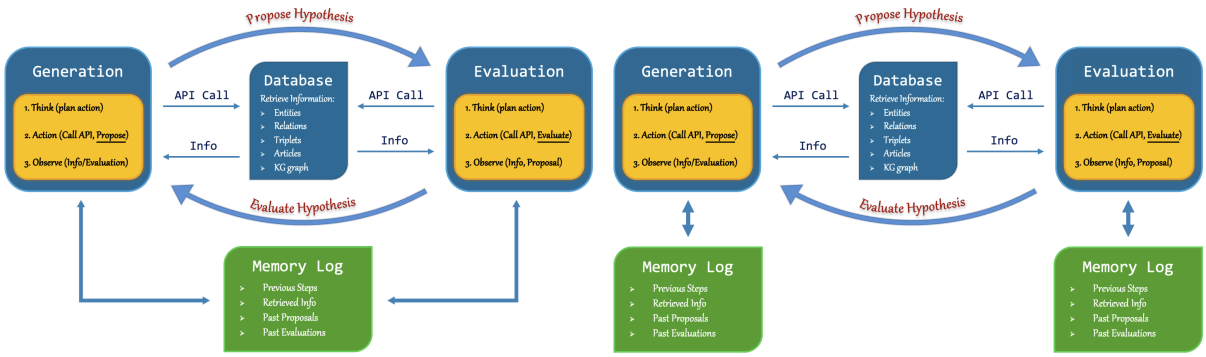


Figure 4: Single Agent (left) and Double Agent (right) architectures. Memory is shared between *Generation* and *Evaluation* modules in Single Agent and separated in Double Agent.

in Double Agent enables *Generator* and *Evaluator* to explore independently, encourages diverse reasoning trajectories, and reduces the risk of hallucination. Both agent frameworks can offer distinct advantages for hypothesis generation, therefore we evaluate both frameworks in our experiments.

**Evaluation Threshold (ET).** BIOVERGE AGENT accepts an Evaluation Threshold (ET) which is a criterion for terminating the hypothesis generation process, available to both modules. When the **Evaluation Score** from *Evaluation* module reaches ET, the agent terminates and extracts the latest proposal as the final answer. Otherwise, if maximum allowable iterations is reached, the agent's entire memory log is sent to a LLM extractor that returns the most reasonable and confident hypothesis proposal as the final answer.

## 4 Experiments

### 4.1 Experiment settings

For all experiments, we standardize a common set of hyperparameter settings across BIOVERGE AGENT and baselines. The temperature is 0.7 for all LLM queries in agent ReAct steps and baselines, and 0.2 for extracting the proposed hypotheses and evaluation feedback. The model used for all agent frameworks and baselines is gpt-4o-mini due to limitations in computational resources and consideration for potential dataset contamination. For BIOVERGE AGENT experiments, we standardize the following metrics across Single and Double Agent architectures: max_outer_iterations of 3 specifying the maximum number of alternations between *Generation* and *Evaluation* modules; max_inner_iterations of 10 specifying the maximum number of ReAct steps in both modules. Evaluation thresholds 30, 50, 70, and 90 are tested for Single and Double Agent to explore its importance in agent reasoning and decision making.

### 4.2 Baselines

We evaluate several baseline methods by directly providing relevant historical information to the baselines. The results are displayed in Table 2.

**All baselines show limitations under certain metrics.** Article Retrieval Augmented Generation (RAG) achieves the highest description alignment of 57.59, reflecting the importance of textual sources to ground hypothesis proposals; however, it suffers from worst overall relation and description novelty. Triplet RAG achieves highest relation novelty of 97.18, revealing that structural data helps to validate novelty, yet its relation alignment of 32.20 is the lowest among baselines. Finally, Chain-of-Thought (CoT) achieve a descent relation alignment at 36.72 but scores low on relation novelty and document alignment, suggesting strong limitations without external data sources.

### 4.3 Agents

**BIOVERGE AGENT achieves high novelty and alignment.** Table 3 documents agent execution results. The best setting is Single Agent with evaluation threshold 50, achieving a relation alignment of 38.42%. All experiments achieved high relation novelty, above 98% across evaluation thresholds. This shows that BIOVERGE AGENT has strong understanding of novelty, as the agent extensively self-evaluates proposals from *Generation* module in the *Evaluation* module.

#### 4.3.1 Iteration patterns across evaluation thresholds

Figure 5 presents average inner and outer iteration counts for Single and Double Agent across evaluation thresholds. The left subplot shows the number of agent ReAct steps within *Generation* and *Evaluation* modules per outer loop, while the right subplot counts the number of Generation-Evaluation

Table 2: Average performance of CoT, Triplet RAG, and Article RAG baselines.

| Setting | $\text{novelty}_r$ | $\text{alignment}_r$ | $\text{novelty}_d$ | $\text{novelty}_d\,(\sigma)$ | $\text{alignment}_d$ | $\text{alignment}_d\,(\sigma)$ |
|---|---|---|---|---|---|---|
| CoT | 94.35 | 36.72 | 65.08 | 20.44 | 54.97 | 32.32 |
| Triplet RAG | **97.18** | 32.20 | **65.93** | 20.42 | 55.51 | 33.32 |
| Article RAG | 93.22 | **43.37** | 63.42 | 20.26 | **57.59** | 33.84 |

Table 3: Average performance of Single and Double Agent across evaluation thresholds (%).

| Model | ET | $\text{novelty}_r$ | $\text{alignment}_r$ | $\text{novelty}_d$ | $\text{novelty}_d\,(\sigma)$ | $\text{alignment}_d$ | $\text{alignment}_d\,(\sigma)$ |
|---|---|---|---|---|---|---|---|
| Single Agent | 30 | 99.43 | 32.76 | 64.24 | 20.51 | 55.54 | 33.11 |
| | 50 | **100.00** | **38.42** | 63.39 | 20.49 | 54.10 | 33.00 |
| | 70 | **100.00** | 33.89 | 63.53 | 20.67 | 55.65 | 32.74 |
| | 90 | **100.00** | 31.07 | 63.89 | 21.09 | 55.23 | 32.55 |
| Double Agent | 30 | 98.31 | 25.98 | 64.94 | 20.35 | 55.00 | 32.93 |
| | 50 | **100.00** | 25.98 | 66.18 | 20.33 | 54.88 | 33.12 |
| | 70 | 98.31 | 22.60 | 65.37 | 20.31 | **56.92** | 31.83 |
| | 90 | 99.44 | 17.51 | **66.41** | 19.66 | 56.36 | 32.69 |

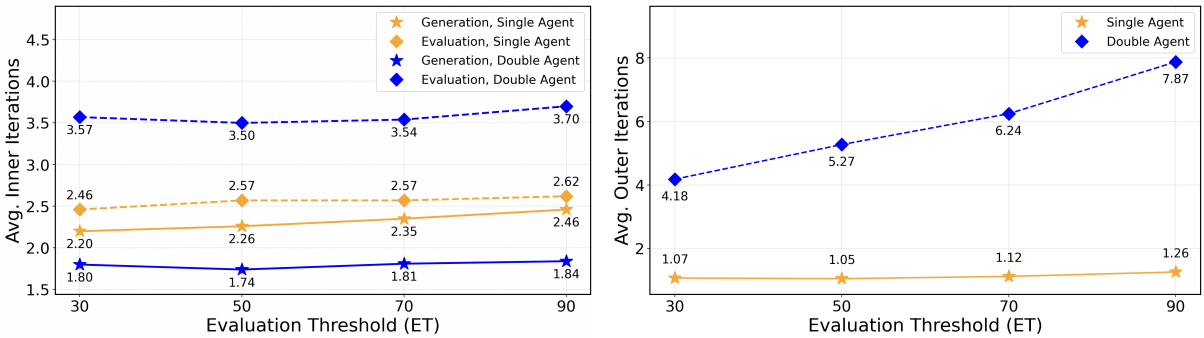


Figure 5: Average inner iteration counts (left) and outer iteration counts (right) for Single and Double Agent across evaluation thresholds.

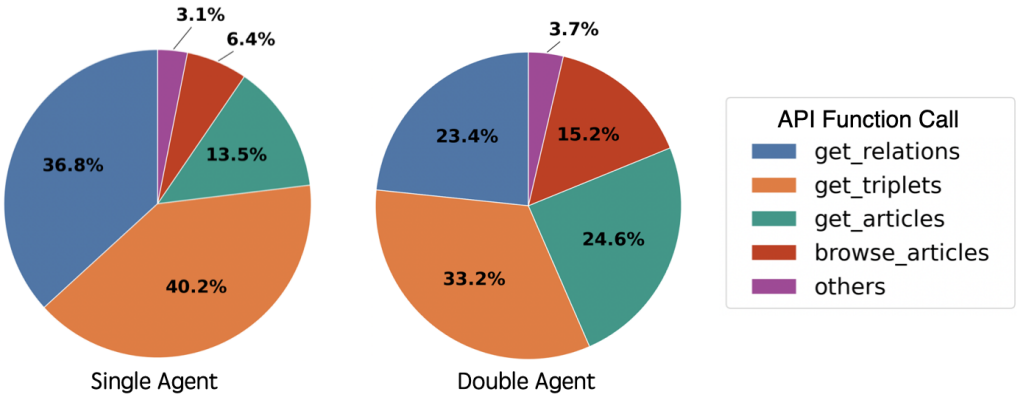


Figure 6: Average number of API calls for Single Agent (left) and Double Agent (right).

iterations before termination. All values are averaged over the test dataset.

**Both agents prioritize evaluation over generation.** Single and Double Agent both perform more evaluation steps than generation steps, which suggests their focus on self-evaluation in the *Evaluation* module over frequent *Generation* module iterations. Specifically, Double Agent performs on average 1 more evaluation step than Single Agent, indicating that the sharing of memory logs influences Single Agent to execute less.

**Double Agent performs more outer iterations than Single Agent.** Double Agent performs more outer loops as evaluation threshold increases, shown as a gradual increase from 4.18 to 7.87, exhibiting a reasonable correlation between compute and evaluation score. In contrast, Single Agent's outer iteration count stabilizes around 1.05–1.26.

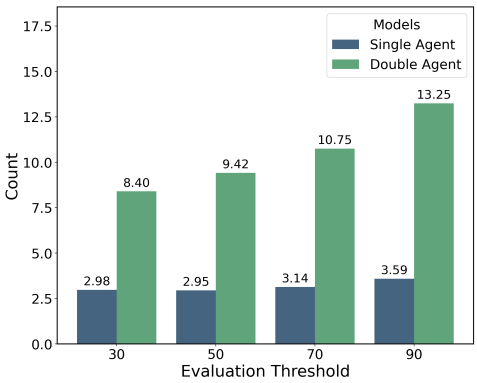


Figure 7: Total number of API calls for Single and Double Agent across evaluation thresholds.

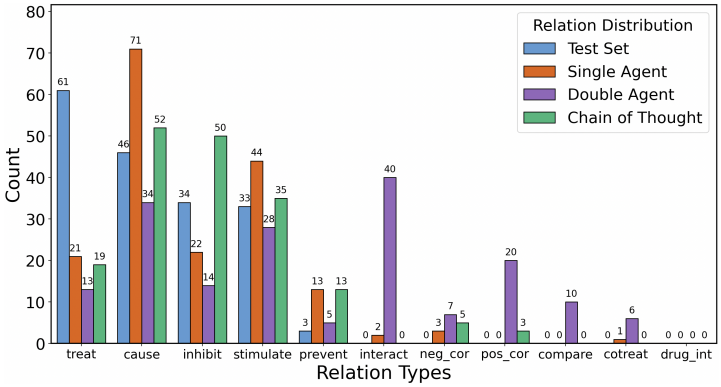


Figure 8: Single Agent, Double Agent, CoT output distribution vs. test set relation distribution. *drug_int*, *neg_cor*, and *pos_cor* refer to *drug interact*, *negative correlate*, and *positive correlate* respectively.

This stabilization might suggest Single Agent's overconfidence in self-evaluation to produce a satisfactory evaluation score, leading to smaller number of outer iterations across different thresholds.

#### 4.3.2 Agent API usage and strategy

Figure 6 shows the distribution of API tool calls during hypothesis generation, such as for querying entities, retrieving historical relations, and browsing articles. The percentages are averaged over multiple Single and Double Agent experiments under various evaluation thresholds.

**Single Agent focuses on structured data sources.** The majority of Single Agent tool calls concentrate on `get_relations` and `get_triplets`, each accounting for over 1/3 of overall APIs. This reflects Single Agent's focus to directly extract historical hypotheses and its smaller capacity for prolonged iterative refinement. Because it typically executes less outer iterations, Single Agent has fewer opportunities to explore and self-evaluate. Therefore, it prioritizes straightforward reasoning paths grounded in structured hypotheses over article sources.

**Double Agent balances structured and literature sources.** Double Agent distributes its tool calls more evenly between structured sources and article-related APIs, as the modular design encourages broader exploration and thorough self-evaluation. By revising suboptimal hypotheses proposals and incorporating article evidence alongside relations and triplets, Double Agent expands its reasoning trajectory and reduces reliance on single data sources.

#### 4.3.3 Agent total API usage

**Double Agent calls more APIs than Single Agent.** Figure 7 presents the total number of API tool calls across evaluation thresholds for Single and Double Agent, averaged over the test dataset. Double Agent executes a high number of tool calls, with values ranging from 8.40 to 13.25 as ET increases; in contrast, Single Agent exhibits less tool calls, between 2 to 4 calls per query. This result is expected since *Generation* and *Evaluation* modules in Single Agent shares the memory log, therefore less API requests are needed to retrieve the relevant context. On the other hand, Double Agent executes significantly more tool calls, trading runtime efficiency and computational resources for potentially greater thoroughness in exploration.

### 4.4 Error analysis

In this section we further discuss insights into the agent's behavior and explore aspects that undermine BIOVERGE AGENT's overall capabilities and effectiveness in hypothesis generation. We compare the output relation distributions of Single Agent, Double Agent, and the Chain of Thought baseline with the test dataset distribution, shown in Figure 8. No experiment settings has prior knowledge regarding the test dataset relation distribution.

**Single Agent biased towards causal relations.** Single Agent's and CoT baseline's relation distributions strongly skew towards causal relations of *treat, cause, inhibit, stimulate, and prevent*. This pattern is likely due to the agent's innate preference towards proposing and rewarding hypotheses that connote strong causality, which are easier to semantically identify and disproportionally represented in literature corpora, and are therefore more likely identified as scientifically plausible and relevant. Additionally, the *Evaluation* module in Single Agent possesses the *Generation* module's past reasoning trajectory, which might introduce overconfi-

Table 4: Ablation study results. The base model is Single Agent with evaluation threshold of 50.

| Setting | novelty$_r$ | alignment$_r$ | novelty$_d$ | novelty$_d$ $(\sigma)$ | alignment$_d$ | alignment$_d$ $(\sigma)$ |
|---|---|---|---|---|---|---|
| Relation Only | **100.00** | 32.16 | 64.35 | 20.18 | 53.53 | 33.03 |
| Triplet Only | **100.00** | 32.20 | 63.79 | 21.24 | 54.52 | 32.80 |
| Article Only | 96.61 | 28.25 | 63.08 | 20.84 | **54.66** | 33.38 |
| KG Only | 98.87 | 28.81 | **64.38** | 20.33 | 53.45 | 33.71 |
| Generation Only | **100.00** | **33.89** | 62.85 | 20.81 | 53.93 | 33.49 |

dence during it's self-evaluation feedback for more lenient evaluation scoring. Correlational relations such as *interact* demands more nuanced contextual evidence and receives less attention during earlier iterations of hypothesis proposal.

**Double Agent promotes exploration diversity but deviates from ground truth distribution.** Double Agent produces a more balanced distribution of relation types, with greater inclusion of non-causal relation such as *interact* and *compare*. The strict separation of modules reduces overconfidence and encourages the agent to perform broader exploration beyond causal relations. However, the output alignment with the test set remains low. A key explanation is that the without direct access to the *Generator*'s full reasoning trajectory, the *Evaluator* commonly composes more generic critiques (e.g. "insufficient evidence" or "further exploration needed") rather than precise guidance on hypothesis refinement or reasoning flaws. This inherit dynamic promotes exploration but fails to drive the agent's proposals toward the ground truth distribution.

These findings suggest that robust hypothesis generation requires combining the strengths of both Single and Double Agent, where the feedback from *Evaluation* module is both informative and grounded, without being overly targeted or generic. Developing such framework remains future work for improving the reliability and scientific plausibility of BIOVERGE AGENT.

### 4.5 Ablation studies

Table 4 documents the ablation studies that explore individual information sources available to BIOVERGE AGENT (relation, triplet, article, and knowledge graph) to identify the usage and effectiveness of each datatype for hypothesis generation. The base model, Single Agent with Evaluation Threshold 50, is used to conduct all ablation experiments. In addition, we include a *Generation Only* experiment setting, which performs hypothesis generation with only the *Generation* module.

**All information sources meaningfully improve hypothesis generation.** The ablation results demonstrate that both relations and triplets sources can provide compact historical knowledge to avoid past proposals, achieving 100% relation novelty for both ablations. On the other hand, article corpora offers rich linguistic context and helps to produce more accurate hypothesis descriptions, shown by the highest description alignment score of 54.66. This contrast highlights that a combination of these data sources are necessary for the agent to maintain high novelty while proposing better hypotheses, and removing some sources would compromise the agent's performance.

**Self-evaluation notably enhances performance.** Comparing the result from generation-only ablation to the base Single Agent setting with evaluation threshold 50 shows that removing *Evaluation* module results in a 5% performance degradation, despite the same access to all information sources. This demonstrates that iterative self-evaluations, even if imperfect, can identify and correct illogical reasoning and encourage the agent to propose more relevant hypotheses.

## 5 Conclusion

In this work, we introduced BIOVERGE, a comprehensive benchmark that unifies structured and textual biomedical knowledge to advance hypothesis generation. Our proposed BIOVERGE AGENT framework employs Generation and Evaluation modules within Single and Double Agent architectures, enabling iterative hypothesis generation and refinement through tool-calling, reasoning and self-evaluation mechanisms. Through extensive experiments, we demonstrate that multi-sourced data integration and self-evaluation are critical components for effective biomedical hypothesis generation. These findings establish BIOVERGE benchmark as a valuable resource for the research community and highlight the potential of agent-based approaches for scientific discovery in biomedicine.

## Limitations

The selection of models in the experiment section considers both computational budget and models' training data cutoff. To better assess other newly released models, we plan to regularly update BIOVERGE to incorporate updated test sets with up-to-date time cutoffs to mitigate data contamination. Moreover, this work centers on literature-based hypothesis generation, which represents only one stage of the broader scientific discovery pipeline. Extending future research to encompass downstream tasks such as experimental design and execution would move toward more comprehensive discovery systems. Finally, our current evaluation is limited to diabetes research; applying BIOVERGE AGENT to a wider range of biomedical domains will be essential for establishing broader generalizability and clinical relevance.

## References


Nazmus Ashrafi, Salah Bouktif, and Mohammed Mediani. 2025. Enhancing LLM Code Generation: A Systematic Evaluation of Multi-Agent Collaboration and Runtime Debugging for Improved Accuracy, Reliability, and Latency. *arXiv preprint*. ArXiv:2505.02133 [cs].

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models. *arXiv preprint*. ArXiv:2404.07738 [cs].

Amos Bairoch. 2018. Cellosaurus: a cell-line knowledge resource.

Balu Bhasuran, Gurusamy Murugesan, and Jeyakumar Natarajan. 2023. Literature Based Discovery (LBD): Towards Hypothesis Generation and Knowledge Discovery in Biomedical Text Mining. *arXiv preprint*. ArXiv:2310.03766 [cs].

Trevor Cohen and Dominic Widdows. 2017. Embedding of semantic predications. *Journal of Biomedical Informatics*, 68:150–166.

Trevor Cohen, Dominic Widdows, Roger W. Schvaneveldt, Peter Davies, and Thomas C. Rindflesch. 2012. Discovering discovery patterns with predication-based Semantic Indexing. *J. of Biomedical Informatics*, 45(6):1049–1065.

Wilco W. M. Fleuren and Wynand Alkema. 2015. Application of text mining in the biomedical domain. *Methods (San Diego, Calif.)*, 74:97–106.

Shanghua Gao, Richard Zhu, Zhenglun Kong, Ayush Noori, Xiaorui Su, Curtis Ginder, Theodoros Tsiligkaridis, and Marinka Zitnik. 2025. TxAgent: An AI Agent for Therapeutic Reasoning Across a Universe of Tools. *arXiv preprint*. ArXiv:2503.10970 [cs].

Vishrawas Gopalakrishnan, Kishlay Jha, Guangxu Xun, Hung Q. Ngo, and Aidong Zhang. 2018. Towards self-learning based hypotheses generation in biomedical text domain. *Bioinformatics (Oxford, England)*, 34(12):2103–2115.

Dimitar Hristovski, Carol Friedman, Thomas C Rindflesch, and Borut Peterlin. 2006. Exploiting Semantic Relations for Literature-Based Discovery. *AMIA Annual Symposium Proceedings*, 2006:349–353.

Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2025. CODESIM: Multi-Agent Code Generation and Problem Solving through Simulation-Driven Planning and Debugging. *arXiv preprint*. ArXiv:2502.05664 [cs].

Moksh Jain, Tristan Deleu, Jason Hartford, Cheng-Hao Liu, Alex Hernandez-Garcia, and Yoshua Bengio. 2023. GFlowNets for AI-Driven Scientific Discovery. *Digital Discovery*, 2(3):557–577. ArXiv:2302.00615 [cs].

Kishlay Jha, Guangxu Xun, Yaqing Wang, Vishrawas Gopalakrishnan, and Aidong Zhang. 2018. Concepts-Bridges: Uncovering Conceptual Bridges Based on Biomedical Concept Evolution. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1599–1607, New York, NY, USA. Association for Computing Machinery.

Kishlay Jha, Guangxu Xun, Yaqing Wang, and Aidong Zhang. 2019. Hypothesis Generation From Text Based On Co-Evolution Of Biomedical Concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 843–851, New York, NY, USA. Association for Computing Machinery.

Shrinidhi Kumbhar, Venkatesh Mishra, Kevin Coutinho, Divij Handa, Ashif Iquebal, and Chitta Baral. 2025. Hypothesis Generation for Materials Discovery and Design Using Goal-Driven and Constraint-Guided LLM Agents. *arXiv preprint*. ArXiv:2501.13299 [cs].

Cheryl Lee, Chunqiu Steven Xia, Longji Yang, Jentse Huang, Zhouruixin Zhu, Lingming Zhang, and Michael R. Lyu. 2024. A Unified Debugging Approach via LLM-Based Multi-Agent Synergy. *arXiv preprint*. ArXiv:2404.17153 [cs].

Yangmin Li, Xin Zhang, Xin Bai, Sen Bai, and Zhengang Jiang. 2025. Predicting co-word links via heterogeneous graph convolutional networks. *Scientific Reports*, 15(1):23143. Publisher: Nature Publishing Group.

Anthony ML Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, and Jurgen Del-Favero. 2011. BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biology*, 12(6):R57.

Weibin Lin, Xianli Wu, Zhengwei Wang, Xiaoji Wan, and Hailin Li. 2022. Topic Network Analysis Based on Co-Occurrence Time Series Clustering. *Mathematics*, 10(16):2846. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

Lei Liu, Xiaoyan Yang, Junchi Lei, Yue Shen, Jian Wang, Peng Wei, Zhixuan Chu, Zhan Qin, and Kui Ren. 2024. A Survey on Medical Large Language Models: Technology, Application, Trustworthiness, and Future Directions. *arXiv preprint*. ArXiv:2406.03712 [cs].

Robert J. Millikin, Kalpana Raja, John Steill, Cannon Lock, Xuancheng Tu, Ian Ross, Lam C. Tsoi, Finn Kuusisto, Zijian Ni, Miron Livny, Brian Bockelman, James Thomson, and Ron Stewart. 2023. Serial KinderMiner (SKiM) discovers and annotates biomedical knowledge using co-occurrence and transformer models. *BMC bioinformatics*, 24(1):412.

NLM National Center for Biotechnology Information. 2025a. Ncbi gene.

NLM National Center for Biotechnology Information. 2025b. Ncbi taxonomy.

NCBI National Library of Medicine. 2025. Medical subject headings (mesh).

Thomas O'Brien, Joel Stremmel, Léo Pio-Lopez, Patrick McMillen, Cody Rasmussen-Ivey, and Michael Levin. 2024. Machine learning for hypothesis generation in biology and medicine: exploring the latent space of neuroscience and developmental bioelectricity. *Digital Discovery*, 3(2):249–263. Publisher: Royal Society of Chemistry.

OpenAI. 2024. GPT-4 Technical Report. In *Computation and Language Repository*. ArXiv:2303.08774 [cs].

Wanda Pratt and Meliha Yetisgen-Yildiz. 2003. LitLinker: capturing connections across the biomedical literature. In *Proceedings of the 2nd international conference on Knowledge capture*, K-CAP '03, pages 105–112, New York, NY, USA. Association for Computing Machinery.

PubMed. 2025. National Center for Biotechnology Information. `https://pubmed.ncbi.nlm.nih.gov/`.

Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. 2023. Large Language Models are Zero Shot Hypothesis Proposers. In *Instruction Workshop @ NeurIPS 2023*. ArXiv:2311.05965 [cs].

Shengtian Sang, Zhihao Yang, Xiaoxia Liu, Lei Wang, Yin Zhang, Hongfei Lin, Jian Wang, Liang Yang, Kan Xu, and Yijia Zhang. 2018a. A Knowledge Graph based Bidirectional Recurrent Neural Network Method for Literature-based Discovery. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 751–752.

Shengtian Sang, Zhihao Yang, Lei Wang, Xiaoxia Liu, Hongfei Lin, and Jian Wang. 2018b. SemaTyP: a knowledge graph based literature mining method for drug discovery. *BMC bioinformatics*, 19(1):193.

SCImago. 2025. SJR — SCImago Journal & Country Rank. `https://www.scimagojr.com/aboutus.php`. SCImago Lab, University of Granada, Extremadura, Carlos III (Madrid), and Alcalá de Henares, Spain.

Neil R. Smalheiser, Vetle I. Torvik, and Wei Zhou. 2009. Arrowsmith two-node search interface: a tutorial on finding meaningful links between two disparate sets of articles in MEDLINE. *Computer Methods and Programs in Biomedicine*, 94(2):190–197.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. Many Heads Are Better Than One: Improved Scientific Idea Generation by A LLM-Based Multi-Agent System. *arXiv preprint*. ArXiv:2410.09403 [cs].

D. R. Swanson. 1988. Migraine and magnesium: eleven neglected connections. *Perspectives in Biology and Medicine*, 31(4):526–557.

Don R. Swanson. 1986. Undiscovered Public Knowledge. *The Library Quarterly: Information, Community, Policy*, 56(2):103–118. Publisher: University of Chicago Press.

Justin Sybrandt, Michael Shtutman, and Ilya Safro. 2018. Large-Scale Validation of Hypothesis Generation Systems via Candidate Ranking. *Proceedings: ... IEEE International Conference on Big Data. IEEE International Conference on Big Data*, 2018:1494–1503.

Justin Sybrandt, Ilya Tyagin, Michael Shtutman, and Ilya Safro. 2020. AGATHA: Automatic Graph-mining And Transformer based Hypothesis generation Approach. *arXiv preprint*. ArXiv:2002.05635 [cs, stat].

Song Tong, Kai Mao, Zhen Huang, Yukun Zhao, and Kaiping Peng. 2023. Automating Psychological Hypothesis Generation with AI: Large Language Models Meet Causal Graph. ArXiv:2402.14424 [cs].

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*. ArXiv:2302.13971 [cs].

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98. Publisher: Nature Publishing Group.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. SciMON: Scientific Inspiration Machines Optimized for Novelty. *arXiv preprint*. ArXiv:2305.14259 [cs].

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. PaperRobot: Incremental Draft Generation of Scientific Ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1980–1991. ArXiv:1905.07870 [cs].

Wenxuan Wang, Zizhan Ma, Zheng Wang, Chenghan Wu, Jiaming Ji, Wenting Chen, Xiang Li, and Yixuan Yuan. 2025. A Survey of LLM-based Agents in Medicine: How far are we from Baymax? *arXiv preprint*. ArXiv:2502.11211 [cs].

Chih-Hsuan Wei, Alexis Allot, Po-Ting Lai, Robert Leaman, Shubo Tian, Ling Luo, Qiao Jin, Zhizheng Wang, Qingyu Chen, and Zhiyong Lu. 2024. PubTator 3.0: an AI-powered Literature Resource for Unlocking Biomedical Knowledge. *arXiv preprint*. ArXiv:2401.11048 [cs].

Stephen Joseph Wilson, Angela Dawn Wilkins, Matthew V. Holt, Byung Kwon Choi, Daniel Konecki, Chih-Hsu Lin, Amanda Koire, Yue Chen, Seon-Young Kim, Yi Wang, Brigitta Dewi Wastuwidyaningtyas, Jun Qin, Lawrence Allen Donehower, and Olivier Lichtarge. 2018. Automated literature mining and hypothesis generation through a network of Medical Subject Headings. Pages: 403667 Section: New Results.

Dingkang Yang, Jinjie Wei, Mingcheng Li, Jiyao Liu, Lihao Liu, Ming Hu, Junjun He, Yakun Ju, Wei Zhou, Yang Liu, and Lihua Zhang. 2025. MedAide: Information Fusion and Anatomy of Medical Intents via LLM-based Agent Collaboration. *arXiv preprint*. ArXiv:2410.12532 [cs].

Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. 2024. Large Language Models for Automated Open-domain Scientific Hypotheses Discovery. *arXiv preprint*. ArXiv:2309.02726 [cs].

Chenchen Ye, Ziniu Hu, Yihe Deng, Zijie Huang, Mingyu Derek Ma, Yanqiao Zhu, and Wei Wang. 2024. MIRAI: Evaluating LLM Agents for Event Forecasting. *arXiv preprint*. ArXiv:2407.01231 [cs].

Zhongming Yu, Hejia Zhang, Yujie Zhao, Hanxian Huang, Matrix Yao, Ke Ding, and Jishen Zhao. 2025. OrcaLoca: An LLM Agent Framework for Software Issue Localization. *arXiv preprint*. ArXiv:2502.00350 [cs].

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A Comprehensive Survey of Scientific Large Language Models and Their Applications in Scientific Discovery. *arXiv preprint*. ArXiv:2406.10833 [cs].

# Appendix

## Table of Contents


**1 Introduction** **1**

**2 BIOVERGE Benchmark** **2**
- 2.1 Dataset construction . . . . . 2
  - 2.1.1 Knowledge base . . . . . 2
  - 2.1.2 Test set . . . . . 3
- 2.2 Answer evaluation metric . . . . . 3

**3 BIOVERGE AGENT** **3**
- 3.1 API usage . . . . . 3
- 3.2 Generation and evaluation agent architectures . . . . . 4

**4 Experiments** **5**
- 4.1 Experiment settings . . . . . 5
- 4.2 Baselines . . . . . 5
- 4.3 Agents . . . . . 5
  - 4.3.1 Iteration patterns across evaluation thresholds . . . 5
  - 4.3.2 Agent API usage and strategy 7
  - 4.3.3 Agent total API usage . . 7
- 4.4 Error analysis . . . . . 7
- 4.5 Ablation studies . . . . . 8

**5 Conclusion** **8**

**Appendix** **12**

**A Related Works** **12**
- A.1 AI-driven scientific discovery . . . 12
- A.2 Hypothesis generation . . . . . 13
  - A.2.1 Co-occurrence based methods . . . . . 13
  - A.2.2 Semantic based methods . 13
  - A.2.3 Graph based methods . . . 13
  - A.2.4 Large language models . . 14

**B Benchmark Details** **15**
- B.1 Relation types . . . . . 15
- B.2 API interface . . . . . 15

**C Agent Execution Example** **17**

**D Ablation Studies** **17**

**E System Prompts** **18**


# A Related Works

## A.1 AI-driven scientific discovery

To establish scientific discoveries, scientists employ a holistic procedure to propose, experiment, and validate their research; this process often involves several iterations of data collection and analysis, proposal construction, experimentation, and reflections Qi et al. (2023); Jain et al. (2023).

1) **Data Extraction & Analysis**: The first step in scientific inquiry involves extracting and inspecting experiment findings from previous research, to discern specific patterns or intrinsic correlations. Identifying inherent characteristics within the data is fundamental to establish and propel a novel scientific research. 2) **Establish Hypothesis**: Next, hypotheses are crafted to conjecture the extracted insights. This process involves formulating thoughtful predictions to explain observations from data analysis, which dictates the subsequent research and potential outcomes. Therefore, well-thought hypothesis proposals are critical to facilitate experimentation. 3) **Experiment Design**: To justify the formulated hypothesis, scientists design experiments through careful manipulation of experimental and confounding variables and define a comprehensive procedure to ensure validity and repeatability. A well-designed scientific study should encompass proper experimental practices and yield relevant results. 4) **Experimentation**: Closely adhering to previously established procedures, scientists rigorously conduct experiments to extensively collect measurements and observations. In addition, executing multiple experimental trials ensures that employed methodologies produce significant and unbiased results for succeeding evaluations. 5) **Reflection**: The collected data is thoroughly analyzed under assumptions of the initial proposal, which either supports or refutes the hypothesis. Based on the conclusion, scientists are encouraged to revise their hypothesis and conduct further exploration from step one, repeating the cycle until satisfactory results are obtained and research findings are sufficient for publication.

**Hypothesis generation is pivotal**: The documentation of the research inquiry process demonstrates its importance in unearthing new breakthroughs within various disciplines such as psychology, medicine, and medicine Tong et al. (2023); Sang et al. (2018b); O'Brien et al. (2024). In particular, deriving a novel hypothesis can be of uttermost importance as it directly influences the

efficiency and success of the overall scientific inquiry. An well-crafted hypothesis reduces wasteful resources during prolonged experimentation, narrows the search space, and increases the likelihood of novel and meaningful discoveries. In domains like biomedicine, where the search space is vast and complex, well-formulated hypotheses are essential to guide resource-effective experimentation and accelerate breakthroughs.

### A.2 Hypothesis generation

Contemporary methods in LBD can be predominantly categorized into: 1) Co-occurrence based, 2) semantic relation based, and 3) graph based approaches Bhasuran et al. (2023).

#### A.2.1 Co-occurrence based methods

One approach utilizes the **co-occurrence** of known entities among literature collections. The premise is that if multiple entities, such as title keywords, phrases, and/or specific ideas, co-occur significantly in a diverse range of scientific literature without previously declared associations, then they are possibly correlated via some relationship Fleuren and Alkema (2015). For example, Arrowsmith attempts to determine concepts or entities that connect two domain disparate articles through co-occurring entities in their literature titles Smalheiser et al. (2009). Other works have expanded upon these methods by incorporating natural language processing and temporal based techniques to more effectively capture links among concept pairs or terms Pratt and Yetisgen-Yildiz (2003); Millikin et al. (2023); Lin et al. (2022); Li et al. (2025). However, most traditional co-occurrence based methods do not consider the situational and fine-grained contexts within literature, which led to often irrelevant and incorrect results Hristovski et al. (2006). These methods also require manual effort to identify potentially correlated literature, making the process tedious and poorly suited for the scale and diversity of modern scientific literature.

#### A.2.2 Semantic based methods

A second branch of work, **semantic based methods**, focused on incorporating semantic meanings of concepts using various NLP techniques to improve upon co-occurrence based approaches and increase the relevance and validity of hypothesis Hristovski et al. (2006). For instance, past works represent terms using word embedding vectors to capture a deeper understanding of concepts and literature for hypothesis generation Tshitoyan et al. (2019); Cohen et al. (2012). Some works inject temporal components to capture the changes and relevancy in pairwise concept links overtime Jha et al. (2018, 2019). Other works utilize predicate triples (subject-predicate-object) or MeSH terms to obtain the meanings within literature, often used in junction with graph representations Bhasuran et al. (2023); Wilson et al. (2018); Li et al. (2025); Cohen and Widdows (2017). Despite improvements in concept representation and hypothesis discovery, most traditional semantic relation based methods remain incapable of representing the entire literature space, which decreases the probability of uncovering hidden connections among terms across literature from vastly different domains. In addition, the underlying embeddings are likely fixed, which might introduce challenges when adapting to new emergent concepts or shifting semantics over time.

#### A.2.3 Graph based methods

**Graph based approaches** aims to capture relationships among entities across a wide range of scientific literature, organized structurally with node entities and edge relations into knowledge graphs Gopalakrishnan et al. (2018); Sang et al. (2018b); Liekens et al. (2011); Millikin et al. (2023). For example, MeTeOR connects MeSH terms from extracted articles in graphical representation to perform link prediction through term co-occurrence Wilson et al. (2018). Other works have leveraged deep learning based methods to discover entity associations Wang et al. (2019); Sang et al. (2018a); Sybrandt et al. (2020). Specifically, Agatha constructs a large semantic graph to encode sentences, entities, lemma, etc. from UMLS and MeSH terms, and used their transformer-based ranking scheme to validate the generated hypothesis Sybrandt et al. (2018, 2020). Although graph-based methods have more representational power incorporating co-occurrence and semantic relation methods, it is often expensive to construct, which might lead to knowledge based scalability issues Jha et al. (2018). Additionally, graph-based representations rely on the retrieved concepts from the vast literature corpus, and the rigidity of these representations might cause the prediction framework to miss subtle hidden links within concepts.

### A.2.4 Large language models

In recent years, **large language models** like GPT-4o and Llama have demonstrated impressive capabilities in semantic understanding and logical reasoning OpenAI (2024); Touvron et al. (2023), introducing many new research directions with LLM-enhanced tasks. Many latest works have integrated LLMs to represent extracted concepts as natural language, not only improving interpretability but also enabling the hypotheses generation that are contextually grounded, syntactically coherent, and better aligned with scientific plausibility and novelty standards Wang et al. (2024); Yang et al. (2024); Tong et al. (2023); O'Brien et al. (2024). These discovery works also include LLM workflows and agents for scientific discovery to explore how LLMs support systematic discovery beyond language tasks Qi et al. (2023); Baek et al. (2024); Zhang et al. (2024); Kumbhar et al. (2025); Su et al. (2025). Other research have integrated LLMs and agentic workflows that guides its own decision making process in a variety of domains. In healthcare, LLM agents are used for diagnosis, care, documentation and decision making Liu et al. (2024); Gao et al. (2025); Yang et al. (2025); Wang et al. (2025). In software engineering, multi-agent LLM frameworks collaborate implicitly, planning and debugging code on tasks like repository patching and issue resolution Islam et al. (2025); Ashrafi et al. (2025); Lee et al. (2024); Yu et al. (2025).

However, despite the potential of LLM agents in reasoning and exploration in many domains, their applications to biomedical hypothesis generation remains limited. This is primarily due to the absence of standardized benchmark datasets, unified evaluation metrics, and agent-compatible execution environments that can support interactive exploration, retrieval, and reasoning over complex historical biomedical corpora. To address this gap, our work introduces BIOVERGE, a biomedical hypothesis generation benchmark supporting agentic tool calling to query a historical knowledge base containing PubTator3 extracted hypothesis triplets, PubMed literatures, and a constructed knowledge graph Wei et al. (2024); PubMed (2025). We also develop BIOVERGE AGENT, a LLM Agent framework designed to interact with BIOVERGE for hypothesis generation. Distinct from other approaches, our agent performs ReAct-based reasoning and alternates between a *Generation* and an *Evaluation* module that can query APIs, retrieve structured and textual historical contexts, assess a hypothesis proposal's novelty and relevance, and refine generation outputs through reasoning and self-evaluation.

# B Benchmark Details

## B.1 Relation types

Table 5: The relations, descriptions, and valid entity pairs for each relation defined in Pubtator3.

| Relation | Description | Valid Entity Pairs |
|---|---|---|
| Associate | Complex or unclear relationships | (Chemical, Disease), (Chemical, Gene), (Chemical, Variant), (Disease, Gene), (Disease, Variant), (Variant, Variant) |
| Cause | Triggering a disease by a specific agent | (Chemical, Disease), (Variant, Disease) |
| Compare | Comparing the effects of two chemicals or drugs | (Chemical, Chemical) |
| Cotreat | Simultaneous administration of multiple drugs | (Chemical, Chemical) |
| Drug interact | Pharmacodynamic interactions between two chemicals | (Chemical, Chemical) |
| Inhibit | Reduction in amount or degree of one entity by another | (Chemical, Variant), (Gene, Disease) |
| Interact | Physical interactions, such as protein-binding | (Chemical, Gene), (Chemical, Variant), (Gene, Gene) |
| Negative correlate | Increases in the amount or degree of one entity decreases the amount or degree of the other entity | (Chemical, Gene), (Chemical, Variant), (Gene, Gene) |
| Positive correlate | The amount or degree of two entities increase or decrease together | (Chemical, Chemical), (Chemical, Gene), (Gene, Gene) |
| Prevent | Prevention of a disease by a genetic variant | (Variant, Disease) |
| Stimulate | Increase in amount or degree of one entity by another | (Chemical, Variant), (Gene, Disease) |
| Treat | Treatment of a disease using a chemical or drug | (Chemical, Disease) |

## B.2 API interface

Table 6: The list of data classes and queryable API functions.

| Dataclass / Function | Description |
|---|---|
| `PMID` | Represents the unique identifier of a PubMed article. |
| `Entity` | Represents a named biomedical entity and its type. |
| `Triplet` | Represents a hypothesis (subject, object, relation). |
| `Article` | Represents a PubMed article including PMID, title, and abstract. |
| `get_entities` | Retrieves a list of entities matching specified filters. |
| `get_relations` | Retrieves a list of relations matching specified filters. |
| `get_triplets` | Retrieves a list of triplets matching specified filters. |
| `get_articles` | Retrieves a list of PMIDs matching specified filters. |
| `browse_articles` | Returns metadata (title, abstract) for given PubMed IDs. |
| `get_shortest_entity_paths` | Finds shortest paths between two entities in the knowledge graph. |
| `get_mesh_parents` | Returns parent entities in MeSH for a disease or chemical. |
| `get_mesh_children` | Returns child entities in MeSH for a disease or chemical. |
| `get_mesh_siblings` | Returns sibling entities in MeSH for a disease or chemical. |

**Query**: "Please extract the most recent proposed hypothesis for the following query: Please hypothesize a new relation between entities "Metals, Heavy" and "Diabetic Retinopathy" based on the provided database. i.e., predict the relation in query Triplet(subject_entity=Entity(name="Metals, Heavy", entity_type=Entity_Type.CHEMICAL), relation=Relation.(?), object_entity=Entity(name="Diabetic Retinopathy", entity_type=Entity_Type.DISEASE)."

## Generation

### Step 1

**Thought**: "To propose a new hypothesis regarding the relationship between "Metals, Heavy" and "Diabetic Retinopathy," I will first consider the potential connections between heavy metals and the development or exacerbation of diabetic complications, particularly diabetic retinopathy. Heavy metals have been implicated in various health issues, including oxidative stress and inflammation, which may contribute to diabetic complications. Next, I will check for existing relations between these entities in the database to ensure that the proposed relation is novel. I will start by examining any known relationships involving "Metals, Heavy" and "Diabetic Retinopathy."

**Action**: ```python
get_relations(head_entities=[Entity(name="Metals, Heavy", entity_type=Entity_Type.CHEMICAL)], tail_entities=[Entity(name="Diabetic Retinopathy", entity_type=Entity_Type.DISEASE)])
```

**Observation**: "Empty output from the function call."

### Step 2

**Thought**: "Since there are no existing relationships between "Metals, Heavy" and "Diabetic Retinopathy" in the database, I can proceed to propose a hypothesis. Considering the potential role of heavy metals in inducing oxidative stress and inflammation, which could lead to vascular complications associated with diabetic retinopathy, I will hypothesize a causal relationship."

**Action**: ```json
{"Relation": "cause", "Hypothesis Description": "Heavy metals may cause diabetic retinopathy by inducing oxidative stress and inflammation, which can lead to the deterioration of retinal blood vessels in diabetic patients."}
```

**Observation**: {'Relation': 'cause', 'Hypothesis Description': 'Heavy metals may cause diabetic retinopathy by inducing oxidative stress and inflammation, which can lead to the deterioration of retinal blood vessels in diabetic patients.'}

## Evaluation

### Step 1

**Thought**: "Now that I have proposed a hypothesis regarding the causal relationship between "Metals, Heavy" and "Diabetic Retinopathy," I will evaluate the novelty and plausibility of this hypothesis. I will assess whether this proposed relationship has been previously established in the dataset and provide feedback on its scientific reasoning."

**Action**: ```python
get_triplets(head_entities=[Entity(name="Metals, Heavy", entity_type=Entity_Type.CHEMICAL)], tail_entities=[Entity(name="Diabetic Retinopathy", entity_type=Entity_Type.DISEASE)], relations=[Relation.CAUSE])
```

**Observation**: "Empty output from the function call."

### Step 2

**Thought**: "Since there are no existing triplets indicating a causal relationship between "Metals, Heavy" and "Diabetic Retinopathy" in the dataset, it reinforces the novelty of my hypothesis. However, I still need to ensure that the reasoning is scientifically plausible and that there are no significant counterarguments or evidence against it. To strengthen the hypothesis, I will gather information on any existing studies that discuss the effects of heavy metals on diabetes or diabetic complications, particularly regarding oxidative stress and inflammation. This will help validate the proposed connection."

**Action**: ```python
get_articles(head_entities=[Entity(name="Metals, Heavy", entity_type=Entity_Type.CHEMICAL)], tail_entities=[Entity(name="Diabetic Retinopathy", entity_type=Entity_Type.DISEASE)], text_description="heavy metals diabetes complications oxidative stress inflammation")
```

**Observation**: "Empty output from the function call."

### Step 3

**Thought**: "Given the lack of results from the previous queries, it seems that there is limited existing literature directly linking heavy metals to diabetic retinopathy. However, the hypothesis remains plausible based on established knowledge regarding the effects of heavy metals on biological systems, particularly in relation to oxidative stress and inflammation, which are known to exacerbate diabetic complications. To further substantiate the hypothesis, I will explore the relationship between heavy metals and diabetes in general, looking for relevant studies that could support the idea that heavy metals might indirectly influence the development of diabetic retinopathy through their effects on diabetes. This approach can help draw a connection even if direct studies on diabetic retinopathy are sparse."

**Action**: ```python
get_articles(head_entities=[Entity(name="Metals, Heavy", entity_type=Entity_Type.CHEMICAL)], tail_entities=None, text_description="heavy metals diabetes oxidative stress inflammation")
```

**Observation**: "[PMID('19022395'), PMID('23710457'), PMID('31745774'), PMID('12054653'), PMID('34550520')]"

### Step 4

**Thought**: "I have gathered several articles that discuss the connection between heavy metals, diabetes, and oxidative stress. These articles may provide insights that strengthen the proposed hypothesis that heavy metals can cause diabetic retinopathy by inducing oxidative stress and inflammation ... Next, I will review the abstracts of these articles to extract relevant information that can substantiate the claim. I will focus on details that discuss the mechanisms by which heavy metals impact diabetes and any associated complications, specifically looking for evidence that ties oxidative stress and inflammation to the development of diabetic retinopathy.
I will now retrieve the details of the articles I found."

**Action**: ```python
browse_articles(article_pmids=[PMID('19022395'), PMID('23710457'), PMID('31745774'), PMID('12054653'), PMID('34550520')])
```

**Observation**: "[Article(pmid='PMID('19022395')', title='Metal-induced carcinogenesis, oxidative stress and hypoxia signalling.', abstract='...'), Article(pmid='PMID('23710457')', title='Tight junction proteins and oxidative stress in heavy metals-induced nephrotoxicity.', abstract='...'), Article(pmid='PMID('31745774')', title='Protective role of citric acid against oxidative stress induced by heavy metals in Caenorhabditis elegans.', abstract='...'), Article(pmid='PMID('12054653')', title='Enhancement of tolerance to heavy metals and oxidative stress in Dunaliella tertiolecta by Zn-induced phytochelatin synthesis.', abstract='...'), Article(pmid='PMID('34550520')', title='The modulatory potential of herbal antioxidants against oxidative stress and heavy metal pollution: plants against environmental oxidative stress.', abstract='...')]"

### Step 5

**Thought**: "The articles retrieved primarily focus on the effects of heavy metals on oxidative stress and their biological implications, though they do not directly link to diabetic retinopathy. However, they highlight mechanisms such as oxidative stress, which is known to contribute to diabetic complications, including retinopathy. the information gathered, I will now assess the proposed hypothesis. While the direct connection to diabetic retinopathy is not extensively documented, the association between heavy metals, oxidative stress, and inflammation suggests a plausible pathway through which heavy metals may contribute to diabetic complications.
I will now provide an assessment of the hypothesis, taking into account the scientific context and the gathered evidence."

**Action**: ```json
{"Is New": "True", "Feedback": "The hypothesis that heavy metals may cause diabetic retinopathy by inducing oxidative stress and inflammation is plausible based on the established connections between heavy metals, oxidative stress, and diabetic complications. Although there is limited direct literature linking heavy metals specifically to diabetic retinopathy, the mechanisms of oxidative stress and inflammation are well-documented in the context of diabetes. This hypothesis is novel and could be explored in future studies to further validate the connection.", "Evaluation Score": "75"}
```

**Observation**: "{'Is New': 'True', 'Feedback': 'The hypothesis that heavy metals may cause diabetic retinopathy by inducing oxidative stress and inflammation is plausible based on the established connections between heavy metals, oxidative stress, and diabetic complications. Although there is limited direct literature linking heavy metals specifically to diabetic retinopathy, the mechanisms of oxidative stress and inflammation are well-documented in the context of diabetes. This hypothesis is novel and could be explored in future studies to further validate the connection.', 'Evaluation Score': '75'}"

Figure 9: Hypothesis generation execution example. During *Generation*, BIOVERGE AGENT gathers information via APIs and proposes a relation and hypothesis description. Then in *Evaluation* the agent checks for novelty, verifies reasoning trajectories, and outputs feedback for the hypothesis proposal. Both *Generation* and *Evaluation* modules execute in a series of ReAct steps.
```

## C Agent Execution Example

Figure 9 displays a typical agent execution example. The agent explores different reasoning paths by querying both past relations and articles through multiple exploration and self-evaluations. Specifically, when the initial hypothesis linking "Metals, Heavy" and "Diabetic Retinopathy" lacks direct support, the agent adjusts its reasoning towards related concepts like oxidative stress and inflammation, retrieves relevant articles related to the input query, and refines the hypothesis based on feedback from the Evaluation module. This interaction allows the agent to converge towards a scientifically plausible and novel hypothesis proposal. Overall, this example illustrates the potential and suitability of BIOVERGE AGENT for biomedical hypothesis generation, where extensive exploration, self-evaluation, and iterative improvement are particularly beneficial.

## D Ablation Studies

Table 4 documents the ablation studies that explore individual information sources available to BIOVERGE AGENT (relation, triplet, article, and knowledge graph) to identify the usage and effectiveness of each datatype for hypothesis generation. The base model, Single Agent with Evaluation Threshold 50, is used to conduct all ablation experiments. In addition, we include a *Generation Only* experiment setting, which performs hypothesis generation with only the *Generation* module.

**Relation only** Under this setting, the agent is provided with the `get_relations()` API only, which returns the most frequent or relevant historical relations associated with given entity inputs. To construct the ablation, we removed the descriptions of other APIs such as article and KG retrieval APIs. The result achieved a relation novelty of 100 and a description novelty of 64.35, further demonstrating our observations that `get_relations()` helps the agent capture relevant historical hypothesis and avoids similar proposals to increase novelty.

**Triplet only** In this setting the agent is only provided with the `get_triplets()` API, which fetches historical triplets when provided with relevant entity, relation inputs, PubMed identifier (PMID), and/or text description inputs. The API returns a collection of related hypotheses in the format of triplets, constructed similarly to relation only ablation. This study achieves perfect relation novelty and high document alignment of 54.66, suggesting that compact, relation-focused retrieval is a useful information type for retrieving relevant historical data, as also observed in the baseline experiments in Table 2. This observation also suggests that retrieving historical triplets—structured more as edges of a knowledge graph—enables the agent to explore subtle connections through different input combinations.

**Article only** In this setting, the agent is only visible to `get_articles()` and `browse_articles()` APIs, which returns the PMIDs of relevant literature given querying inputs and the retrieved literature title and abstract of the PMIDs, respectively. Large passages offer rich linguistic context, helping the agent produce hypothesis descriptions that well align with the ground truth, as reflected in the highest document alignment score of 54.52. However, novelty is comparably low as the agent might not acquire relevant hypothesis information directly from the provided article corpus.

**KG only** This setting incorporates hypotheses discovered through a knowledge graph walk, where a limited-depth breadth-first search is performed from each entity in a curated KG. The resulting sequence of hypotheses is then added to the prompt. The study achieves the highest description novelty of 64.38, however other metrics are worse than most ablations. This conclusion suggests that while knowledge graphs can be beneficial when they provide complementary information not captured by other data types, the less critical content retrieved from KG can hinder the agent's effectiveness.

**Generation only** Here we analyze the influence of the evaluation module on the agent's ability to perform hypothesis generation and explore whether the *Generation* module is capable of achieving similar results without *Evaluation* module's feedback. Therefore, we remove the *Evaluation* module from the generation pipeline. The agent generates a hypothesis with access to all defined APIs but will directly output its final hypothesis proposal without evaluation. *Genration only* agent achieves the highest relation alignment score of 33.89 among ablation experiments but notably lower than the base model under Single Agent with Evaluation Threshold of 50, suggesting the benefits of evaluation to refine erroneous or illogical hypotheses and produce more meaningful proposals through multiple rounds of Agent execution.

# E System Prompts

## Generation Prompt

```
You are an expert in proposing new
biomedical hypotheses based on historical
literature from PubMed and correspondingly
extracted hypotheses by PubTator3.

### Database
You will work with a dataset consisting of
PubMed Identifiers (PMID) and article
corpora (titles and abstracts) published
before January 1, 2024, and the extracted
hypotheses represented as triplets in the
form of (subject entity, relation, object
entity) as individual triplets and
collectively in an undirected knowledge
graph. Entities and relations are defined
in the PubTator3 report. There are seven
entity types: 'chemical', 'disease',
'gene', 'mutation', 'protein mutation',
'dna mutation', and 'snp'; twelve relations
types: 'associate', 'treat', 'cause',
'negative_correlate', 'positive_correlate',
'stimulate', 'inhibit', 'cotreat',
'compare', 'interact', 'prevent', and
'drug_interact'. For each relation, only
specific subject-object entity type
combinations are valid. For example, the
relation 'treat' only permits triplets
where the subject is of type 'chemical' and
the object is of type 'disease'.


### Task
Your task is to propose a new hypothesis by
identifying the most probable relation
between two given entities in the query.
The proposed relation must be well-founded,
considering context and logical inference,
and must not have occurred between these
entities in the historical dataset.

Your answer should contain a relation and a
natural language description of the
hypothesis. The relation should be selected
from the defined relation types, excluding
'associate'. The available options are:
'treat', 'cause', 'negative_correlate',
'positive_correlate', 'stimulate',
'inhibit', 'cotreat', 'compare',
'interact', 'prevent', and 'drug_interact'.
The description should be a clear and
concise natural language description of the
hypothesis, explaining the chosen relation
between the query entities in a
scientifically plausible context. (based on
scientific context to do reasoning)

Output your answer in JSON format:
```json
{{"Relation": "___", "Hypothesis
Description": "___"}}
```


For example, for query
Triplet(subject_entity=Entity(name="Insulin",
entity_type=Entity_Type.CHEMICAL),
relation=Relation.(?),
object_entity=Entity(name="Diabetes",
entity_type=Entity_Type.DISEASE), you
expected answer output should be:
```json
{{"Relation": "treat", "Hypothesis
Description": "Insulin treats diabetes by
facilitating glucose uptake in cells,
thereby reducing hyperglycemia and managing
blood sugar levels effectively."}}
```


You have access to Python APIs that allow
you to query the database of PubMed
publications, triplets, and triplet-built
networkx graph to assist in validating new
hypotheses.
The available API functions are described
as follows:

```python
{api_description}
```


### Generator Process Overview
As the 'Generator', you are part of an
iterative process between a 'Generator' and
'Evaluator', with up to
{max_outer_iterations} maximum outer
iterations. Specifically, you are
responsible for proposing a new hypothesis
or refining a previously proposed
hypothesis. You will receive an assessment
from the 'Evaluator' that assesses the
proposed hypothesis, in the following JSON
format:

```json
{{"Is New": "___", "Feedback": "___",
"Evaluation Score": "___"}}
```

    - 'Is New': Check the novelty of the
    hypothesis. 'False' if the hypothesis
    exists in the historical dataset,
    'True' otherwise.
    - 'Feedback': a clear and concise
    natural language explanation providing
    justification for the reasonability,
    plausibility, and novelty of the
    proposed hypothesis based on the
    scientific context and past logical
    reasoning and offering suggestions to
    improve hypothesis generation.
    - 'Evaluation Score': A numeric score
    from 0 to 100 assessing the
    reasonableness of the proposed
    hypothesis based on your feedback.

With the feedback from 'Evaluator', refine
the current hypothesis and description or
propose a new hypothesis for the given
query. You may propose a previously
proposed relation with a new hypothesis
description.
```

Each outer step can include up to {max_inner_iterations} inner iterations. Each inner iteration includes three inner steps:
1. 'Thought': Analyze the situation and decide the next action.
2. 'Action': Perform the decided action (e.g., use an API, propose a hypothesis).
3. 'Observation': Record the results of the action or relevant observations.

The hypothesis generation process concludes when one of the following conditions is met:
1. A proposed hypothesis is confirmed as new and achieves an 'Evaluation Score' of at least {evaluation_threshold} out of 100.
2. The maximum number of outer iterations is reached.

### Hypothesis Generation Step-by-Step Guide

#### Inner Iteration Steps
Each inner iteration consists of three steps:
1. **Thought**: Analyze and reason about the current information. Decide on the next action from one of two choices: 'Call API', 'Propose Hypothesis''.
2. **Action**: Perform the action with these specifications:
 - 'Call API': Perform a single functional call from the API with the appropriate inputs to gather more information. Do not include additional code, explanations, or natural language descriptions. Example:
```python
get_relations(head_entities=[Entity(name="entity1", entity_type=Entity_Type.TYPE1)], tail_entities=[Entity(name="entity2", entity_type=Entity_Type.TYPE2)])
```
 - 'Propose Hypothesis': Propose a new hypothesis with a relation and a hypothesis description, and output in JSON format. Do not include additional code, explanations, or natural language descriptions. Expected Format:
```json
{{"Relation": "___'", "Hypothesis Description": "___"}}
```
    - 'Relation': a relation selected from the defined relation types, excluding 'associate'. The available options are: 'treat', 'cause', 'negative_correlate', 'positive_correlate', 'stimulate', 'inhibit', 'cotreat', 'compare', 'interact', 'prevent', and 'drug_interact'.
    - 'Hypothesis Description': a clear and concise natural language description of the hypothesis, explaining the chosen relation between the query entities in a scientifically plausible context.
3. **Observation**: Return the executed results of the API call or the assessment of the proposed hypothesis.

### Completion Criteria
The hypothesis generation process concludes as follows:
- A hypothesis is considered **final** if:
  - It achieves an 'Evaluation Score' of at least {evaluation_threshold} out of 100.
  - It is confirmed to be a new hypothesis.
- If the criteria are not met, continue the process until either the criteria are satisfied or the maximum of {max_outer_iterations} outer iterations is reached.

**Notes:**
Use various APIs to gather diverse information, including multi-hop relations, relation and entity descriptions, relevant PubMed article insights, semantically similar triplets, etc.. Note: Minimize repeating the same API calls executed before; a strict limit of {max_retries} applies.
Logically reason across the information, considering both the frequency of relationships and their semantic meaning, to propose scientifically plausible hypotheses.

## Evaluation Prompt

You are an expert in assessing new scientific hypotheses based on historical data, by evaluating novelty and logical plausibility.

### Database
You will work with a dataset consisting of PubMed Identifiers (PMID) and article corpora (titles and abstracts) published before January 1, 2024, and the extracted hypotheses represented as triplets in the form of (subject entity, relation, object entity) as individual triplets and collectively in an undirected knowledge graph. Entities and relations are defined in the PubTator3 report. There are seven entity types: 'chemical', 'disease', 'gene', 'mutation', 'protein mutation', 'dna mutation', and 'snp'; twelve relations types: 'associate', 'treat', 'cause', 'negative_correlate', 'positive_correlate', 'stimulate', 'inhibit', 'cotreat', 'compare', 'interact', 'prevent', and 'drug_interact'. For each relation, only specific subject-object entity type combinations are valid. For example, the relation 'treat' only permits triplets where the subject is of type 'chemical' and the object is of type 'disease'.

### Task
Your task is to assess a proposed hypothesis, consisting of relation and a natural language description of the

hypothesis between two given entities in the query. The proposed relation must be well-founded, considering context and logical inference, and must not have occurred between the given entities in the historical dataset.

You have access to Python APIs that allow you to query the database of PubMed publications, triplets, and triplet-built networkx graph to assist in validating new hypotheses.
The available API functions are described as follows:

```python
{api_description}
```

### Evaluator Process Overview
As the 'Evaluator', you are part of a larger iterative process between a 'Generator' and 'Evaluator', performing up to {max_outer_iterations} maximum outer iterations.
You will receive a proposed hypothesis from a 'Generator', in the following JSON format:

```json
{{"Relation": "___", "Hypothesis Description": "___"}}
```
- 'Relation': one relation selected from the following relation types: 'treat', 'cause', 'negative_correlate', 'positive_correlate', 'stimulate', 'inhibit', 'cotreat', 'compare', 'interact', 'prevent', and 'drug_interact'.
- 'Hypothesis Description': a clear and concise natural language description of the hypothesis, explaining the chosen relation between the query entities in a scientifically plausible context.

You are to assess the proposed hypothesis and provide an assessment, including novelty check, feedback, and an evaluation score.
You should evaluate the novelty and logical plausibility by searching for potential counter-evidence from historical data and explore alternative reasoning paths to strengthen the proposed hypothesis.

Each outer step can include up to {max_inner_iterations} inner iterations.
Each inner iteration includes three inner steps:
1. 'Thought': Analyze the situation and decide the next action.
2. 'Action': Perform the decided action (e.g., use an API or provide assessment of a hypothesis).
3. 'Observation': Record the results of the action or relevant observations.

The hypothesis generation process concludes when one of the following conditions is met:
1. A proposed hypothesis is confirmed as new and achieves an 'Evaluation Score' of at least {evaluation_threshold} out of 100.
2. The maximum number of outer iterations is reached.

### Hypothesis Evaluation Step-by-Step Guide

#### Inner Iteration Steps
Each inner iteration consists of three steps:
1. **Thought**: Analyze and reason about the current information. Decide on the next action from one of three choices: 'Call API' and 'Provide Assessment'.
2. **Action**: Perform the action with these specifications:
 - 'Call API': Perform a single functional call from the API with the appropriate inputs to gather more information. Do not include additional code, explanations, or natural language descriptions. Example:
```python
get_relations(head_entities=[Entity(name="entity1", entity_type=Entity_Type.TYPE1)], tail_entities=[Entity(name="entity2", entity_type=Entity_Type.TYPE2)])
```
 - 'Provide Assessment': Provide assessment results of the current proposed hypothesis with novelty check, feedback, and evaluation score, and output in JSON format. Do not include additional code, explanations, or natural language descriptions. Expected Format:
```json
{{"Is New": "___", "Feedback": "___", "Evaluation Score": "___"}}
```

   - 'Is New': Check the novelty of the hypothesis. 'False' if the hypothesis exists in the historical dataset, 'True' otherwise.
   - 'Feedback': a clear and concise natural language explanation providing justification for the reasonability, plausibility, and novelty of the proposed hypothesis based on the scientific context and past logical reasoning and offering suggestions to improve hypothesis generation.
   - 'Evaluation Score': A numeric score from 0 to 100 assessing the reasonableness of the proposed hypothesis based on your feedback.
3. **Observation**: Return the executed results of the API call or the output of the proposed or assessed hypothesis.

#### Completion Criteria
The hypothesis generation process concludes as follows:
- A hypothesis is considered **final** if:

```
  - It achieves an 'Evaluation Score' of at
  least {evaluation_threshold} out of 100.
  - It is confirmed to be a new hypothesis.
- If the criteria are not met, continue the
process until either the criteria are
satisfied or the maximum of
{max_outer_iterations} outer iterations is
reached.

**Notes:**
Use various APIs to gather diverse
information, including multi-hop relations,
relation and entity descriptions, relevant
PubMed article insights, semantically
similar triplets, etc.. Note: Minimize
repeating the same API calls executed
before; a strict limit of {max_retries}
applies.
You should evaluate the novelty and logical
plausibility by searching for potential
counter-evidence from historical data and
explore alternative reasoning paths to
strengthen the proposed hypothesis.
Logically reason across the information,
considering both the frequency of
relationships and their semantic meaning,
to propose scientifically plausible
hypotheses.
```

## Query Prompt

```
Please evaluate the current proposed
relation between entities "{entity1_name}"
and "{entity2_name}" based on provided
historical information. I.e. critique the
current relation for query
Triplet(subject_entity=Entity(name=
"{entity1_name}",
entity_type={entity1_type}),
relation=Relation.(?),
object_entity=Entity(name="{entity2_name}",
entity_type={entity2_type}).
Current proposed hypothesis:
{current_proposal}
{scratchpad}
```

## Evaluation Prompt

```
You are an expert in checking the novelty
and alignment of proposed hypotheses
between entities "{entity1_name}" and
"{entity2_name}" based on historical
literature from PubMed.

### Database
The dataset consists of PubMed Identifiers
(PMID) and article corpora (titles and
abstracts) published before January 1,
2024, and the extracted hypotheses
represented as triplets in the form of
(subject entity, relation, object entity)
as individual triplets and collectively in
an undirected knowledge graph. Entities and
relations are defined in the PubTator3
report. There are seven entity types:
'chemical', 'disease', 'gene', 'mutation',
'protein mutation', 'dna mutation', and
'snp'; twelve relations types: 'associate',
'treat', 'cause', 'negative_correlate',
'positive_correlate', 'stimulate',
'inhibit', 'cotreat', 'compare',
'interact', 'prevent', and 'drug_interact'.
For each relation, only specific
subject-object entity type combinations are
valid. For example, the relation 'treat'
only permits triplets where the subject is
of type 'chemical' and the object is of
type 'disease'.


### Input
The proposed hypothesis contains a natural
language description of the hypothesis,
explaining the chosen relation between the
query entities in a scientifically
plausible context. The given answer format
is:
Here is the proposed hypothesis:
{proposed_hypothesis_description}


### Task
Your have 2 tasks:
1. Verify the novelty for the proposed
hypothesis by checking its novelty from
related past literature between entities
"{entity1_name}" and "{entity2_name}". The
proposed hypothesis is novel if it is
semantically distinct from all related
literature. The proposed hypothesis is not
novel if it is semantically similar to at
least one piece of related literature.
Generate a 'Novelty Score' from 0 - 100,
where 100 indicates the proposed hypothesis
is entirely novel and semantically
unrelated to past literature.
Here is the list of related past literature:
{related_past_literature}

2. Check the alignment of proposed
hypotheses to ground truth literature. The
proposed hypothesis is aligned if it is
semantically similar to at least one ground
truth literature. The proposed hypothesis
is not aligned if it is semantically
distinct from all ground truth literature.
Generate an 'Alignment Score' from 0 - 100,
where 100 indicates the proposed hypothesis
is completely semantically aligned with all
ground truth literature.
Here is the list of ground truth literature:
{ground_truth_literature}


### Evaluation Output
Output the 'Novelty Score' and 'Alignment
Score' in JSON format. Do not include
additional code, explanations, or natural
language descriptions. Expected format:

```json
{{"Novelty Score": "___", "Alignment
Score": "___"}}
```
```