
GRAPH-NATIVE REINFORCEMENT LEARNING ENABLES TRACEABLE SCIENTIFIC HYPOTHESIS GENERATION THROUGH CONCEPTUAL RECOMBINATION

 **Subhadeep Pal**

Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
Cambridge, MA, USA

 **Shashwat Sourav**

Department of Physics
Washington University in St. Louis
St. Louis, MO, USA
Computing and Computational Sciences Directorate
Oak Ridge National Laboratory
Oak Ridge, TN, USA
Lawrence Berkeley National Laboratory
Berkeley, CA, USA

 **Tirthankar Ghosal**

Computer Science and Mathematics Division
Computing and Computational Sciences Directorate
Oak Ridge National Laboratory
Oak Ridge, TN, USA

 **Markus J. Buehler***

Department of Civil and Environmental Engineering
Department of Mechanical Engineering
Schwarzman College of Computing
Massachusetts Institute of Technology
Cambridge, MA, USA

*Corresponding author: mbuehler@MIT.EDU

ABSTRACT

Accelerating materials discovery requires AI systems that can generate scientifically valid hypotheses through multi-step, domain-grounded reasoning. Standard large language models often produce fluent but weakly traceable responses to open-ended materials design problems, making it difficult to determine whether final answers are supported by coherent intermediate reasoning. We develop Graph-PRefLexOR, a family of graph-native reasoning models fine-tuned with Group Relative Policy Optimization (GRPO) to organize reasoning into explicit phases for mechanism exploration, graph construction, pattern extraction, and hypothesis synthesis. This design links neural language generation with symbolic relational structure, enabling causal connections to be constructed, inspected, and reused. On 100 open-ended questions from materials science and mechanics literature, Graph-PRefLexOR achieves 40-65% improvements over corresponding base models, with the largest gains in reasoning traceability. Embedding analyses show broader semantic exploration and approximately $2\times-3\times$ greater semantic diversity than baselines. Semantic backtracking and layer-wise hidden-state analyses further show stronger alignment between structured reasoning and final answers. Finally, test-time graph expansion reveals that additional compute primarily increases long-range conceptual recombination within a bounded semantic space, rather than simply expanding semantic coverage. These results establish graph-native reinforcement learning as a pathway toward interpretable AI systems for scientific hypothesis generation in materials design and other scientific applications.

Keywords Graph-native reasoning, scientific hypothesis generation, reinforcement learning, materials design, large language models

1 Introduction

Scientific discovery increasingly depends on the ability to connect concepts, mechanisms, and evidence across domains that are often studied in isolation. This challenge is especially pronounced in materials science and mechanics, where macroscopic properties emerge from coupled processes spanning molecular structure, mesoscale organization, interfaces, defects, processing history, and boundary conditions [1, 2, 3, 4]. Generating useful hypotheses in such settings requires more than retrieving relevant facts or summarizing prior work; it requires organizing relationships among entities, mechanisms, constraints, and outcomes. Yet scientific knowledge remains fragmented across papers, disciplines, terminologies, and modeling frameworks, leaving many potentially important connections implicit or difficult to evaluate [5]. The central problem, therefore, is how to build AI systems that can transform dispersed scientific information into interpretable reasoning structures capable of supporting cross-domain hypothesis generation.

Large language models (LLMs) offer a promising substrate for this task because much of scientific knowledge is encoded in text, and LLMs can summarize literature, compare concepts, generate explanations, and assist in hypothesis formulation [6, 7, 8, 9, 10]. Recent AI-for-science workflows have extended these capabilities toward materials discovery, protein design, code generation, autonomous experimentation, and scientific agents [11, 12, 13, 14, 15, 16]. However, most LLM reasoning remains expressed as linear text. Chain-of-thought and related prompting strategies can improve performance by exposing intermediate reasoning, but these traces often lack an explicit representation of the entities, relations, dependencies, and causal links that organize scientific explanations [17, 18, 19]. This limitation is consequential for materials reasoning, where hypotheses often require connecting concepts across different domains and understanding failure modes at multiple scales. A textual trace may describe such links, but it does not directly encode which concepts are connected, what type of relation connects them, or how local mechanisms compose into a higher-level explanation.

Prior approaches have begun to address parts of this problem. Retrieval-augmented generation improves access to relevant documents, knowledge graphs encode entities and relations explicitly, and agentic workflows decompose complex scientific tasks across specialized components [20, 21, 22, 23, 24, 25, 26]. Graph representations are particularly attractive because they make scientific relationships inspectable: concepts can be represented as nodes, while mechanisms, causal dependencies, analogies, constraints, or failure pathways can be represented as directed edges [27, 28]. Nevertheless, existing systems often use graphs as external retrieval substrates or post hoc representations rather than as the model’s native reasoning format. Similarly, agentic systems may exchange artifacts or natural-language messages without forcing the underlying reasoning to become relational, parseable, or reusable. Thus, while existing methods improve access, orchestration, and organization of scientific information, they do not fully solve the problem of making the model’s intermediate reasoning itself graph-structured and evaluable.

Here, we address this gap with Graph-PRefLexOR, a graph-native reasoning model for open-ended scientific hypothesis generation [29, 30]. Building on earlier Graph-PRefLexOR formulations that mapped a task to a knowledge graph, abstract patterns, and a final answer, the present model exposes a more granular sentinel-based reasoning trace consisting of `<brainstorm>`, `<graph>`, `<graph_json>`, `<patterns>`, and `<synthesis>` phases. These stages separate mechanism exploration, concept abstraction, machine-readable graph construction, higher-order pattern identification, and final hypothesis synthesis. We train the model using Group Relative Policy Optimization (GRPO) [31], a reinforcement-learning method that improves model behavior through relative comparisons among groups of generated outputs rather than relying on fixed target answers alone. In this setting, GRPO encourages the model to generate reasoning traces that are not merely fluent but also structured, relational, and traceable. This differs from our earlier ORPO/EXO-style preference-optimization approaches by directly optimizing the model toward higher-quality reasoning behavior through group-relative rewards, making it well suited for open-ended scientific problems where multiple plausible hypotheses may exist.

We evaluate Graph-PRefLexOR on a manually curated benchmark of 100 open-ended questions derived from materials science and mechanics literature. The benchmark probes cross-domain linkage, causal mapping, hidden-variable identification, model abstraction, and hypothesis generation capabilities that are difficult to assess using standard factual or multiple-choice evaluations. Across 1.7B, 3B, and 8B model scales, Graph-PRefLexOR outperforms its corresponding base models in reasoning quality, intellectual depth, and reasoning traceability, with the strongest improvements observed in traceability. Embedding-based analyses further show that Graph-PRefLexOR reasoning traces occupy more organized semantic regions, follow more directional trajectories, and exhibit approximately $2\times$ – $3\times$ greater semantic diversity than baseline traces. Semantic backtracking and layer-wise hidden-state analyses show that final answers remain strongly aligned with the structured reasoning pathway, particularly the `<synthesis>` stage, indicating improved continuity between intermediate reasoning and response generation. We further show that the graph-native format supports test-time graph expansion: by accumulating emitted graph structures into a growing memory graph, additional inference-time compute produces statistically novel long-range conceptual recombinations within a bounded semantic space. Together, these results indicate that graph-native GRPO improves scientific reasoning

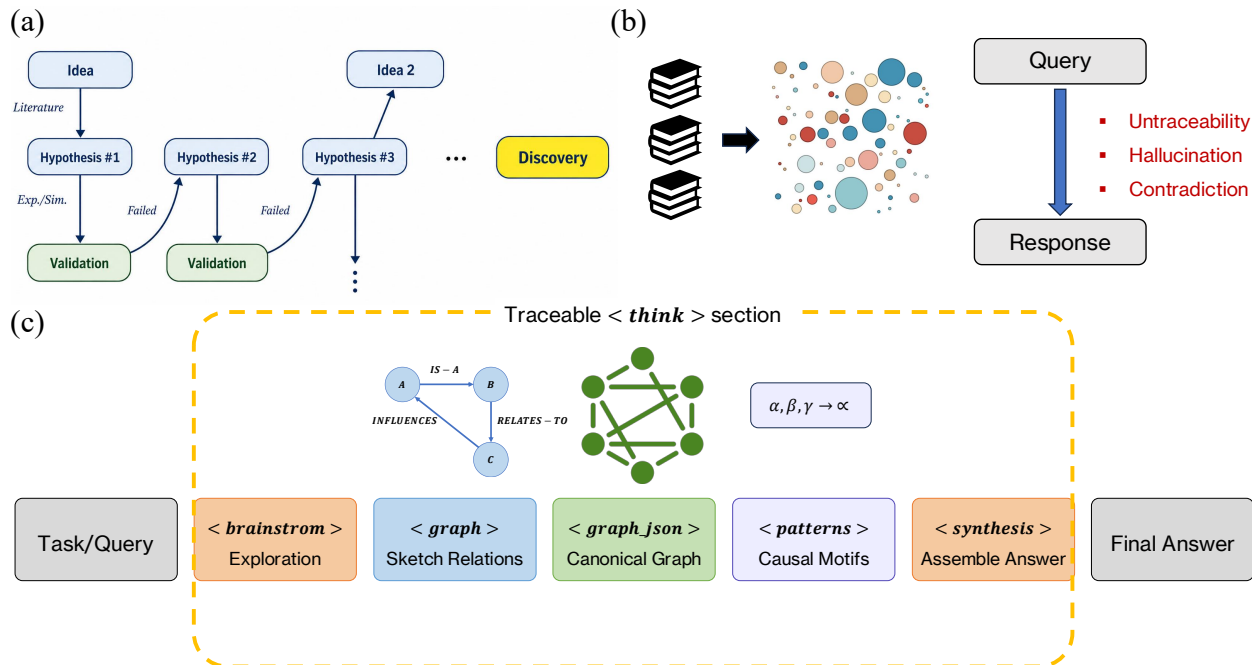


Figure 1: (a) Scientific discovery often proceeds through iterative hypothesis generation, validation, re-ideaation and refinement. (b) Standard LLM responses to scientific queries can be difficult to trace, leading to untraceability, hallucination, or contradiction. (c) Graph-PRefLexOR addresses this limitation by organizing the `<think>` section into explicit reasoning phases: `<brainstorm>` for mechanism exploration, `<graph>` for relation sketching, `<graph_json>` for canonical graph construction, `<patterns>` for causal motif extraction, and `<synthesis>` for assembling the final answer.

not simply by changing final answers, but by restructuring the intermediate computational pathway through which hypotheses are generated, aligned, and iteratively recombined.

2 Results and Discussion

2.1 Evaluation of Structured Reasoning

One of the primary objectives of this study is to quantitatively assess the quality of reasoning traces generated by the graph-native Group Relative Policy Optimization (GRPO) model compared to its corresponding base models [30, 29, 31]. To this end, we develop three GRPO-based variants, collectively termed Graph-PRefLexOR, at scales of 8B, 3B, and 1.7B parameters. These models are initialized from Qwen3-8B [32], Llama-3.2-3B-Instruct [33], and Qwen3-1.7B [32], respectively, and subsequently fine-tuned to perform structured, multi-stage reasoning.

Specifically, the models are trained to organize their internal reasoning into a sequence of structured phases within the thinking process: `<brainstorm>`, `<graph>`, `<graph_json>`, `<patterns>`, and `<synthesis>`. The `<brainstorm>` phase performs divergent exploration, generating candidate mechanisms, hypotheses, and potential failure modes grounded in domain knowledge. The `<graph>` phase then abstracts this reasoning into a conceptual representation by identifying core entities (e.g., sequence, structure, processing, properties) and their causal relationships, forming an interpretable reasoning scaffold. This abstraction is formalized in the `<graph_json>` phase as a machine-readable directed graph, where nodes correspond to entities and edges encode typed relationships (e.g., "source": "Sequence", "relation": "encodes", "target": "Structure"), drawn from a predefined relational vocabulary [27]. The `<patterns>` phase subsequently extracts higher-order regularities from this graph, such as causal chains (e.g., sequence \rightarrow structure \rightarrow properties \rightarrow failure), scale-bridging relationships, and feedback loops that capture recurring mechanistic motifs. Finally, the `<synthesis>` phase integrates these patterns into a coherent, testable hypothesis, explicitly linking multi-scale mechanisms to predicted outcomes and potential failure modes.

To evaluate the performance gains achieved by the GRPO-based models, we construct an open-ended benchmark comprising 100 questions designed to probe cross-disciplinary linkage, causal mapping, and hypothesis generation capabilities, derived from published materials science and mechanics literature. Additional details regarding dataset construction are provided in the Methods section. We perform inference across six models with reasoning-enabled

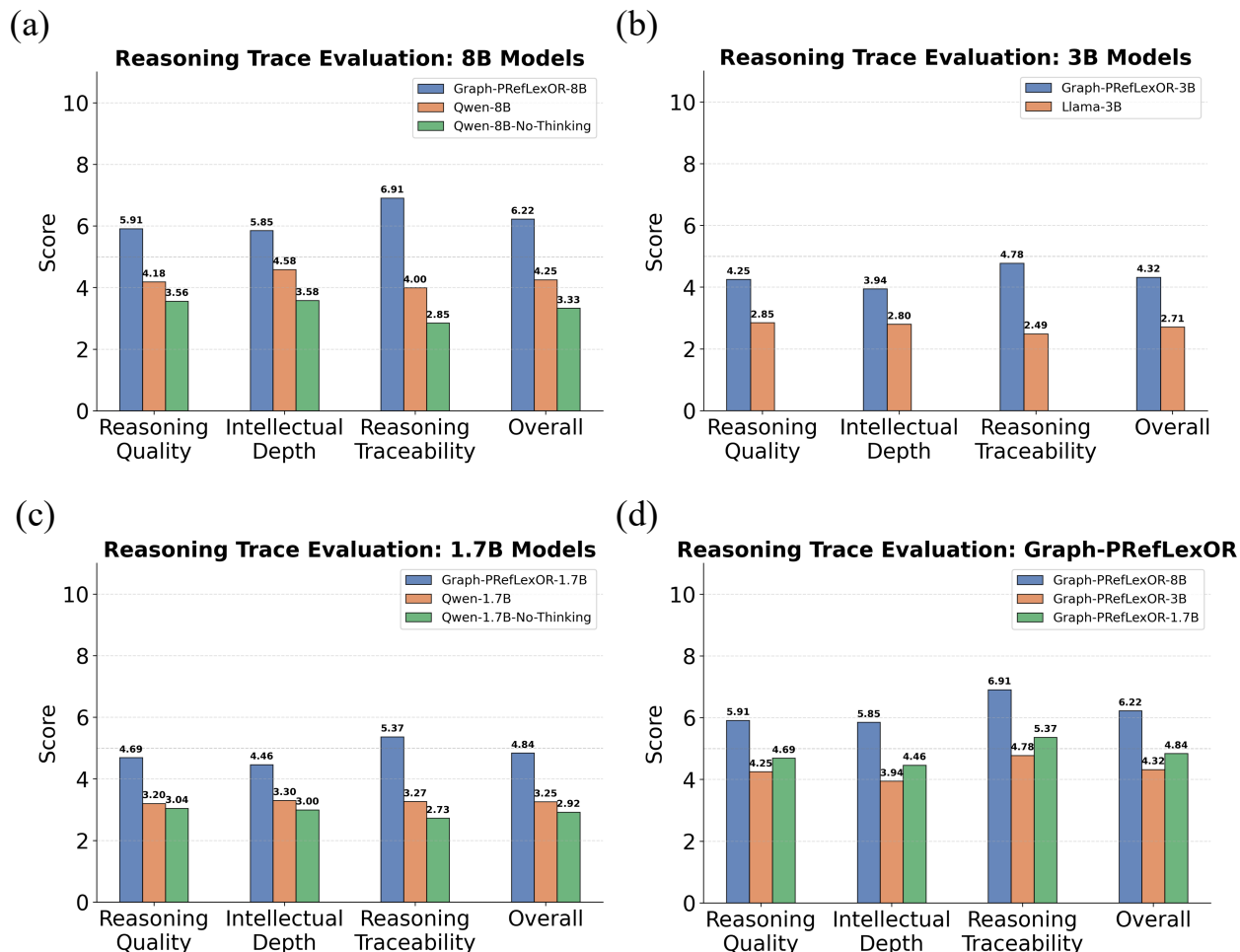


Figure 2: Evaluation of structured reasoning across model scales on open-ended scientific questions ($N = 100$), assessed using Claude Opus-4.7. Metrics include Reasoning Quality, Intellectual Depth, Reasoning Traceability, and Overall score (0–10). (a) Graph-PRefLexOR-8B vs. Qwen3-8B (with no-thinking variant), (b) Graph-PRefLexOR-3B vs. Llama-3.2-3B-Instruct, and (c) Graph-PRefLexOR-1.7B vs. Qwen3-1.7B (with no-thinking variant). Across all scales, Graph-PRefLexOR consistently outperforms its base models on all metrics, with the largest gains in reasoning traceability, indicating improved structural organization and causal transparency in hypothesis generation from research articles.

decoding (except for the Llama-3.2-3B-Instruct baseline), allowing each model to explicitly generate intermediate reasoning during the `<think>` phase.

Because the final responses are conditioned on this internal reasoning process, we evaluate the quality of the generated `<think>` traces directly. For this purpose, we define three complementary evaluation metrics: *Reasoning Quality*, *Intellectual Depth*, and *Reasoning Traceability*, along with an aggregate *Overall Score* computed as their mean. Each metric is scored on a 0–10 scale, and evaluation is performed using Claude *opus-4.7* as an independent judge [34, 35]. Notably, as OpenAI GPT models are employed during dataset generation, we use Claude-based evaluation to mitigate potential model-family bias. Collectively, these metrics capture distinct dimensions of reasoning performance, enabling a structured assessment of both the correctness and organization of the generated reasoning traces.

Figure 2 summarizes the performance of the Graph-PRefLexOR models relative to their corresponding base models across the three evaluation metrics. Across all model scales, the GRPO-trained graph models consistently outperform their baselines, exhibiting improvements of approximately 40–65% in aggregate performance, with the largest gains observed in *Reasoning Traceability*. This trend indicates that structured, graph-native reasoning primarily enhances the model’s ability to construct mechanistic and causally grounded explanations, rather than merely improving surface-level coherence. Moreover, summarizing core concepts using a directed knowledge graph further enriches reasoning quality [27].

For the Llama-3.2-3B-Instruct baseline, evaluation is restricted to the final response, as the model lacks an explicit reasoning phase. To further isolate the contribution of structured reasoning, we additionally evaluate Qwen-based

Biological immune systems and multi-agent AI frameworks both address open-ended problems through division of labor among specialized units with distinct roles and communication protocols. In biological systems, B-cells, T-cells, and memory cells coordinate to detect, respond to, and adapt against novel pathogens, while in multi-agent AI systems, planner, executor, and critic agents collaborate to generate, evaluate, and refine solutions to complex design tasks. Construct a structured mapping between these two systems by identifying correspondences between their components, communication pathways, adaptation mechanisms, and feedback loops. Then analyze where this analogy breaks down at a mechanistic level, particularly in terms of learning, memory formation, and long-term adaptation. Using this gap, propose one concrete hypothesis for a capability that next-generation multi-agent scientific systems would need to develop in order to achieve the adaptive, long-horizon robustness observed in biological immune systems, and explain the mechanism by which this capability would operate.

Figure 3: Representative cross-disciplinary hypothesis-generation question used to evaluate Graph-PRefLexOR. The question is derived from Ref. [10] and probes analogical mapping, mechanistic breakdown, and long-horizon adaptive reasoning.

models with reasoning disabled (*no-thinking* setting). The resulting performance degradation closely mirrors that observed for the Llama baseline, with overall reductions on the order of 30–50%, confirming that the majority of the observed gains arise from the explicit reasoning process rather than architectural differences alone.

A second key trend emerges with respect to model scale and base model characteristics. As shown in Fig. 2d, the 8B Graph-PRefLexOR model consistently outperforms its smaller counterparts, achieving approximately 25–30% higher scores than the 1.7B variant across all metrics. This improvement is primarily attributable to increased representational capacity, which enables more expressive and coherent graph construction and pattern extraction. Furthermore, the 3B model, initialized from Llama-3.2-3B-Instruct, underperforms relative to the 1.7B Qwen-based model. This behavior suggests that the absence of an inherent reasoning scaffold in the base model limits the effectiveness of GRPO-based structured reasoning.

2.2 Qualitative Comparison of Reasoning Traces

To provide a qualitative comparison of model behavior, we examine representative responses from Graph-PRefLexOR-8B and Qwen3-8B to the same benchmark question. Figure 3 shows an example question used to evaluate open-ended reasoning quality. The question is derived from the multi-agent protein-design framework of Ghafarollahi and Buehler [10], but is written to be self-contained and does not require direct knowledge of the original paper. The question asks the model to construct an analogy between two superficially distinct hierarchical systems: biological immune response and multi-agent AI coordination. Specifically, it requires the model to identify correspondences among specialized components, communication pathways, adaptation mechanisms, and feedback loops. It then asks the model to analyze where the analogy breaks down mechanistically, particularly with respect to learning, memory formation, and long-term adaptation, and to use this gap to propose a concrete future capability for next-generation multi-agent scientific systems. This structure makes the question a useful probe of cross-domain mapping, mechanistic reasoning, limitation analysis, and hypothesis generation.

Figure 4 shows the corresponding Graph-PRefLexOR-8B reasoning trace. The model decomposes its reasoning into explicit phases, beginning with divergent mechanism exploration in `<brainstorm>`, followed by entity–relation abstraction in `<graph>`, higher-order pattern extraction in `<patterns>`, and final hypothesis integration in `<synthesis>`. This structure makes the reasoning trace directly inspectable: the analogy is first explored conceptually, then converted into mapped components and relations, and finally synthesized into a mechanistic hypothesis. In this example, the model identifies *adaptive memory expansion* as a candidate capability for long-horizon robustness in multi-agent AI systems, motivated by biological mechanisms such as clonal selection, affinity maturation, and persistent immune memory.

The graph and pattern representations extracted from the same response are shown in Fig. 5. The directed graph encodes key immune-system concepts, multi-agent AI components, and proposed bridging mechanisms as nodes connected by typed relations. Biological immune-system concepts are grouped separately from multi-agent AI concepts, while the proposed bridging mechanism links adaptive memory expansion to clonal selection and long-term robustness. The pattern panel further compresses the graph into higher-order motifs, such as division of labor → specialization → coordination → adaptation and learning → memory → robustness. These motifs reveal how Graph-PRefLexOR converts a long-form reasoning trace into a compact relational structure that can be inspected and reused.

In contrast, the Qwen3-8B baseline produces a substantially longer linear response that contains relevant concepts but lacks an explicit relational scaffold. Because the Qwen3-8B response is substantially longer than the Graph-PRefLexOR trace, only its qualitative behavior is summarized in the main text; the full baseline response is provided in Supplementary Information. The response proceeds through extended prose, repeated uncertainty, and backtracking before arriving at a final conclusion. While such a response can still identify useful factors, its intermediate reasoning is less compact, less parseable, and less directly organized around entities, relations, and reusable causal motifs.

```

<brainstorm>
Biological immune systems (B-cells, T-cells, memory cells) and multi-agent AI (planner, executor, critic) share high-level parallels: division of labor (specialized roles), communication (chemical signals/protocols), adaptation (response to novel threats/solutions), feedback loops (evaluation/refinement). Key concepts: B-cells produce antibodies (planner generates hypotheses), T-cells activate responses (executor implements), memory cells store past encounters (critic evaluates and stores knowledge). Communication: cytokines (chemical signals) vs. message passing (explicit protocols). Adaptation: clonal selection (evolution of better agents) vs. reinforcement learning (policy updates). Feedback: antigen-presenting cells (APCs) bridge B/T cells (planner- critic interface) vs. critic feedback to planner. Hypotheses: Mapping holds at architectural level but breaks mechanistically in learning (B-cells learn via affinity maturation, T-cells via MHC presentation), memory (long-lived cells vs. AI memory decay), long-term robustness (evolutionary adaptation vs. training limitations). Breakdown: AI lacks biological-like learning (e.g., no clonal expansion), memory is brittle, no innate diversity. Hypothesis: Next-gen systems need "adaptive memory expansion" via dynamic agent proliferation and mutation, mimicking clonal selection, enabling long-horizon robustness through self-organized evolution of specialized agents. Variables: specialization level, communication fidelity, adaptation speed, memory persistence, learning diversity. Interactions: specialization → efficient division of labor → faster adaptation; communication → coordination → feedback loops; memory → long-term robustness; learning diversity → innovation in novel problems. Key insight: Biological systems excel in open-ended, long-horizon problems due to self-organized evolution and memory; AI needs analogous self-organized learning mechanisms.
</brainstorm>

<graph>
Core entities: B-cells (planner), T-cells (executor), Memory cells (critic), Antigen-presenting cells (planner-critic interface), Cytokines (communication), Clonal selection (adaptation), Affinity maturation (learning), Long-term memory (memory formation), Multi-agent AI (planner-agent), Executor-agent, Critic-agent, Message passing (communication), Reinforcement learning (adaptation), Policy updates (learning), Memory decay (memory breakdown), Adaptive memory expansion (hypothesis). Relationships: B-cells → produce antibodies (planner generates solutions); T-cells → activate responses (executor implements); Memory cells → store past encounters (critic evaluates and stores); Antigen-presenting cells → bridge B/T cells (planner-critic interface); Cytokines → coordinate immune response (communication); Clonal selection → adapts B/T cells (adaptation); Affinity maturation → enhances B-cell learning (learning); Long-term memory → persists in memory cells (memory formation); Planner-agent → generates hypotheses (planner); Executor-agent → executes plans (executor); Critic-agent → evaluates outcomes (critic); Message passing → coordinates agents (communication); Reinforcement learning → updates policies (adaptation); Policy updates → refine solutions (learning); Memory decay → weakens AI memory (breakdown); Adaptive memory expansion → mimics clonal selection (hypothesis); Adaptive memory expansion → enables long-term robustness (hypothesis).
</graph>

<patterns>
1. Division of labor → specialization → coordination → adaptation: B-cells (planner) → T-cells (executor) → Memory-cells (critic) via APCs (interface) → cytokines (communication) → clonal selection (adaptation); Planner-agent → Executor-agent → Critic-agent via message passing → reinforcement learning → policy updates. 2. Learning → memory → robustness: Affinity-maturation (learning) → Long-term-memory (memory) → robustness; Policy-updates (learning) → Memory-decay (breakdown) → Adaptive-memory-expansion (repair).
</patterns>

<synthesis>
The graph reveals a coherent high-level analogy: both systems use specialized agents (B/T vs. planner/executor/critic) with communication (cytokines vs. message passing) to coordinate division of labor, enabling adaptation through feedback loops (APCs/planner-critic interface). Patterns show specialization → coordination → adaptation as a core loop, with learning/memory as enablers of long-term robustness. However, the analogy breaks mechanistically: biological systems achieve learning via affinity maturation (B-cells evolve antibodies) and clonal selection (T-cells via MHC), while AI relies on brittle policy updates and memory decay. Hypothesis: Next-gen systems need "adaptive memory expansion" (dynamic agent proliferation/mutation), mimicking clonal selection, to enable self-organized evolution of specialized agents, overcoming AI's limitations in open-ended, long-horizon problems.
</synthesis>

```

Figure 4: Representative Graph-PRefLexOR-8B reasoning to the benchmark question in Fig. 3. The response is organized into structured reasoning phases, including <brainstorm>, <graph>, <patterns>, and <synthesis>. This format exposes the intermediate pathway from analogical mapping to mechanistic gap identification and final hypothesis generation.

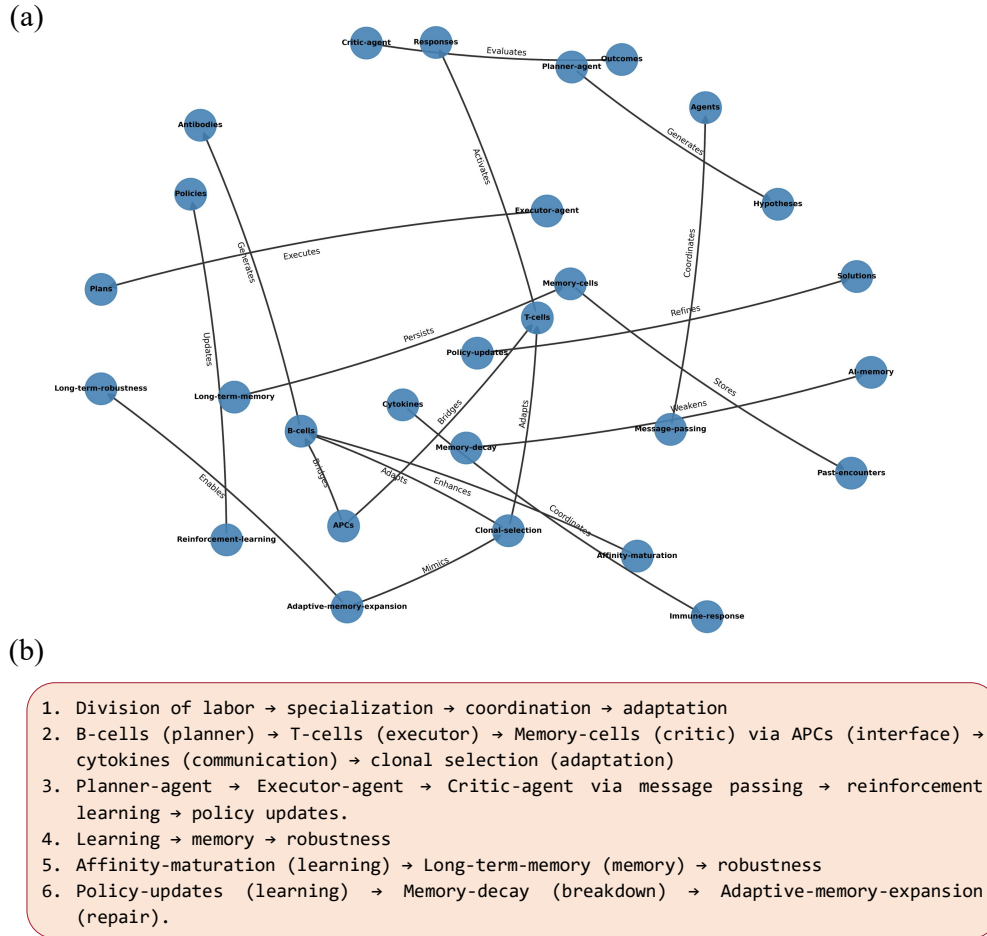


Figure 5: Graph and pattern representation extracted from the Graph-PRefLexOR-8B response. (a) Directed graph linking biological immune-system concepts, multi-agent AI components, and the proposed bridging mechanism. (b) Higher-order reasoning patterns extracted from the graph, summarizing the main causal motifs used for hypothesis synthesis.

This qualitative comparison motivates the embedding-based analyses that follow, where we quantify whether these visible structural differences correspond to measurable changes in semantic organization, trajectory directionality, and reasoning diversity.

2.3 Geometry of Reasoning Representations

2.3.1 Semantic Organization

To further characterize the semantic structure of the generated reasoning traces, we project both intermediate reasoning steps and final answers into a shared embedding space using the `google/embeddinggemma_300m` model, which maps text to 768-dimensional vectors [36]. For each model, we embed both the reasoning traces (i.e., `<think>` content) and the corresponding final answers. The reasoning traces are segmented into atomic reasoning steps, where each step is defined as a complete sentence without paragraph breaks. This results in an $n \times 768$ representation for each sample, where n denotes the number of reasoning steps.

We then apply Principal Component Analysis (PCA) to project these embeddings into a two-dimensional space for visualization [37]. A key observation is that baseline models consistently generate longer reasoning traces, resulting in a larger number of reasoning steps compared to the GRPO-trained models. Direct point-wise visualization would therefore bias the comparison toward baseline models due to their higher sampling density. To address this, we estimate the underlying distribution of reasoning steps using Gaussian kernel density estimation (KDE) and visualize the resulting density as contour plots over a shared embedding grid [38]. This enables a distributional comparison that is invariant to the number of reasoning steps.

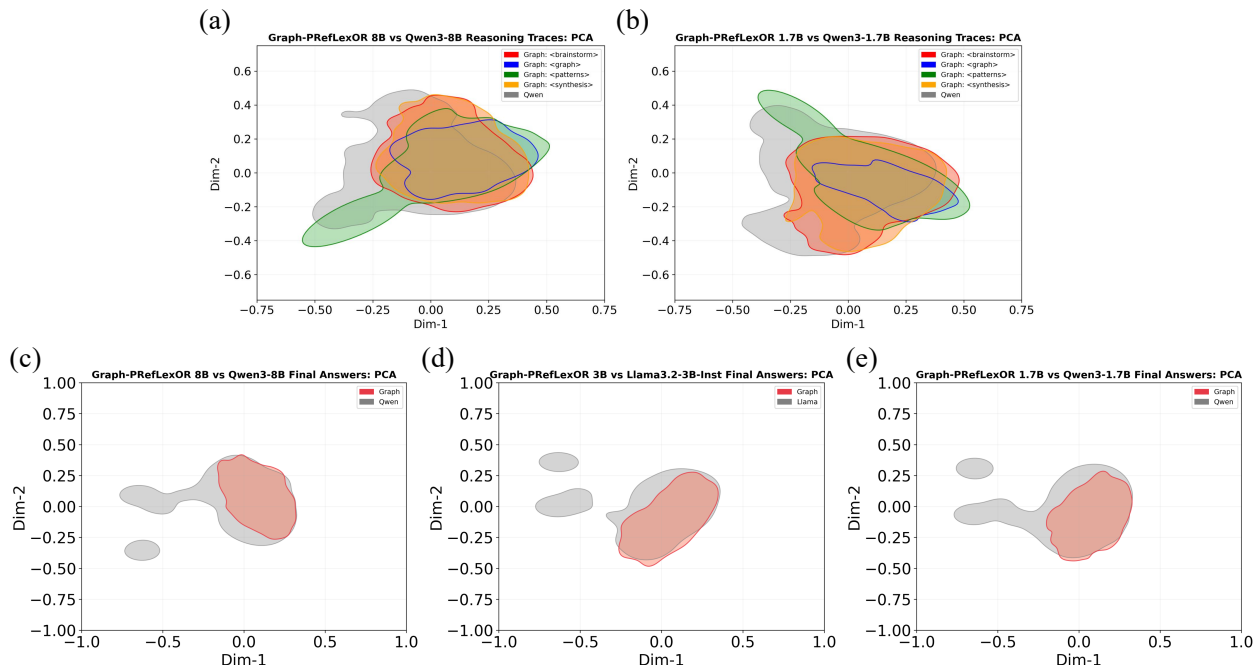


Figure 6: PCA projection of reasoning traces and final answers comparing Graph-PRefLexOR and base models across scales. (a) Graph-PRefLexOR-8B vs. Qwen3-8B reasoning traces, (b) Graph-PRefLexOR-1.7B vs. Qwen3-1.7B reasoning traces, and (c–e) corresponding comparisons of final answers for 8B, 3B, and 1.7B models, respectively. Reasoning traces are decomposed into structured components (<brainstorm>, <graph>, <patterns>, and <synthesis>), revealing more organized and separable distributions relative to baseline models. In contrast, final answer representations exhibit tighter clustering and greater alignment between Graph-PRefLexOR and base models, indicating that performance gains arise primarily from improved intermediate reasoning structure rather than differences in final outputs.

For reasoning trace analysis, we focus only on the 8B and 1.7B models to isolate the effects of scale while maintaining comparable base architectures. The GRPO-generated reasoning steps are further partitioned according to their structured phases: <brainstorm>, <graph>, <patterns>, and <synthesis>, and KDE is applied separately to each phase to reveal their semantic organization. In contrast, baseline models are represented as a single aggregated distribution. For final answer analysis, we apply the same embedding and KDE procedure across all three model scales, treating each answer as a single semantic distribution without phase decomposition.

Figure 6 reveals several key trends in the semantic organization of reasoning traces and final outputs. First, the reasoning traces of Graph-PRefLexOR exhibit a more structured, separable distribution across the embedding space than those of baseline models. The phase-wise decomposition yields distinct yet partially overlapping regions, indicating that each phase occupies a specialized semantic subspace while contributing to a coherent reasoning trajectory. In contrast, baseline models form a single, diffuse distribution.

Second, the Graph-PRefLexOR distributions span a broader and more directional manifold, particularly evident in the <brainstorm> and <patterns> phases, which extend into regions not covered by the baseline models. This suggests enhanced exploration of the hypothesis space and the emergence of higher-order abstractions that are not captured by standard autoregressive reasoning.

Third, despite these substantial differences in intermediate reasoning representations, the final answer embeddings exhibit significantly tighter clustering and strong overlap between Graph-PRefLexOR and baseline models across all scales (Fig. 6c–e). This convergence indicates that both model classes converge on semantically similar endpoints, even though their underlying reasoning trajectories differ markedly. However, the baseline final answer embeddings exhibit multiple separated clusters in the PCA space, suggesting fragmentation in semantic representations across samples. In contrast, Graph-PRefLexOR produces a more compact and unified distribution, indicating greater consistency in the final answer space. Collectively, these observations support the hypothesis that the primary advantage of Graph-PRefLexOR lies in the organization and expressivity of intermediate reasoning, rather than in substantial shifts in the final answer distribution.

2.3.2 Directed Trajectories

The PCA analysis indicates that Graph-PRefLexOR occupies broader and more structured regions of the latent embedding space than the corresponding base models. However, this distributional view captures only where the

representations lie, not how the model moves through semantic space during reasoning and answer generation. To examine this dynamic aspect, we represent each response as an ordered trajectory in the PCA-projected embedding space. For Graph-PRefLexOR, reasoning trajectories are constructed from the four structured reasoning phases: <brainstorm>, <graph>, <patterns>, and <synthesis>. Final answers are instead divided into four equal-length sequential chunks. For baseline models, which lack explicit phase annotations, both reasoning traces and final answers are partitioned into four equal-length sequential chunks. Consecutive phases or chunks are then connected by directed arrows, enabling visualization of how each model traverses semantic space over the course of reasoning and response generation.

This formulation enables a direct comparison between structured and unstructured reasoning dynamics in latent space. In Graph-PRefLexOR, trajectories correspond to transitions between functionally distinct reasoning operations, revealing how the model progressively reorganizes semantic representations throughout the reasoning process. As shown in Fig. 7a, the trajectory initially shifts from the <brainstorm> phase toward the <graph> phase, reflecting a transition from divergent hypothesis generation to the abstraction of key entities and their relationships. The subsequent <patterns> phase remains closer to the graph representation, consistent with its role in identifying higher-order causal and symbolic relationships among the extracted concepts. Finally, the <synthesis> phase integrates the hypotheses, concepts, and patterns into a coherent mechanistic explanation and partially returns toward the semantic region occupied by <brainstorm>, suggesting that synthesis reconnects structured abstractions with the original hypothesis space. Collectively, these transitions produce broad, directional movements through the embedding space, indicating a richer and more organized reasoning process.

In contrast, Qwen3-8B trajectories remain comparatively localized and entangled, with sequential chunks exhibiting substantial overlap and limited directional separation. This behavior is consistent with unstructured reasoning patterns such as backtracking, self-correction, and repetition, which compress the trajectories into narrower regions of latent space. Consequently, baseline trajectories resemble generic sequential continuation through generated text rather than transitions between functionally distinct reasoning states.

For answer trajectories (Fig. 7b), a similar but less pronounced trend is observed. Because final answers are divided into four equal, sequential chunks in both models, the trajectories reflect the evolution of the semantic content of the response during answer generation. Graph-PRefLexOR exhibits a broader early displacement, with the second chunk moving furthest from the first, followed by subsequent chunks that remain comparatively closer in semantic space. This suggests that the final answer initially expands or reframes the response before consolidating into a more stable explanatory structure.

In contrast, Qwen3-8B trajectories remain more compact throughout the answer-generation process, with stronger overlap among sequential chunks and limited directional separation. Thus, although both models show greater convergence in final answer space than in reasoning-trace space, Graph-PRefLexOR retains a more pronounced pattern of early semantic diversification followed by synthesis. Overall, these observations reinforce that Graph-PRefLexOR induces more structured and directional semantic evolution, whereas the baseline model exhibits comparatively localized and less differentiated trajectories.

2.4 Quantifying Semantic Diversity

The trajectory analysis provides qualitative evidence that Graph-PRefLexOR generates richer and more differentiated reasoning representations. To complement this visualization with a quantitative measure, we compute centroid-based semantic distances in the original 768-dimensional embedding space. For reasoning traces, Graph-PRefLexOR embeddings are grouped according to their structured phases. For each response, we compute the centroid of each phase and evaluate all six pairwise cosine distances between phase centroids. These distances are then averaged to obtain a single response-level measure of semantic diversity across reasoning stages.

For baseline models, which lack explicit phase annotations, each reasoning trace is partitioned into four approximately equal sequential chunks and treated as pseudo-phases as before. The same centroid-distance calculation is applied, enabling a matched comparison between structured and unstructured reasoning. This analysis is performed for the 1.7B and 8B model pairs. We apply an analogous procedure to final answers across all model scales (1.7B, 3B, and 8B), dividing each answer into four equal sequential chunks for both Graph-PRefLexOR and the corresponding baseline model. The resulting distributions are visualized using violin plots with jittered samples, median indicators, and mean \pm standard deviation overlays.

Figure 8 shows that Graph-PRefLexOR consistently exhibits higher semantic diversity than its corresponding base models across both reasoning traces and final answers. For reasoning traces (Fig. 8a), the mean inter-phase cosine distance increases from 0.07 to 0.20 at 1.7B and from 0.08 to 0.21 at 8B, corresponding to $2.9\times$ and $2.6\times$ gains,

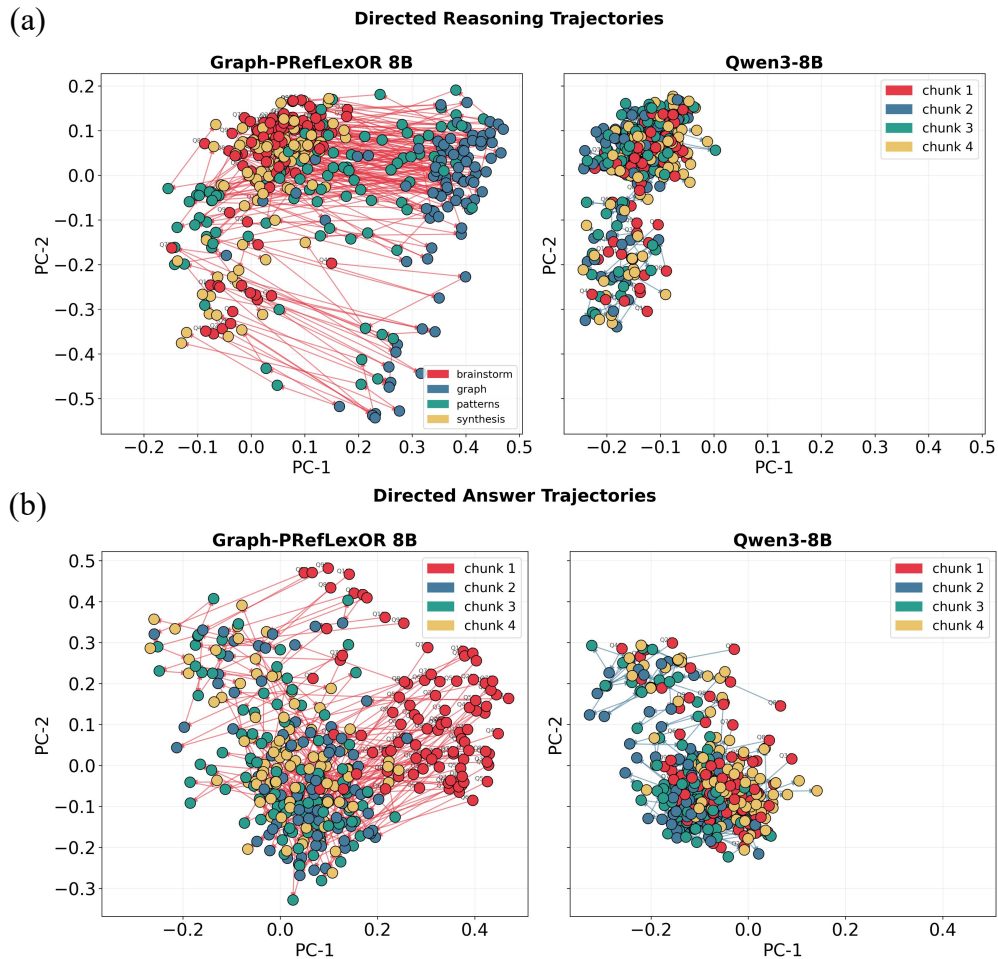


Figure 7: PCA projection of directed (a) reasoning, and (b) answer trajectories between Graph-PRefLexOR-8B and Qwen3-8B. For Graph-PRefLexOR, trajectories explicitly follow structured stages (<brainstorm>, <graph>, <patterns>, and <synthesis>), forming coherent, directional transitions in latent space. In contrast, base model trajectories (shown as sequential chunks) remain more localized and less structured. For answer trajectories, four sequential chunks are used for both models.

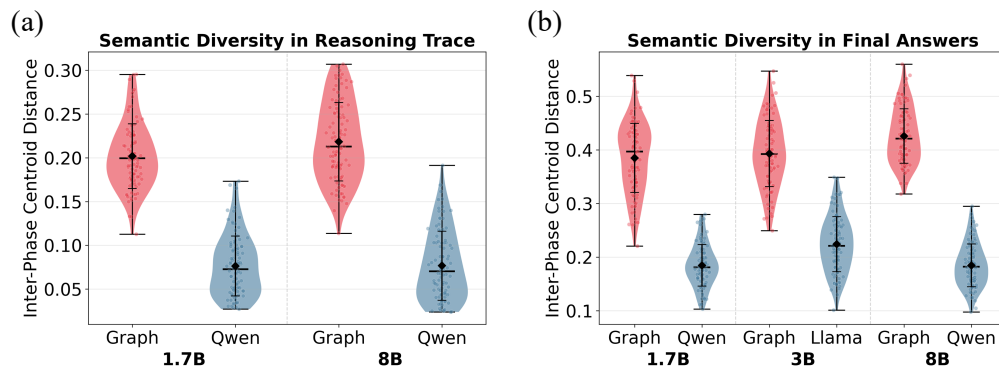


Figure 8: Semantic diversity measured via inter-phase centroid distance for (a) reasoning traces and (b) final answers across model scales. Violin plots show the distribution of sample-level semantic diversity scores for Graph-PRefLexOR and the corresponding base models. Individual points denote responses, horizontal black lines indicate medians, black diamonds denote means, and vertical error bars represent one standard deviation. Graph-PRefLexOR consistently exhibits higher inter-phase distances across all scales, indicating greater semantic separation between reasoning stages and increased diversity in generated representations. This effect is more pronounced in reasoning traces and remains evident, though reduced, in final answers.

Table 1: Mean semantic diversity and relative gains of Graph-PRefLexOR over corresponding base models.

Output Type	Model Scale	Graph-PRefLexOR	Base Model	Absolute Increase	Gain
Reasoning trace	1.7B	0.20	0.07	0.13	2.9×
Reasoning trace	8B	0.21	0.08	0.13	2.6×
Final answer	1.7B	0.39	0.18	0.21	2.2×
Final answer	3B	0.39	0.22	0.17	1.8×
Final answer	8B	0.43	0.18	0.25	2.4×

respectively (Table 1). These larger distances indicate stronger semantic differentiation among reasoning stages, consistent with the structured decomposition of reasoning into distinct functional phases.

For final answers (Fig. 8b), Graph-PRefLexOR again maintains higher semantic diversity, although the relative separation is smaller than in the reasoning traces. The mean inter-chunk distance increases from 0.18 to 0.39 at 1.7B, from 0.22 to 0.39 at 3B, and from 0.18 to 0.43 at 8B, corresponding to 2.2×, 1.8×, and 2.4× gains, respectively (Table 1). Thus, Graph-PRefLexOR induces approximately 2-3× greater semantic improvement across model scales, with the strongest effect occurring during intermediate reasoning. This indicates structured reasoning broadens semantic exploration during computation while partially converging during final answer synthesis.

2.5 Semantic Backtracking

The preceding analyses show that Graph-PRefLexOR produces more structured and semantically differentiated reasoning traces than the corresponding base model. We next examine whether the final answer remains aligned with the model’s own intermediate reasoning. This question is related to prior work on chain-of-thought faithfulness, which asks whether a model’s stated reasoning actually supports its final answer [17, 19, 39]. More broadly, it follows the view that model outputs should be assessed by both final-answer quality, and the extent to which they are supported by appropriate intermediate evidence or explanations [40]. To test this alignment, we perform a semantic backtracking analysis on the same 100-question benchmark. For each response, we embed the final answer and compare it against candidate reference texts using cosine similarity. The reference with the highest similarity is treated as the closest semantic source of the final answer. For Qwen3-8B, the candidate sources are its own <think> trace, Graph-PRefLexOR-8B’s final answer, and Graph-PRefLexOR-8B’s <brainstorm>, <graph>, <patterns>, and <synthesis> phases. For Graph-PRefLexOR-8B, we perform the symmetric comparison using its own structured reasoning stages, Qwen3-8B’s thinking trace, and Qwen3-8B’s final answer.

Figure 9a shows a clear asymmetry between the two models. Qwen3-8B final answers are closest to their own thinking traces in only 16 of 100 cases. In the remaining cases, they align more closely with Graph-PRefLexOR-8B-derived outputs, most frequently with Graph-PRefLexOR-8B’s final answer itself, which is the closest reference in 46 cases. Additional cases align with Graph-PRefLexOR-8B’s <brainstorm> phase in 14 cases, <graph> phase in 13 cases, <synthesis> phase in 8 cases, and <patterns> phase in 3 cases (See Figure 9b). This indicates that Qwen3-8B can produce final answers that occupy a semantic region similar to Graph-PRefLexOR-8B outputs, even when its own visible reasoning trace is not the closest semantic precursor. In contrast, Graph-PRefLexOR-8B final answers remain strongly anchored to its own structured reasoning pathway. In the cross-model comparison, Graph-PRefLexOR-8B final answers are closest to one of its own reasoning phases in 92 of 100 cases. The dominant source is <synthesis>, which is closest in 84 cases, whereas only 8 cases align more closely with Qwen3-8B outputs (See Figure 9c). This suggests that Graph-PRefLexOR-8B maintains a tighter semantic connection between intermediate reasoning and final-answer generation. To further isolate the internal structure of this alignment, we compare each Graph-PRefLexOR-8B final answer only against its own reasoning stages: <brainstorm>, <graph>, <patterns>, and <synthesis>.

As shown in Fig. 10a, Graph-PRefLexOR 8B final answers are closest to the <synthesis> stage in 89 of 100 cases, compared with 9 cases for <brainstorm> and 2 cases for <graph>. The mean similarity analysis shows the same trend, with <synthesis> exhibiting the highest average cosine similarity to the final answer, followed by <brainstorm>, <graph>, and <patterns>. This is consistent with the intended role of <synthesis> as the final integrative stage, where candidate mechanisms, graph-encoded relations, and extracted patterns are consolidated into answer-ready prose. Together, these results indicate that Graph-PRefLexOR-8B provides stronger reasoning-answer alignment than Qwen3-8B. While Qwen3-8B often reaches final answers that are semantically close to Graph-PRefLexOR 8B outputs, its own thinking trace is rarely the closest semantic source. By contrast, Graph-PRefLexOR-8B final answers are consistently grounded in its structured reasoning pathway, especially the <synthesis> stage. This supports the conclusion that graph-native reasoning improves not only the quality of intermediate reasoning traces, but also the coherence between reasoning and final response generation.

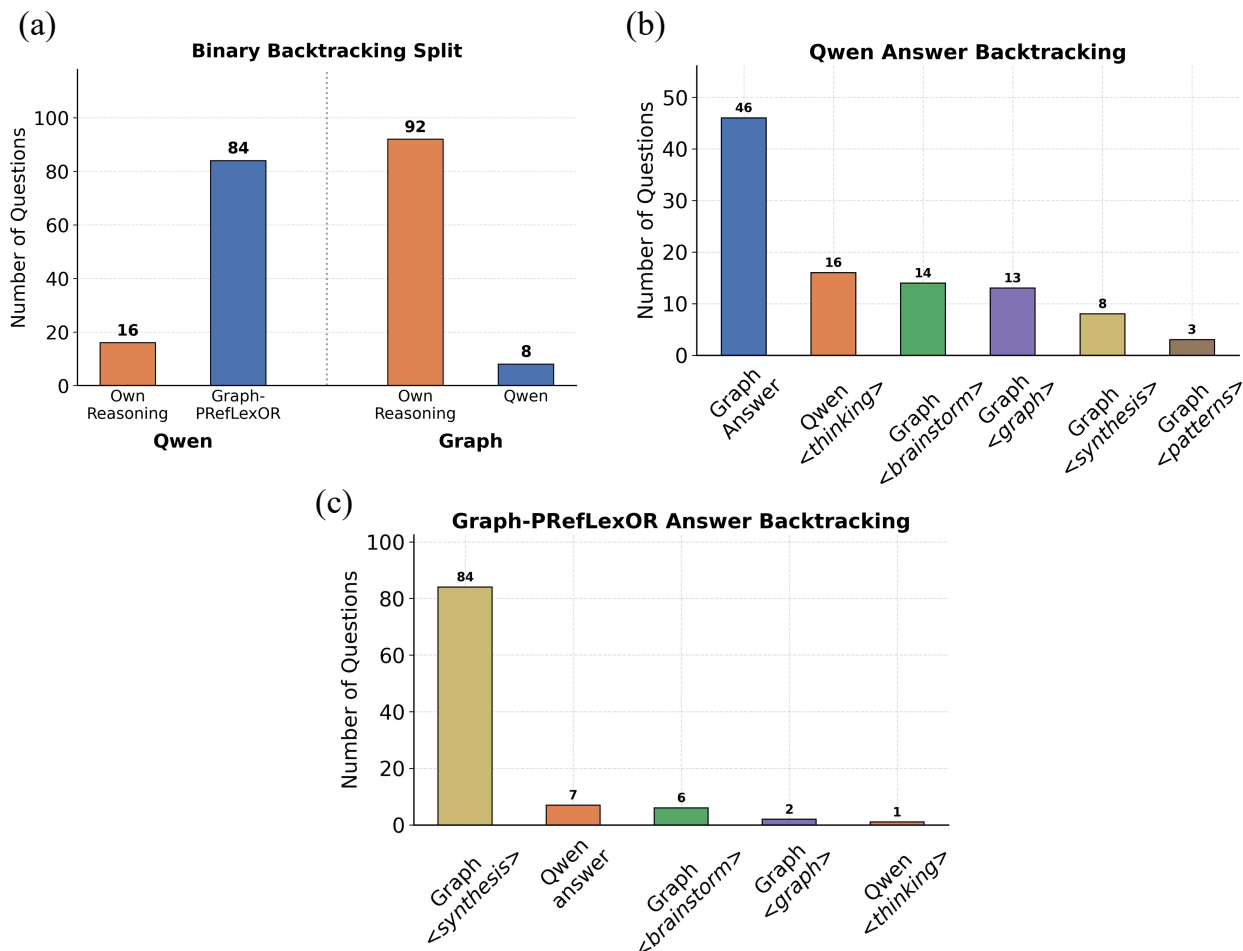


Figure 9: Semantic backtracking analysis of final answer alignment for Qwen3-8B and Graph-PRefLexOR 8B across 100 open-ended scientific questions. (a) Binary split showing whether each final answer is closest to its own reasoning trace or to the other model’s outputs. (b) Source distribution for Qwen3-8B final answers, which align with its own <think> trace in only 16/100 cases and more often align with Graph-PRefLexOR outputs. (c) Source distribution for Graph-PRefLexOR-8B final answers, which align with its own structured reasoning stages in 92/100 cases, predominantly the <synthesis> stage.

2.6 Layer-Wise Reasoning-Answer Divergence

We next examine whether the semantic alignment observed at the embedding level is reflected in the models’ internal hidden-state representations. Layer-wise hidden-state analysis is commonly used to study how transformer representations evolve across depth [41, 42], while representation-similarity methods provide a general framework for comparing neural states across layers and conditions [43, 44]. This analysis is particularly relevant here because visible reasoning traces may not always faithfully support the final answer [19, 39, 45]. For Qwen3-8B, we compute the cosine distance between hidden states averaged over thinking tokens and hidden states averaged over final-answer tokens at each transformer layer. For Graph-PRefLexOR-8B, we compute the analogous distance between hidden states from the structured reasoning trace and those from the final answer. This yields a layer-wise measure of how strongly each model separates intermediate reasoning from final response generation.

Figure 11 shows that Qwen3-8B exhibits a substantially larger gap between thinking-state and answer-state representations than Graph-PRefLexOR-8B. The divergence increases sharply around layers 7-10 and rises again at the final layer, suggesting that the baseline model undergoes a stronger representational shift between visible reasoning and final answer generation. By contrast, Graph-PRefLexOR-8B maintains a smaller reasoning-answer distance across most layers, consistent with a more continuous transition from structured intermediate reasoning to the final response. To connect this layer-wise behavior with the semantic backtracking results, we further quantify examples according to whether the final answer backtracks to the model’s own reasoning. For Qwen3-8B, we divide the benchmark into cases where the final answer is closest to its own <think> trace and cases where it is closest to another reference source. For Graph-PRefLexOR-8B, we analogously separate cases where the final answer aligns with its own structured reasoning stages from cases where it aligns with Qwen-derived outputs.

Graph-PRefLexOR Internal Backtracking

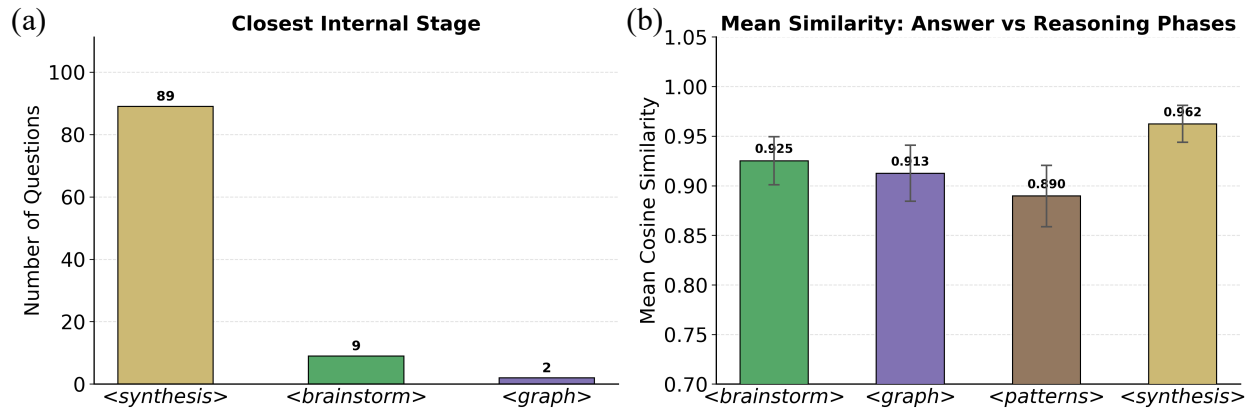


Figure 10: Internal semantic backtracking of Graph-PRefLexOR-8B final answers. (a) Closest structured reasoning stage for each final answer across 100 benchmark questions. (b) Mean cosine similarity between the final answer and each reasoning phase. Final answers align most frequently and most strongly with the <synthesis> stage, indicating that response generation is primarily grounded in the final integrative reasoning step.

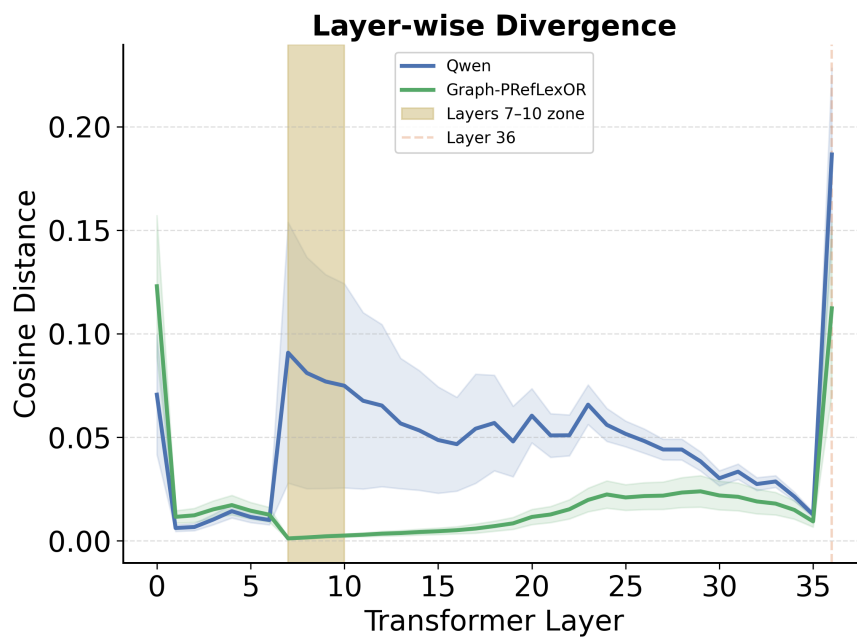


Figure 11: Layer-wise hidden-state divergence between reasoning and final-answer representations for Qwen3-8B and Graph-PRefLexOR-8B. Qwen3-8B exhibits a larger reasoning-answer separation, with a pronounced increase around layers 7-10 and a final-layer spike. In contrast, Graph-PRefLexOR-8B maintains lower divergence across most layers, indicating a more continuous transition from structured reasoning to final-answer generation. Shaded regions denote \pm one standard deviation across questions.

Figure 12a shows that the Qwen3-8B reasoning-answer gap depends strongly on backtracking behavior. When the final answer backtracks to Qwen’s own thinking trace, hidden-state divergence is lower. When it does not, the divergence is larger, with the clearest separation again emerging around layers 7-10 and at the final layer. This suggests that layers 7-10 mark an early transition region where the final-answer pathway begins to separate from the visible thinking trace. We discuss in detail about this in the Supplementary Information. Graph-PRefLexOR 8B shows a more stable pattern across backtracking groups (See Figure 12b). Most examples backtrack to the model’s own structured reasoning stages, and both groups maintain lower divergence than Qwen3-8B. This indicates that Graph-PRefLexOR-8B preserves a tighter representational connection between intermediate reasoning and final-answer generation, even when examples are stratified by alignment source. Together, these hidden-state analyses support the semantic backtracking results.

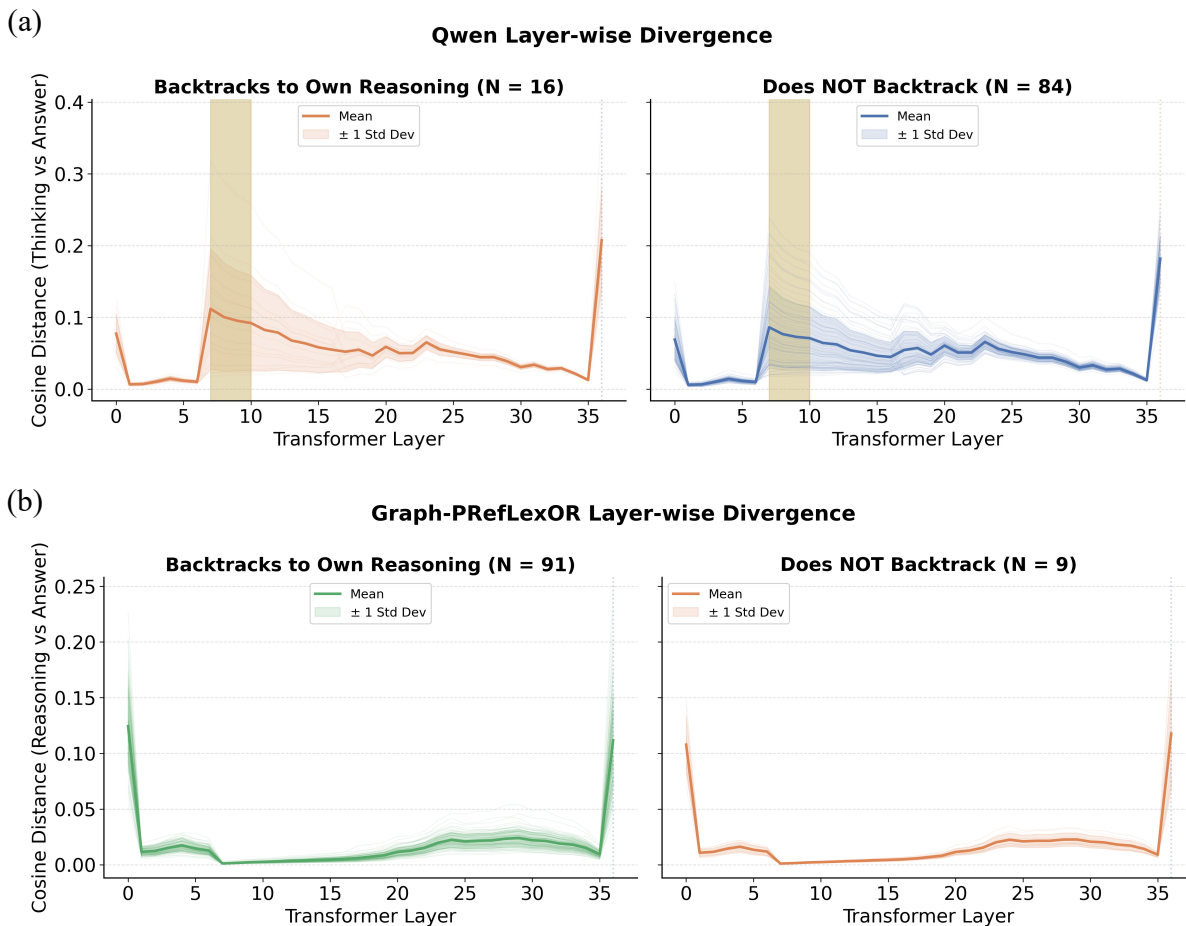


Figure 12: Backtracking-conditioned layer-wise hidden-state divergence for Qwen3-8B and Graph-PRefLexOR-8B. (a) Qwen3-8B divergence between thinking and final-answer, separated by whether the final answer backtracks to the model’s own thinking trace or to another source. Non-backtracking cases show larger divergence, particularly around layers 7-10 and at the final layer. (b) Graph-PRefLexOR-8B divergence between structured reasoning and final-answer states, separated by whether the final answer backtracks to its own reasoning stages or to another source. Graph-PRefLexOR maintains lower divergence across both groups, indicating stronger continuity between structured reasoning and final-answer generation. Shaded regions denote \pm one standard deviation across questions.

Qwen3-8B can produce final answers that are semantically close to Graph-PRefLexOR-8B outputs, but its internal representations often show a larger separation between visible thinking and final response generation. In contrast, Graph-PRefLexOR-8B maintains lower layer-wise divergence and stronger continuity between structured reasoning and final-answer states. This suggests that graph-native reasoning improves not only the interpretability of the generated trace, but also the internal stability of the reasoning-to-answer pathway.

2.7 Test-Time Graph Expansion

The preceding sections show that Graph-PRefLexOR produces structured reasoning traces, broader semantic exploration, and stronger reasoning–answer alignment in single-response settings. We next ask whether this graph-native reasoning format can also support iterative scientific ideation when additional test-time compute is available. To test this, we convert the reasoner into a self-expanding graph engine (Fig. 13). At each iteration, the model answers a question and emits a small ontological graph within its reasoning trace. This graph is merged into a growing memory graph, G_t , using embedding-based de-duplication, allowing accumulated graph structure rather than the context window alone to carry information across iterations. An expansion strategy then reads G_t and selects concepts or concept pairs to guide the next question. We compare four strategies: *frontier* expands low-degree leaves and the central high-betweenness hub; *novelty* targets the nodes farthest from the embedding centroid; *leap* forces a mechanism between the most dissimilar concepts and imports a principle from an unrelated field (\star); and *converse* uses a separate questioner model to introduce a new concept the graph does not yet contain. Iterating this loop tests whether additional inference-time computation expands, densifies, or recombines the model’s scientific idea space. Further detail about these strategies are given in the Methods section.

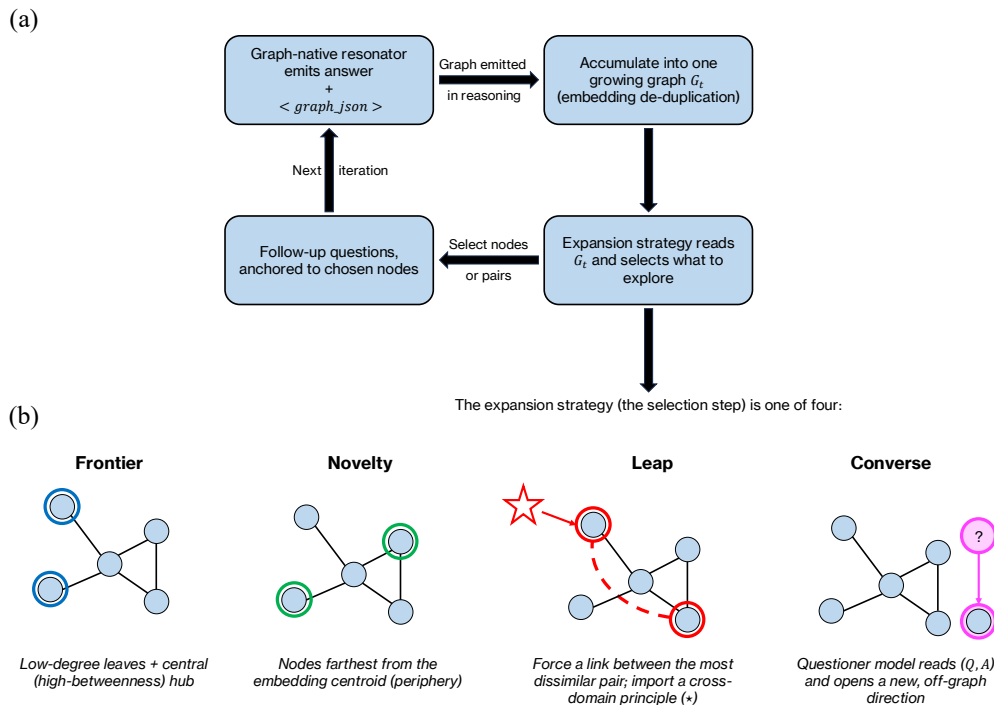


Figure 13: Graph-native ideation loop for test-time graph expansion. At each iteration, the reasoner answers a question, emits a small ontological graph, and merges it into a growing memory graph G_t using embedding-based de-duplication. An expansion strategy then selects concepts or concept pairs from G_t to generate the next question. The four strategies are *frontier*, which expands low-degree leaves and central hubs; *novelty*, which targets embedding-peripheral concepts; *leap*, which forces distant recombination and cross-domain import; and *converse*, which uses a questioner model to introduce off-graph directions.

Treating the model as a self-expanding ontological representation of thinking reveals that additional test-time compute produces recombinational rather than purely expansive growth (Fig. 14). Across all expansion strategies, the number of distinct concepts continues to increase, but both the explored embedding volume and the maximum distance from the seed plateau within a few hundred iterations. This indicates that the semantic territory associated with a fixed topic is bounded and rapidly covered. In contrast, the cumulative number of surprising recombinations, defined as atypically dissimilar concept pairs that are later bridged through a shared intermediate, continues to grow super-linearly through 2,000 iterations. Thus, additional compute does not primarily discover ever more distant regions of semantic space; instead, it densifies a bounded idea space by creating new bridges among already accessible concepts. The expansion strategies differ most clearly along this recombinational axis. The divergent *leap* policy converts compute into surprising recombinations most efficiently, whereas *converse* produces the largest number of concepts but the fewest bridges, indicating that concept generation and recombinational synthesis are distinct capabilities. The resulting idea space is organized around a small set of high-centrality hubs (Fig. 15a), while high-betweenness broker concepts, including cross-domain imports such as swarm intelligence and nanoscale assembly, mediate much of the bridging between sub-fields (Fig. 15b).

The dynamics of the *leap* run (Fig. 16) indicate a steady exploratory regime rather than a transition from exploration to consolidation. Although the rate of concept addition decreases as the topic saturates, the fraction of genuinely novel concepts remains close to one half throughout (Fig. 16a). At the same time, newly introduced concepts continue to appear at an approximately constant embedding distance from the seed (Fig. 16e), and the semantic span of recombination edges remains similarly stable (Fig. 16c). Thus, *leap* does not converge inward; instead, it persistently recombinates concepts across a bounded semantic radius.

The structural signature of this process is mesoscale densification. The number of communities increases from roughly six to seventy-five, while modularity declines steadily (Fig. 16b), indicating that sub-fields continue to proliferate even as they become increasingly interconnected. Growth is also cumulative rather than strictly sequential: several concepts introduced early remain dormant for more than 1,000 iterations before abruptly emerging as hubs (Fig. 16d). This shows that the system repeatedly revisits and amplifies earlier ideas. Together, these mechanisms explain the scaling behavior in Fig. 14: within a bounded semantic territory that is continuously and evenly recombinated, the number of bridged concept pairs, and hence the count of surprising recombinations, continues to increase without clear saturation.

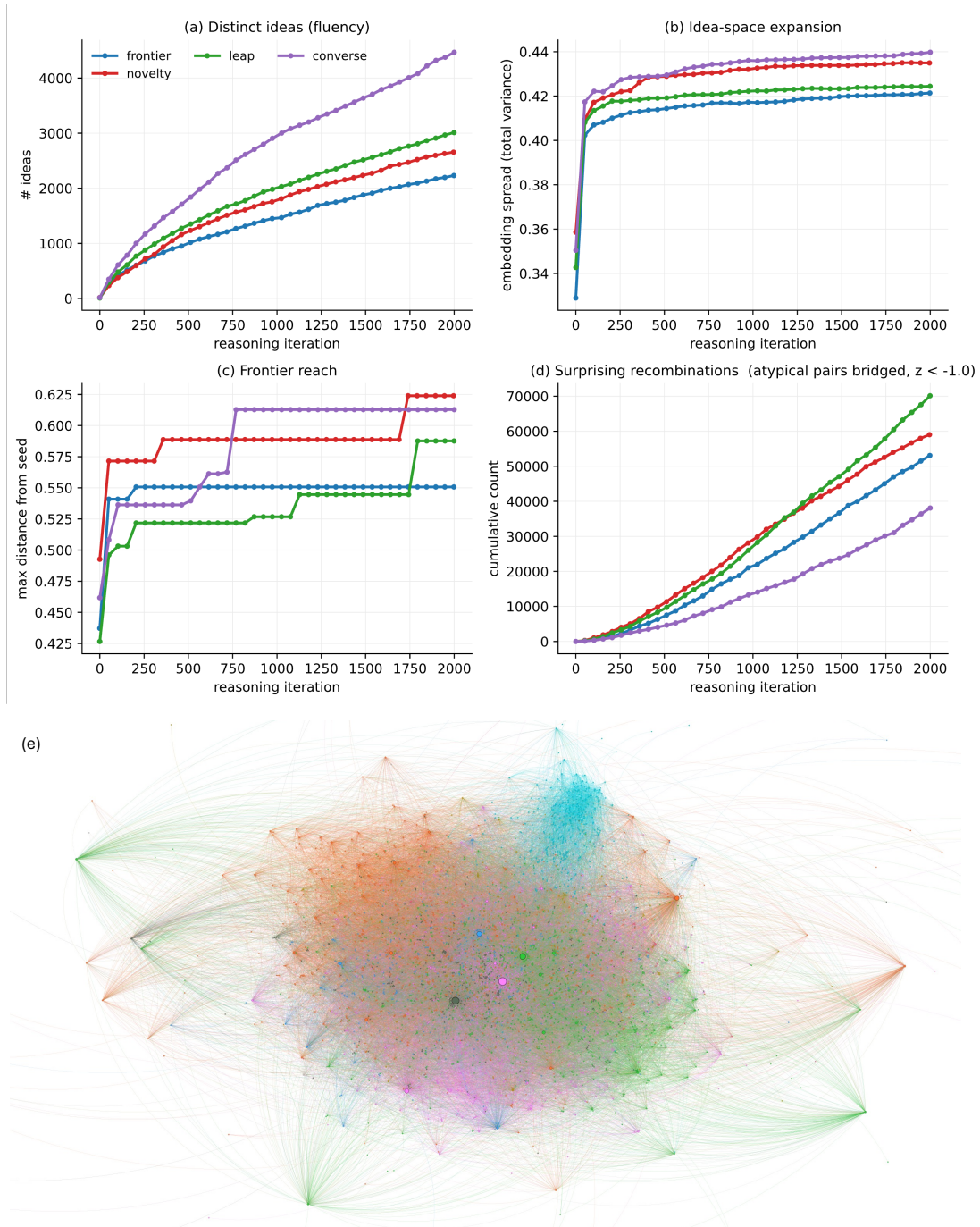


Figure 14: Test-time compute expands a bounded idea space through recombination. Four size-robust metrics are shown as a function of reasoning iteration up to 2,000 iterations for the four expansion strategies. The number of distinct concepts continues to increase (a), whereas the explored embedding volume (b) and maximum distance from the seed (c) saturate within a few hundred iterations, indicating that the semantic territory of a fixed topic is bounded and rapidly covered. In contrast, surprising recombinations (d), defined as atypically dissimilar concept pairs ($z < -1$ relative to the global pairwise-similarity null) later bridged through a shared intermediate concept, grow super-linearly and most rapidly under *leap*. Panel (e) shows the resulting *leap* graph after 3,700 iterations, with node color denoting modularity class and node size denoting degree (4,419 nodes and 37,064 edges). All reported z values denote standardized deviations from the corresponding randomized null distribution.

The connections formed by the model are statistically novel rather than artifacts of graph size (Fig. 17). To test this, we compare the accumulated graph against randomized null models that preserve its size and wiring. First, recurring relational motifs, defined as relation-typed two-step patterns, are strongly over-represented relative to a label-shuffled null model ($z \approx 100$ – 160). This indicates that the graph reuses structured relational templates rather than forming

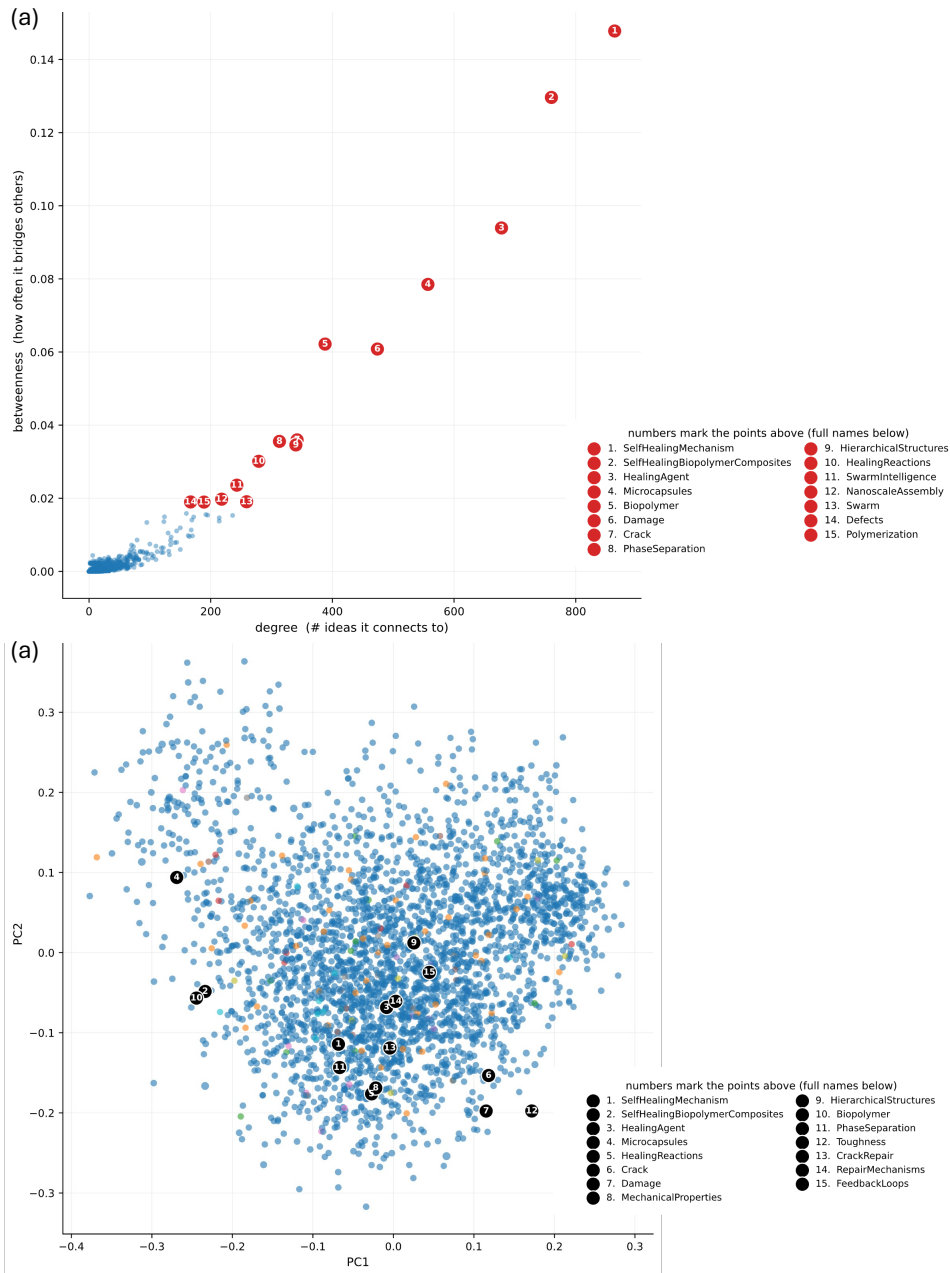


Figure 15: Semantic organization and broker concepts in the *leap* run. (a) Principal-component projection of concept embeddings, with colors indicating greedy-modularity communities and marker size indicating PageRank. The fifteen highest-PageRank concepts are numbered and listed below the map. (b) Broker concepts plotted by degree and betweenness. A small set of high-betweenness concepts mediates most cross-community bridging, including cross-domain imports such as swarm intelligence and nanoscale assembly.

arbitrary associations. The graph also exhibits substantially higher modularity than expected by chance ($Q = 0.29$, $z = +37.5$), while its edges preferentially connect semantically dissimilar concepts (edge heterophily $z = +18.2$). Thus, the accumulated structure is both organized and heterophilic, combining thematic modularity with cross-concept linkage.

The location of novelty is more specific. We quantify each concept pairing using combination typicality, which measures how unusual a pair of concepts is relative to the global concept-similarity distribution; more negative values indicate more atypical combinations. Direct edges are mildly homophilic and slightly more typical than random concept pairs (median $z \approx +0.4$), suggesting that explicit graph links often connect related ideas. In contrast, conceptual

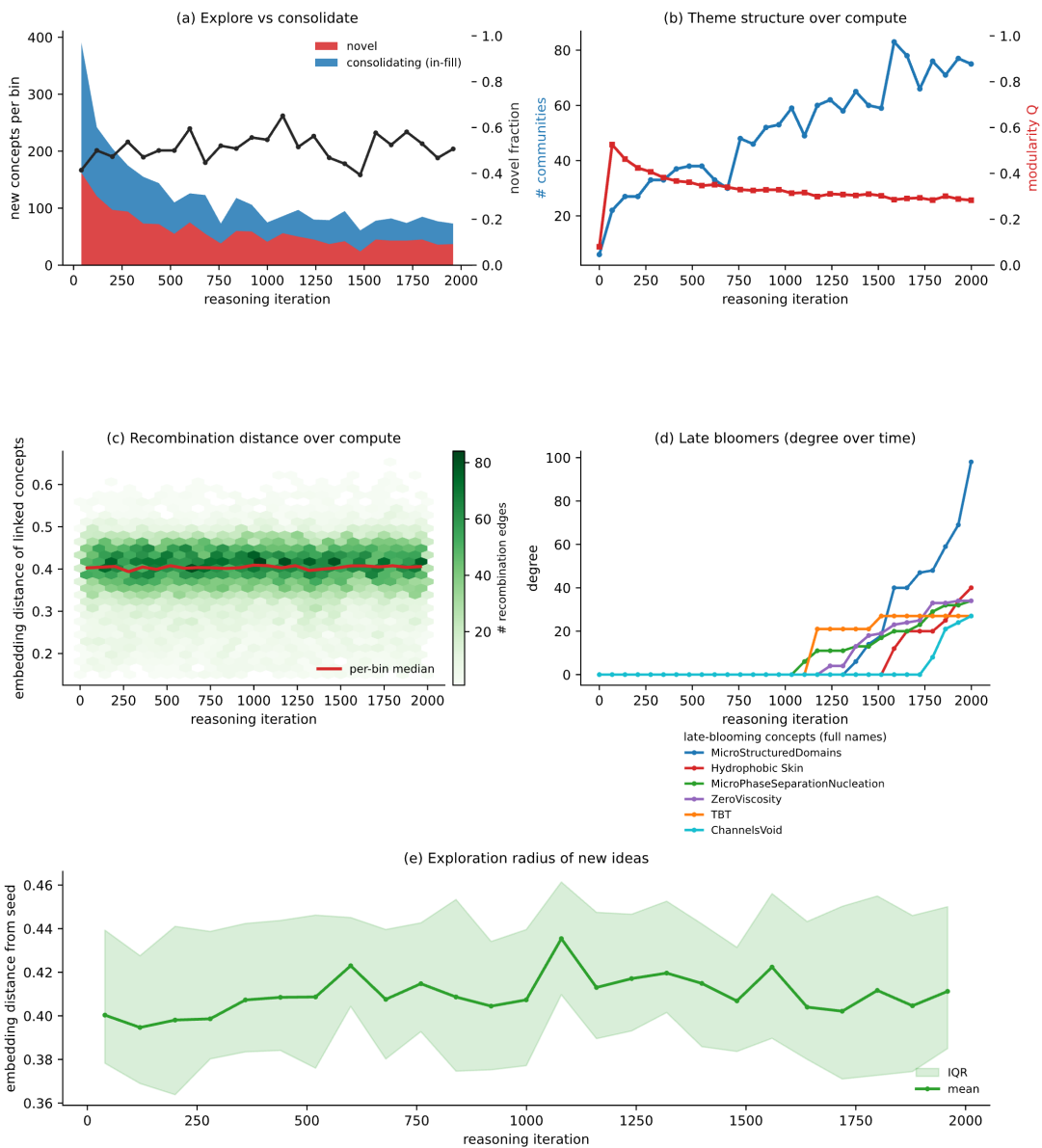


Figure 16: Growth dynamics of the *leap* run. The final graph is replayed in birth-iteration order, and each panel is evaluated using embedding geometry or mesoscale community structure rather than raw graph distance. (a) New concepts per iteration bin, separated into novel concepts and consolidating in-fill concepts; the black line shows the fraction of novel concepts. (b) Number of greedy-modularity communities and modularity Q . (c) Embedding distance between the endpoints of each recombination edge as a function of iteration. (d) Degree trajectories of six late-blooming concepts that acquire most of their final degree in the second half of the run. (e) Mean embedding distance of newly added concepts from the seed, with interquartile range. Together, the panels show sustained novelty, stable semantic reach, increasing cross-community interconnection, and delayed activation of high-impact concepts.

bridges across graph distance two lie deep in the atypical tail (median $z \approx -3.4$; Mann–Whitney $p = 1.1 \times 10^{-7}$), indicating that they connect concepts that are normally unrelated. Novelty therefore resides primarily in the long-range recombinations implied by the graph, rather than in the direct edges themselves. These long-range bridges are the same structures whose cumulative count continues to grow with additional test-time compute (Fig. 14).

3 Conclusion

In this work, we introduced a new generation of Graph-PrefLexOR models trained with Group Relative Policy Optimization (GRPO) for traceable scientific hypothesis generation in materials design. Building on earlier Graph-

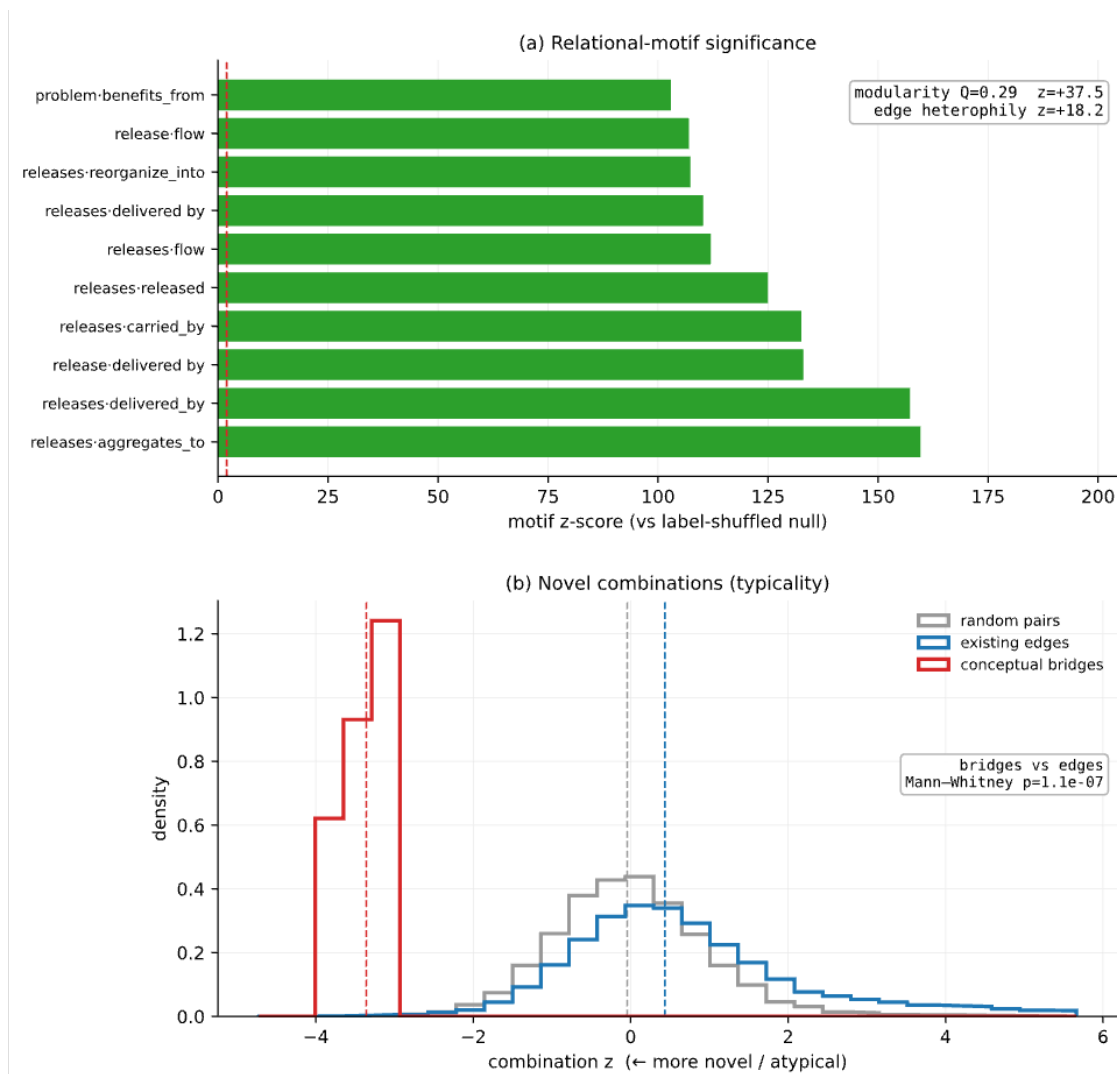


Figure 17: Statistical novelty of mined connections in the *leap* run. (a) Relational-motif significance relative to a label-shuffled null model. The ten most over-represented relation-typed two-step motifs reach $z \approx 100$ – 160 , far exceeding the ordinary significance threshold ($z = 1.96$), indicating that the graph follows consistent relational templates rather than random associations. The graph is also more modular than chance ($Q = 0.29$, $z = +37.5$) and preferentially connects semantically dissimilar concepts (edge heterophily $z = +18.2$). (b) Combination typicality analysis comparing random concept pairs, direct graph edges, and two-hop conceptual bridges. Direct edges are mildly typical (median $z \approx +0.4$), whereas two-hop bridges lie in the atypical tail (median $z \approx -3.4$; Mann-Whitney $p = 1.1 \times 10^{-7}$), showing that novelty is concentrated in long-range conceptual recombinations rather than direct links.

PRefLexOR formulations that mapped a task to a knowledge graph, abstract patterns, and a final answer, the present models use a more granular sentinel-based reasoning format consisting of `<brainstorm>`, `<graph>`, `<graph_json>`, `<patterns>`, and `<synthesis>` stages. This structure decomposes open-ended scientific reasoning into mechanism exploration, concept abstraction, machine-readable graph construction, pattern extraction, and final hypothesis synthesis. Unlike earlier ORPO/EXO-style preference-optimization approaches, the present models use GRPO to optimize structured reasoning behavior, providing an interpretable bridge between neural language generation and symbolic knowledge representation.

Across a manually curated benchmark of 100 open-ended questions derived from materials science and mechanics literature, Graph-PRefLexOR consistently outperformed its corresponding base models in reasoning quality, intellectual depth, and reasoning traceability. The strongest improvements were observed in traceability, indicating that graph-structured reasoning primarily enhances the organization and causal transparency of intermediate reasoning. Comparisons with no-thinking baselines further show that these gains arise from explicit intermediate reasoning rather than model scale or architecture alone. Embedding-based analyses support this conclusion: Graph-PRefLexOR reasoning traces occupy broader and more structured semantic regions, follow more coherent stage-wise trajectories, and exhibit approximately $2 \times$ – $3 \times$ greater semantic diversity than base-model traces.

The reasoning–answer alignment analyses show that these structured traces are not merely decorative. Semantic backtracking indicates that Qwen3-8B final answers align with their own visible thinking traces in only a minority of cases and more often occupy semantic regions closer to Graph-PRefLexOR-derived outputs. In contrast, Graph-PRefLexOR final answers remain strongly anchored to their own structured reasoning pathway, most frequently backtracking to the <synthesis> stage. Layer-wise hidden-state analyses reinforce this pattern: Qwen3-8B exhibits larger reasoning–answer divergence, particularly around layers 7–10 and at the final layer, whereas Graph-PRefLexOR maintains lower divergence across layers and backtracking groups. Together, these results suggest that graph-native reasoning improves not only the visible organization of reasoning traces, but also the semantic and representational continuity between reasoning and final answer generation.

We further showed that graph-native reasoning can be extended from single-response hypothesis generation to iterative test-time ideation. By accumulating the model’s emitted <graph_json> outputs into a growing graph memory and using expansion policies to select follow-up questions, Graph-PRefLexOR becomes a self-expanding graph engine. This analysis reveals that additional test-time compute does not simply expand semantic territory indefinitely. Instead, the explored embedding volume and maximum distance from the seed saturate rapidly, while the number of surprising recombinations continues to grow super-linearly. The most divergent expansion strategy, *leap*, is especially effective at converting compute into long-range conceptual bridges, showing that concept generation and recombinational synthesis are distinct capacities.

Statistical null-model analyses confirm that the resulting idea graphs are not artifacts of graph size. Relation-typed motifs are strongly enriched relative to randomized label assignments, the graph exhibits modular organization beyond chance, and edge heterophily indicates systematic linking of semantically dissimilar concepts. Most importantly, novelty is concentrated not in direct graph edges, which remain mildly homophilic, but in two-hop conceptual bridges that connect normally unrelated concepts. This finding reframes scientific ideation as the progressive densification of a bounded semantic space: test-time compute increases the number of meaningful bridges among accessible concepts, rather than merely pushing the model toward ever more distant regions.

Taken together, these results establish Graph-PRefLexOR as a compact and interpretable framework for scientific hypothesis generation. The model improves final reasoning performance by restructuring the intermediate pathway through which answers are produced, grounding final responses in graph-native reasoning stages, and enabling iterative recombination through accumulated graph memory. More broadly, this work emphasizes that scientific language models should be evaluated not only by final-answer quality, but also by reasoning traceability, semantic diversity, reasoning–answer alignment, representation dynamics, and test-time recombinational growth. Future work will extend Graph-PRefLexOR toward autonomous discovery workflows that integrate literature reasoning, simulation, experimental feedback, and closed-loop design of polymers, composites, and multifunctional materials across scales.

4 Materials and Methods

4.1 Training Strategy

We train a family of graph-native reasoning models, Graph-PRefLexOR at 1.7B, 3B, and 8B parameters (Table 3), with a single two-stage recipe applied to three instruction-tuned backbones that differ in an important respect. The Qwen3-1.7B and Qwen3-8B backbones are already reasoning models with a native thinking mode, so the recipe adapts an existing reasoning ability to our specific graph-native format. The Llama-3.2-3B-Instruct backbone, by contrast, is a standard instruction-tuned model with no built-in reasoning behavior; the 3B therefore represents the harder case of converting an ordinary chat model into a graph-native reasoner essentially from scratch. As we show in §4.1.4, this distinction (not merely parameter count) is what most strongly governs each model’s training dynamics. We first describe how the training corpus is constructed (§4.1.1), then the two-stage training approach (§4.1.2), the composite reward (§4.1.3), and the resulting training dynamics (§4.1.4).

4.1.1 Dataset Construction

The training corpus is built by a teacher-distillation pipeline that converts raw text into structured, graph-native preference pairs. Source passages are sampled from a mixture of streaming corpora (general educational web text (*fineweb-edu*) and a domain-specific biological and mechanical-materials mixture (*bio-silk-mech-mix-80K*)) giving both breadth and a materials-science focus. For each passage, a strong teacher model (GPT-5.1) first writes a single challenging, self-contained, expert-level question grounded in the passage, and then produces the preferred (chosen) response: a complete graph-native reasoning trace that fills the exact sentinel template of §4.1.2: <brainstorm> → <graph> → <graph_json> (a strict nodes/edges JSON) → <patterns> → <synthesis>, closed by </think> and followed by a thorough final answer, under a system prompt instructing it to “reason using a graph-based latent structure.” A

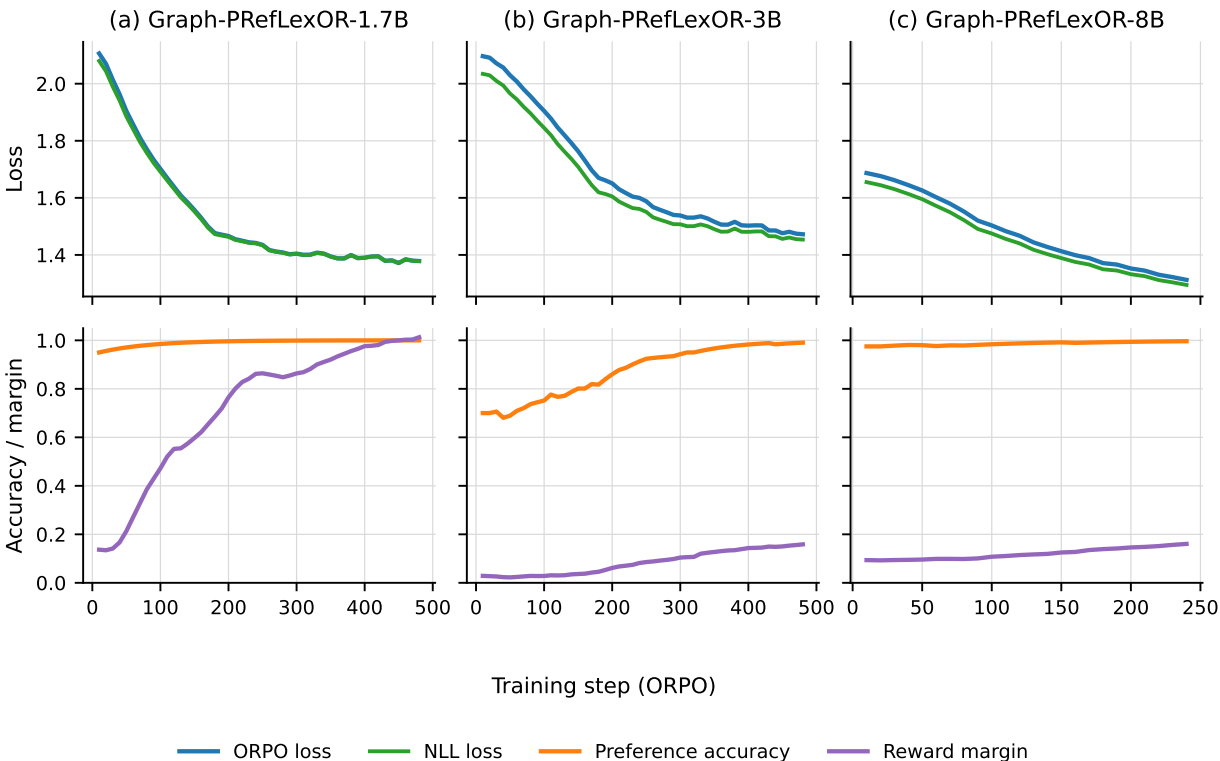


Figure 18: ORPO cold start for all three models. Top row: total ORPO loss and its NLL component; bottom row: preference accuracy and reward margin (both in $[0, 1]$); columns are (a) 1.7B, (b) 3B, (c) 8B, with y shared per row (note the differing ORPO durations, $\sim 480/480/240$ steps). Loss falls and preference accuracy saturates near 1.0 for all backbones; the 1.7B’s much larger reward margin reflects its higher learning rate rather than scale.

separate, deliberately weak model (GPT-5-nano), prompted to give a hurried 1–3 sentence answer with no reasoning and no special tokens, produces the dispreferred (rejected) response. The gold answer is the text following `</think>`, and any example whose `graph_json` fails to parse or whose answer is empty is discarded, so every retained record carries a valid graph and a complete answer. Each record stores the question (prompt), the gold answer, the chosen and rejected completions, and the extracted teacher graph.

This construction directly shapes both training stages. ORPO (§4.1.2) consumes the chosen/rejected pairs, so by design it contrasts a rich, structured graph-reasoning trace against a shallow direct answer (a deliberately large behavioral gap that, as we show in §4.1.4, makes the preference ordering easy to learn and the cold-start preference accuracy saturate quickly). Graph-GRPO consumes only the prompt and gold answer (the chosen/rejected fields are unused), letting the policy explore its own traces while the composite reward) built on the same `graph_json` object that the teacher template seeds grades correctness and graph quality. Distilling from a strong teacher means the target behavior is demonstrated rather than discovered, which is what lets even the small backbones acquire the structured format in a single epoch.

4.1.2 Training Approach

All sizes are trained with the same two-stage recipe (Table 2). The model emits a structured, graph-native reasoning trace rather than free-form chain-of-thought, following the recursive, reflective paradigm of PRefLexOR [29]: all deliberation is enclosed in a `<think>...</think>` block containing, in order, a `<brainstorm>` (open exploration), a `<graph>` (a natural-language sketch of entities and relations), a `<graph_json>` (a machine-readable knowledge graph with typed nodes and `source/relation/target` edges), a `<patterns>` section (motifs read off the graph), and a `<synthesis>`; the final answer is emitted after the closing `</think>` tag. All runs used a single GPU; the 1.7B and 8B (Qwen3) models were trained on an NVIDIA A100-SXM4-80GB, while the 3B (Llama-3.2-3B-Instruct) model was trained on an NVIDIA GH200 480GB.

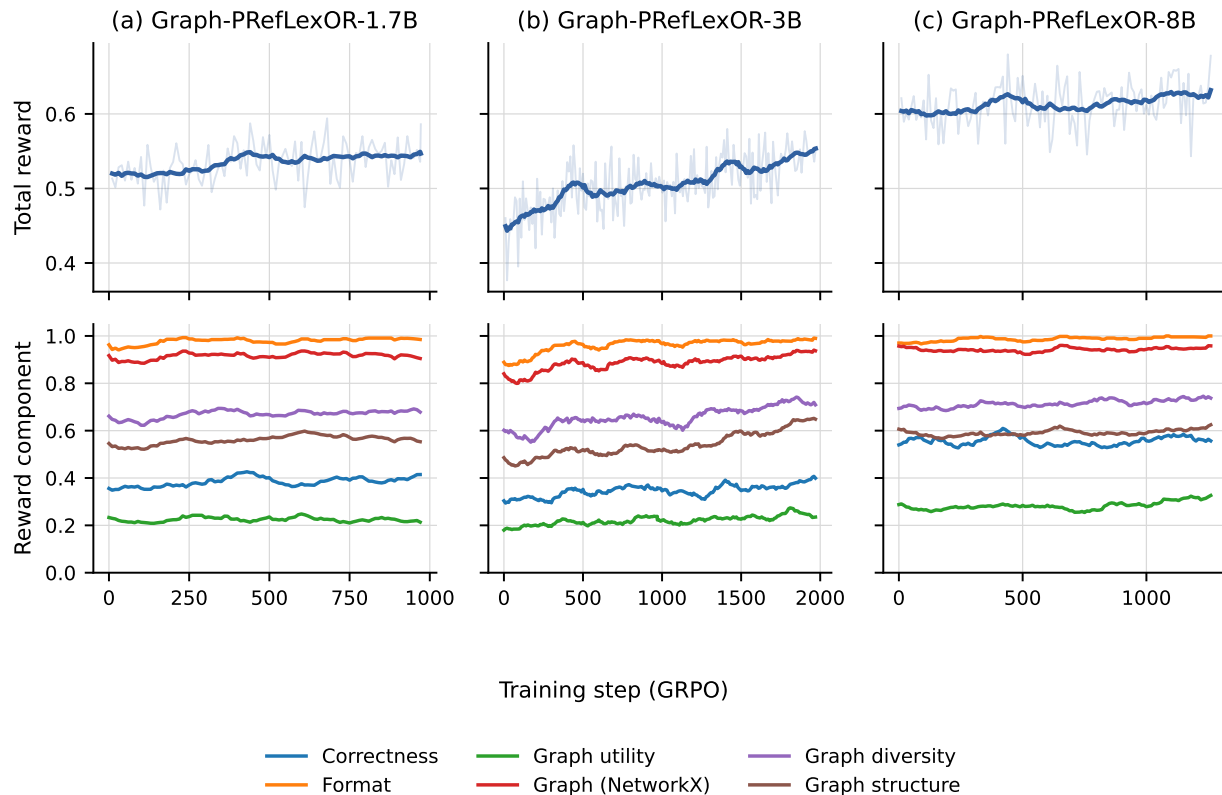


Figure 19: Graph-GRPO reward for all three models. Top row: total composite reward; bottom row: the six reward components; columns are (a) 1.7B, (b) 3B, (c) 8B (differing GRPO durations, $\sim 970/1970/1260$ steps), with y shared per row for direct comparison. The 8B starts highest and the 3B climbs most, while graph utility (green) is the lowest component at every scale.

Rationale for This Reasoning Structure

Each of the sentinel blocks plays a distinct epistemic role, and their order encodes a deliberate divergent-to-convergent scientific reasoning pipeline: *exploration* \rightarrow *formalization* \rightarrow *abstraction* \rightarrow *explanation*. `<brainstorm>` performs a wide search over hypotheses and candidate mechanisms; `<graph>` draws a scientifically grounded conceptual blueprint; `<graph_json>` commits it to a canonical, machine-readable knowledge graph; `<patterns>` compresses the graph into reusable motifs (causal loops, modular micro \rightarrow meso \rightarrow macro hierarchies, bottlenecks, invariants); and `<synthesis>` reads an ordered explanation back off the graph. We adopt this representation for three reasons. First, *relational faithfulness*: scientific reasoning is intrinsically about entities and their causal and structural relationships, so an explicit graph is a more natural, inspectable substrate than linear text, and the multi-scale abstractions targeted by `<patterns>` are precisely graph-theoretic. Second, *verifiability and reward access*: by emitting a canonical `<graph_json>`, the reasoning becomes a parseable object that the reward function can interrogate directly—the validity, structure, diversity, and graph-utility terms of §4.1.3 all operate on this object. Third, *answer faithfulness*: the final answer is produced after `</think>` and must remain consistent with, and derivable from, the graph; coupled with the graph-utility reward, this makes the graph a load-bearing intermediate representation rather than decorative scratch work.

Stage 1: ORPO Cold Start

We first align each backbone with Odds-Ratio Preference Optimization (ORPO) [46], a reference-free objective that combines a supervised negative-log-likelihood (NLL) term on the preferred response y_w with an odds-ratio penalty that suppresses the dispreferred response y_l :

$$\mathcal{L}_{\text{ORPO}} = \underbrace{-\log P_{\theta}(y_w | x)}_{\mathcal{L}_{\text{SFT}}} + \lambda \underbrace{-\log \sigma \left(\log \frac{\text{odds}_{\theta}(y_w | x)}{\text{odds}_{\theta}(y_l | x)} \right)}_{\mathcal{L}_{\text{OR}}}, \quad \text{odds}_{\theta}(y | x) = \frac{P_{\theta}(y | x)}{1 - P_{\theta}(y | x)}, \quad (1)$$



Figure 20: Graph-GRPO dynamics across the three models, with each run rescaled to $[0, 1]$ training progress so the durations align. **(a)** Reasoning-trace length: mean terminated completion length (top) and fraction of completions truncated at the token budget (bottom). **(b)** Optimization diagnostics: within-group reward standard deviation, i.e. the scale of the group-normalized advantage of Eq. (2) (top), and policy entropy (bottom). We find that the 3B alone learns to fit its budget (truncation $20\% \rightarrow 1\%$) and carries the largest reward dispersion.

where σ is the logistic function and λ weights the preference term. ORPO collapses the usual supervised-fine-tuning-then-preference-optimization pipeline into a single monolithic stage and requires no frozen reference model. We train for one epoch with a 5% held-out split (seed 42) for evaluation. This stage is the cold start in the sense of launching RL directly from a base model that almost never produces a well-formed trace yields a sparse, high-variance reward; establishing reliable format adherence and a basic preference for good graph reasoning first makes the subsequent RL signal dense and learnable [29]. The cold start thus plays a different role for each backbone: for the already-reasoning Qwen3 models it primarily adapts existing reasoning ability to the graph-native format, whereas for the standard Llama-3.2-3B-Instruct base it must induce the reasoning behavior itself.

Stage 2: Graph-GRPO

Starting from the cold-start checkpoint, we apply Group Relative Policy Optimization (GRPO) [31]. For each prompt q we sample a group of G completions $\{o_1, \dots, o_G\}$, score each with the composite reward $R(\cdot)$ of §4.1.3, and form a critic-free, group-normalized advantage

$$\hat{A}_i = \frac{R(o_i) - \text{mean}(\{R(o_j)\}_{j=1}^G)}{\text{std}(\{R(o_j)\}_{j=1}^G) + \epsilon}, \quad (2)$$

used in the clipped policy-gradient objective with a KL penalty to a reference policy,

$$\mathcal{J}(\theta) = \mathbb{E}_{q, \{o_i\}} \left[\frac{1}{G} \sum_{i=1}^G \min(\rho_i \hat{A}_i, \text{clip}(\rho_i, 1 - \epsilon, 1 + \epsilon) \hat{A}_i) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right], \quad \rho_i = \frac{\pi_\theta(o_i | q)}{\pi_{\theta_{\text{old}}}(o_i | q)}. \quad (3)$$

Dispensing with a learned value network makes GRPO memory-efficient and well matched to our setting, where the reward is a black-box composite of judge calls and graph analytics rather than a differentiable signal. We sample $G = 8$ completions per prompt, generate with a vLLM backend [47] for throughput, and adapt only LoRA [48] parameters to keep the update lightweight and to mitigate catastrophic forgetting. Per-size budgets are in Table 2.

4.1.3 Reward Definition

Each completion o receives a scalar reward that is a fixed convex combination of six components, each normalized to $[0, 1]$:

$$R(o) = \sum_k w_k r_k(o), \quad \mathbf{w} = (\underbrace{0.30}_{\text{corr}}, \underbrace{0.15}_{\text{fmt}}, \underbrace{0.25}_{\text{util}}, \underbrace{0.10}_{\text{nx}}, \underbrace{0.10}_{\text{div}}, \underbrace{0.10}_{\text{struct}}), \quad \sum_k w_k = 1. \quad (4)$$

Two components are model-graded by an external LLM judge (`grok-4-1-fast-non-reasoning`) and four are computed programmatically from the parsed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. In all reward figures the headline “total reward” is the trainer’s per-step mean $\frac{1}{G} \sum_i R(o_i)$, equivalent to Eq. (4).

Semantic rewards (judge-graded)

Let $J(a, a^*) \in [0, 1]$ denote the judge’s continuous grade of a candidate answer a against the gold answer a^* . *Correctness* grades the post-`</think>` answer a_o directly, $r_{\text{corr}} = J(a_o, a^*)$. *Graph utility* (the central signal) first asks the judge to reconstruct an answer $\hat{a}_o = \text{RECONSTRUCT}(G_o)$ using only the emitted `graph_json` and no outside knowledge, then grades it, $r_{\text{util}} = J(\hat{a}_o, a^*)$. This is an information-bottleneck test: the model is rewarded only when it has offloaded genuinely sufficient information into an explicit, standalone graph.

Format reward

A graded check that the structured trace is present and the graph is parseable, $r_{\text{fmt}} = \text{clip}_{[0,1]}(\sum_{s \in S} c_s \mathbb{K}[s \text{ present}])$, with section credits for `think` (0.15), `brainstorm` (0.10), `graph` (0.15), a JSON-parseable `graph_json` (0.20), `patterns` (0.15), `synthesis` (0.15), and a non-empty node set (0.10). Credit for `graph_json` and everything after it is gated on the JSON parsing successfully, so a malformed graph caps the score.

NetworkX-validity reward

With E_{inv} the edges referencing nonexistent nodes and ℓ the number of self-loops,

$$r_{\text{nx}} = \text{clip}_{[0,1]} \left(0.3 \mathbb{K}[n > 0] + 0.3 \mathbb{K}[E_{\text{val}} > 0] + 0.2 \left(1 - \frac{|E_{\text{inv}}|}{m} \right) + 0.1 \mathbb{K}[\ell = 0] + 0.1 \mathbb{K}[\text{weakly connected}] \right), \quad (5)$$

$E_{\text{val}} = m - |E_{\text{inv}}|$, rewarding internally consistent, connected graphs.

Diversity reward

We embed the m' textual graph elements (node ids and `source relation target`) with a Sentence-BERT model ϕ (`all-MiniLM-L6-v2`) [49] and measure the mean off-diagonal cosine similarity

$$\bar{s} = \frac{1}{m'(m' - 1)} \sum_{k \neq l} \frac{\phi(t_k)^\top \phi(t_l)}{\|\phi(t_k)\| \|\phi(t_l)\|}, \quad r_{\text{div}} = \text{clip}_{[0,1]}(0.9d + b), \quad d = \text{clip}_{[0,1]} \left(1 - \frac{\bar{s} - 0.15}{0.35} \right), \quad (6)$$

with a richness bonus $b = \min(0.1, m'/100)$. This penalizes degenerate, collapsed graphs of near-duplicate nodes that would otherwise reward-hack the validity and structure terms.

Structure reward

A topology score $r_{\text{struct}} = \text{clip}_{[0,1]}(s_{\text{size}} + s_{\text{dens}} + s_{\text{int}} + s_{\text{depth}} + s_{\text{conn}})$, with a size term peaking for 5–20 nodes; $s_{\text{dens}} = \min(0.2, 2\rho)$ for directed density $\rho = \frac{m}{n(n-1)}$; an internal-node term $s_{\text{int}} = 0.3 \frac{n_{\text{int}}}{n}$ over nodes with both in- and out-edges; a depth term $s_{\text{depth}} = \min(0.2, 0.2 \frac{\min(L, 6)}{6})$ for longest DAG path L ; and a weak-connectivity term (0.1). Internal nodes correspond to intermediate inferences and L to reasoning-chain length, shaping the graph into a connected, hierarchical scaffold.

Design rationale

The bulk of the weight (0.55) is placed on the two semantic objectives (correctness and graph utility) that capture the true task, while the four cheap programmatic terms (0.45) act as dense shaping and anti-hacking rewards that keep every sampled trace valid, diverse, and well-structured. Importantly, the components have heterogeneous effective maxima: format and NetworkX-validity can reach 1.0, whereas the diversity and structure terms are soft-capped by design, a

topically coherent graph cannot attain mean pairwise dissimilarity ($d < 1$), and the internal-node and DAG-depth terms in r_{struct} are mutually competing. Their absolute levels are therefore not directly comparable, and the total reward saturates below 1.0 by construction.

4.1.4 Training Dynamics and Results

Across all three sizes the cold start instills the reasoning format and a clear preference ordering, and Graph-GRPO then improves the composite reward; the backbone type and size govern where each model starts and how much headroom remains.

Graph-PRefLexOR-1.7B

On a Qwen3-1.7B backbone (already a reasoning model with a native thinking mode) the ORPO loss falls from ~ 2.11 to ~ 1.38 , with the NLL term essentially coincident with the total (the odds-ratio penalty is negligible under the larger 5×10^{-5} learning rate), while the preference accuracy rises from 0.95 to 1.0 and the implicit reward margin grows steeply from 0.14 to ~ 1.0 (Fig. 18(a)), a far larger margin than the other sizes, reflecting the higher learning rate rather than model scale. Under Graph-GRPO the composite reward improves modestly from ~ 0.52 to ~ 0.55 over the ~ 970 -step run (Fig. 19(a)), with correctness rising $0.36 \rightarrow 0.42$, format and NetworkX-validity near their ceilings, and graph utility the lowest component (~ 0.22).

Graph-PRefLexOR-3B

On the Llama-3.2-3B-Instruct backbone (a standard instruction model with no native reasoning, converted into a graph-native reasoner from scratch) the cold start converges cleanly within one epoch: the total ORPO loss and its NLL component fall from ~ 2.1 to ~ 1.47 and plateau after ~ 300 steps, separated by only ~ 0.02 – 0.03 (a light odds-ratio regularizer), while the preference accuracy reaches ≈ 1.0 and the reward margin grows from 0.03 to 0.16, with the same accuracy on the held-out split (Fig. 18(b)). The modest absolute margin reflects the deliberately large chosen/rejected gap built into the data (§4.1.1): the pairs are easy to separate, so this stage installs format and preference adherence rather than final reasoning quality. Graph-GRPO then improves the composite reward from ~ 0.46 to ~ 0.55 over the ~ 1970 -step (two-segment) schedule (Fig. 19(b)), representing the largest relative RL gain of the three, consistent with the weakest cold start. The programmatic format and NetworkX-validity terms saturate early; structure climbs $0.53 \rightarrow 0.64$ and diversity $0.67 \rightarrow 0.71$; correctness improves only gradually ($0.34 \rightarrow 0.38$) and graph utility is the lowest throughout ($0.17 \rightarrow 0.25$).

Graph-PRefLexOR-8B

On a Qwen3-8B backbone the ORPO loss falls from ~ 1.69 to ~ 1.31 in only ~ 240 steps, and the preference accuracy is ≈ 1.0 from the outset (Fig. 18(c)), the strong base separates preferences almost immediately, so the cold start mainly instills the output format. Under Graph-GRPO the composite reward is high and rises only modestly from ~ 0.60 to ~ 0.63 , with a mid-run dip to ~ 0.60 (Fig. 19(c)). The component view explains why: the 8B leaves the cold start already near the 3B’s converged operating point; correctness sits at ~ 0.57 and the format, validity, diversity, and structure terms are high and stable, so RL consolidates and lightly polishes rather than driving large gains; only graph utility drifts upward ($0.29 \rightarrow 0.32$) and stays the lowest component.

4.1.5 Generation-length dynamics

Beyond reward, Graph-GRPO reshapes the models’ generation behavior (Fig. 20(a)). The 1.7B and 8B models, which rarely exhaust their token budget ($\leq 3\%$ truncated), lengthen their completed reasoning traces over training (to $\sim 2.6\text{k}$ and $\sim 2.4\text{k}$ tokens respectively). The 3B exhibits the opposite and more pronounced dynamic: it initially truncates up to $\sim 20\%$ of completions against the 8000-token budget, and reinforcement learning drives this rate down to $\sim 1\%$ while simultaneously shortening the mean terminated length ($\sim 2.7\text{k} \rightarrow 2.3\text{k}$ tokens). Because a truncated trace cannot emit a well-formed closing answer, it is penalized by the format and correctness terms and thus selected against; GRPO therefore teaches the policy to complete its graph-native reasoning within budget, a regularization of length and termination that the reward curve alone does not reveal.

Optimization Diagnostics

The within-group reward standard deviation (Fig. 20(b), top) sets the scale of the GRPO advantage (Eq. 2): the 3B begins with the largest dispersion (~ 0.11) and the 1.7B and 8B with roughly half that (~ 0.05 – 0.06), so the 3B receives

Table 2: Training configuration for the Graph-PRefLexOR family. All sizes share the reasoning-trace format and the six-component reward of §4.1.3.

Setting	1.7B	3B	8B
Backbone	Qwen3-1.7B	Llama-3.2-3B-Instruct	Qwen3-8B
<i>Stage 1: ORPO cold start</i>			
epochs / max length	1 / 2048	1 / 2048	1 / 2048
learning rate	5×10^{-5}	1×10^{-5}	1×10^{-5}
batch size	1	1	2
<i>Stage 2: Graph-GRPO</i>			
learning rate	5×10^{-6}	5×10^{-6}	5×10^{-6}
group size G	8	8	8
per-device batch / grad. accum.	2 / 4	2 / 4	1 / 8
epochs	1	1	3
max completion length	8000	8000	3500
LoRA (r, α)	(32, 64)	(32, 64)	(16, 32)

Table 3: Models used in this study and corresponding Hugging Face identifiers.

Model	Scale	Base Model	Base type	Hugging Face ID
Graph-PRefLexOR	8B	Qwen3-8B	Reasoning	lamm-mit/Graph-Preflexor-8b_12292025
Graph-PRefLexOR	3B	Llama-3.2-3B-Instruct	Standard	lamm-mit/Graph-Preflexor-3b_08012026
Graph-PRefLexOR	1.7B	Qwen3-1.7B	Reasoning	lamm-mit/Graph-Preflexor-1.7b_08012026

the strongest per-group learning signal (consistent with its larger reward gains) and as its dispersion decays toward ~ 0.07 the signal, and the gains, taper. Policy entropy (bottom) decreases modestly for all models, indicating gradual sharpening; the 1.7B remains the most exploratory and the 3B sharpens most. No run shows vanishing advantages (the fraction of zero-variance groups stays at 0), confirming the reward provides a usable gradient throughout.

4.1.6 Cross-model Summary and Analysis

Three trends are size-invariant. First, the cold start always achieves near-perfect preference accuracy, confirming that ORPO reliably installs the structured trace before RL, aided by the large chosen/rejected gap built into the data. Second, the headroom that Graph-GRPO exploits is governed less by parameter count than by whether the backbone was already a reasoning model: the Qwen3-1.7B and Qwen3-8B models, which begin with native reasoning ability, leave the cold start close to saturation and are mostly consolidated by RL, whereas the Llama-3.2-3B-Instruct model (a standard chat model converted into a reasoner from scratch) begins lowest, carries the largest within-group reward dispersion (Fig. 20(b)), and shows the largest RL gains. This is consistent with the 8B’s high starting correctness and the 3B being the most “moved” by training despite its intermediate size. Third, and most importantly, graph utility is the binding constraint at every scale (0.21–0.32): producing knowledge graphs that are semantically sufficient to reconstruct the answer (not merely valid and well-formed, which all models achieve) is the central open challenge of graph-native reasoning, and the principal target for future scaling.

4.2 Benchmark Question Generation

The open-ended benchmark is constructed using a multi-stage pipeline for paper ingestion, section extraction, question generation, and question refinement (see Figure 21). We collect research papers spanning several domains relevant to scientific reasoning and materials design, including large language models, spider silk, polymer nanocomposites, epoxy networks, and collagen-based protein materials. Each PDF is converted to Markdown using the Marker library, which performs layout-aware text extraction without OCR or LLM-assisted parsing [50].

Each Markdown file is then processed using OpenAI *gpt-4o-mini* with a structured output schema to extract key paper-level fields, including the title, DOI, abstract, results, discussion, and conclusion [51]. The Introduction, methods, and references sections are excluded because they typically contain background, procedural details, or citation metadata rather than the high-level mechanistic findings targeted by this benchmark. The extraction prompt includes robustness rules to account for variations in section naming, including *Results and Discussion*, *Concluding Remarks*, *Summary*, and *Discussion and Outlook*. The abstract, results, discussion, and conclusion sections are subsequently merged into a consolidated text block with explicit section headers. These records are compiled into a JSONL dataset containing the paper title, DOI, source text, and associated metadata. This consolidated representation serves as the input for benchmark question generation.



Figure 21: Workflow for constructing the open-ended scientific reasoning benchmark from research papers.

For each paper, OpenAI *gpt-5.4* with high reasoning effort generates one self-contained, research-level evaluation question. The resulting benchmark contains 100 open-ended questions. Each question is assigned to one of five predefined reasoning categories: *causal_multiscale_reasoning*, *tradeoff_and_non_monotonicity*, *hidden_variable_identification*, *model_abstraction_and_breakdown*, or *cross_domain_mapping*. The generation prompt requires each question to define a clear scientific system, include relevant variables and mechanisms, and pose a non-trivial reasoning challenge involving a tradeoff, hidden variable, failure mode, or mechanistic breakdown. Additional constraints enforce a single-paragraph format, self-contained framing, a target length of 150–200 words, and a mechanistic final task beginning with either *Explain why* or *Then identify*.

Finally, each generated question is passed through a second *gpt-5.4* refinement step to improve readability, grammar, precision, and benchmark suitability. This editing pass preserves the original system, variables, causal structure, question type, and intended reasoning challenge, while avoiding the introduction of new scientific claims or simplification of the task [52]. The final output is a polished JSONL benchmark of 100 self-contained, open-ended scientific reasoning questions designed to evaluate mechanistic reasoning, causal inference, and hypothesis generation. Paper-level metadata, including DOI, title, and topic tags, are available at <https://huggingface.co/datasets/lamm-mit/graph-preflexor-grpo-benchmark>.

4.3 Answer Backtracking and Hidden-State Analysis

We use semantic backtracking to measure which intermediate text is closest to a model’s final answer. For each question i , let a_i denote the final answer and let $\mathcal{C}_i = \{c_{i1}, c_{i2}, \dots, c_{iK}\}$ denote the set of candidate reference texts. We embed the final answer and all candidate references using BAAI/bge-base-en-v1.5 [53]. The BGE family provides general-purpose text embeddings for semantic retrieval and comparison [54], and sentence embeddings are commonly evaluated using cosine-based semantic similarity tasks [55]. All embeddings are ℓ_2 -normalized before comparison.

For each candidate reference c_{ij} , we compute cosine similarity with the final answer:

$$s_{ij} = \cos(e(a_i), e(c_{ij})) = \frac{e(a_i)^\top e(c_{ij})}{\|e(a_i)\|_2 \|e(c_{ij})\|_2}.$$

Because the embeddings are normalized, this is equivalent to the dot product between the two normalized vectors. The backtracking source is then assigned by top-1 nearest-reference selection:

$$j_i^* = \arg \max_j s_{ij}.$$

The source $c_{ij_i^*}$ is the candidate to which the final answer backtracks. We do not use a fixed absolute cosine-similarity threshold. Instead, each answer is assigned to the candidate source with the highest cosine similarity. This avoids choosing an arbitrary cutoff and makes the analysis fully reproducible from the embeddings and candidate list. If two candidates have exactly the same similarity up to numerical precision, we use a fixed deterministic ordering of candidates to break the tie.

For Qwen3-8B, we run the model with thinking enabled and split each output into a visible thinking trace and a final answer. We compare Qwen’s final answer with six candidate references: Qwen’s own thinking trace, Graph-PRefLexOR 8B’s final answer, and Graph-PRefLexOR 8B’s `<brainstorm>`, `<graph>`, `<patterns>`, and `<synthesis>` stages. We report both the full closest-source distribution and a binary split indicating whether Qwen’s final answer backtracks to its own thinking trace or to another source.

For Graph-PRefLexOR 8B, we perform two related analyses. First, we compare Graph-PRefLexOR’s final answer with its own reasoning stages and with Qwen3-8B outputs. The candidate set is:

$$\{\langle \text{brainstorm} \rangle, \langle \text{graph} \rangle, \langle \text{patterns} \rangle, \langle \text{synthesis} \rangle, \text{Qwen thinking}, \text{Qwen answer}\}.$$

This gives a cross-model backtracking distribution analogous to the Qwen analysis. We also report a binary split indicating whether Graph-PRefLexOR’s final answer backtracks to its own reasoning stages or to Qwen3-8B outputs. Second, we perform an internal-only Graph-PRefLexOR analysis by comparing the final answer only with `<brainstorm>`, `<graph>`, `<patterns>`, and `<synthesis>`. This identifies which structured reasoning stage is closest to the final answer when cross-model references are removed.

We also analyze hidden states. For Qwen3-8B, we compute the mean hidden state over thinking tokens and the mean hidden state over final-answer tokens at each transformer layer. For layer ℓ , with thinking-token positions T_i and answer-token positions A_i , we compute

$$h_{i,\ell}^{\text{think}} = \frac{1}{|T_i|} \sum_{t \in T_i} h_{i,\ell,t}, \quad h_{i,\ell}^{\text{ans}} = \frac{1}{|A_i|} \sum_{t \in A_i} h_{i,\ell,t}.$$

The layer-wise thinking-answer divergence is then

$$d_{i,\ell} = 1 - \cos(h_{i,\ell}^{\text{think}}, h_{i,\ell}^{\text{ans}}).$$

For Graph-PRefLexOR 8B, we compute the analogous distance between hidden states from reasoning-stage tokens and hidden states from final-answer tokens. We also compute stage-specific distances by comparing each stage, <brainstorm>, <graph>, <patterns>, and <synthesis>, with the final-answer hidden state. All layer-wise plots report the mean across questions, with shaded bands showing one standard deviation.

Finally, we use two analyses to interpret the layer-wise peaks. First, we train a layer-wise linear probe to distinguish thinking-token hidden states from answer-token hidden states. This tests whether the distinction between reasoning and answering is linearly decodable at each layer. Second, we use a logit-lens-style approach [56] to project selected hidden states into vocabulary space and compare token preferences between reasoning and answer spans. These two analyses are observational. As a causal check, we use activation patching [57]: hidden states from clean runs are patched into corrupted runs, and recovery is measured by final-answer similarity.

4.4 Iterative Graph-native Ideation and Scaling Analysis

We treat the graph-native reasoning model as a self-expanding ideation engine to study how additional test-time compute changes the structure of the generated idea space. Starting from a single seed topic, each iteration t proceeds in three steps. First, the generator answers the current question and emits a structured reasoning trace, from which a parser extracts a local graph consisting of typed concept nodes and labeled relations. Second, this local graph is merged into a global directed graph using embedding-based de-duplication. Each new concept label is embedded with google/embeddinggemma_300m [36], producing a 768-dimensional unit-normalized vector. The concept is merged with an existing canonical node if the cosine similarity is at least 0.85; otherwise, it is added as a new node. Each node and edge is annotated with provenance metadata, including birth iteration t , reasoning depth, and originating response. Third, an expansion strategy inspects the accumulated graph and proposes a small set of follow-up questions, optionally anchored to selected nodes or node pairs, which define subsequent iterations. A best-first frontier orders the resulting work queue. The loop continues until the specified compute budget, defined by the number of model calls, total tokens, or iterations, is exhausted. Each generation is performed as an independent single-turn call; therefore, cross-turn memory is carried entirely by the accumulated graph rather than by the model context window. We run one experiment per expansion strategy using the common seed topic "self-healing biopolymer composites". The call, token, and iteration budgets are set to be effectively unbounded, and all analyses are performed on the first 2,000 iterations of each run.

Expansion Strategies

The four runs differ only in the expansion policy that maps the accumulated graph to the next batch of questions. Each policy acts as an operator that allocates additional test-time compute to a different type of frontier in the evolving idea space. Let $G_t = (V_t, E_t)$ denote the accumulated graph at iteration t , let $\mathbf{x}_i \in \mathbb{R}^{768}$ be the unit-normalized embedding of node i , let Q_t denote the set of nodes already used as anchors, and let $\text{deg}(i)$ and $b(i)$ denote the degree and betweenness centrality of node i , respectively. The fan-out, or number of questions emitted per step, is denoted by k . The running concept centroid is defined as

$$\bar{\mathbf{x}}_t = \frac{1}{|V_t|} \sum_{i \in V_t} \mathbf{x}_i. \tag{7}$$

The *frontier* strategy is graph-analytic and balances outward expansion with consolidation of the graph core. It selects the lowest-degree unvisited nodes together with the single highest-betweenness hub,

$$\mathcal{L}_t = \{ k \text{ nodes of smallest } \text{deg}(i), i \in V_t \setminus Q_t \}, \quad h_t = \arg \max_{i \in V_t \setminus Q_t} b(i), \tag{8}$$

and asks follow-up questions about unresolved mechanisms associated with each target. The *novelty* strategy instead directs exploration toward the embedding periphery by selecting the k nodes least aligned with the current centroid,

$$\mathcal{N}_t = \{ k \text{ nodes of smallest } \mathbf{x}_i^\top \bar{\mathbf{x}}_t, i \in V_t \setminus Q_t \}. \tag{9}$$

Table 4: Expansion strategies used for test-time graph expansion. Each strategy maps the accumulated graph to the next batch of questions, directing compute toward graph frontiers, embedding-peripheral concepts, distant recombinations, or language-level new directions. "Calls" denotes the number of model calls per iteration.

Strategy	Targets selected per step	Driving signal	Search behavior	Calls
<i>frontier</i>	lowest-degree leaves + top-betweenness hub	graph topology	widen frontier, consolidate core	1
<i>novelty</i>	nodes least aligned with centroid	embedding geometry	explore semantic periphery	1
<i>leap</i>	most-dissimilar pairs + cross-domain transfer	embedding geometry	divergent recombination	1
<i>converse</i>	new questions inferred from question and answer	questioner model	escape local saturation	2

The *leap* strategy is deliberately divergent. For each of the $k/2$ most peripheral concepts a , it identifies the most semantically dissimilar partner anywhere in the graph,

$$p(a) = \arg \min_{j \in V_t, j \neq a} \mathbf{x}_a^\top \mathbf{x}_j, \quad (10)$$

and generates a question that forces a concrete mechanistic connection between the pair. In parallel, for the k most peripheral concepts, it imports a principle from an unrelated field, thereby encouraging cross-domain transfer and injecting concepts not yet present in the graph.

The *converse* strategy operates at the level of language rather than graph structure. A separate questioner model, π_q , reads the original seed question q_0 and the model’s latest answer a_t and proposes new questions,

$$\{q'_1, \dots, q'_k\} = \pi_q(q_0, a_t), \quad (11)$$

targeting implications, tensions, cross-domain analogies, or deeper mechanisms. Because this strategy reasons over the generated content rather than over existing graph nodes, it can introduce concepts absent from the accumulated graph and move beyond locally saturated regions. This comes at the cost of two model calls per iteration rather than one. The *frontier*, *novelty*, and *leap* strategies generate node- or node-pair-anchored questions, whereas *converse* generates unanchored follow-up questions. All strategies use the same generator, embedding model, de-duplication threshold, and compute budget, making the resulting runs directly comparable. Table 4 summarizes the four policies.

Scaling of Surprising-insight Yield

To quantify how insight yield scales with test-time compute, we reconstruct the accumulated graph at forty compute checkpoints, $t \in [0, 2000]$. For each checkpoint, we filter the final graph to include only nodes and edges whose birth iteration is less than or equal to t , without re-running the model. All node labels are embedded once using `google/embeddinggemma_300m`. At each checkpoint, we compute four metrics designed to be robust to graph size: the number of distinct concepts, which measures fluency; the total variance of node embeddings, which measures explored idea-space volume; the maximum embedding distance, $1 - \cos$, between any node and the seed, which measures frontier reach; and the cumulative number of surprising recombinations, which serves as the primary insight-yield metric.

To define surprising recombinations, we estimate a global null distribution from the exact mean, μ , and standard deviation, σ , of all pairwise cosine similarities in the final graph. A concept pair is defined as atypical when its combination score,

$$z_{\text{comb}} = \frac{\cos(\mathbf{x}_i, \mathbf{x}_j) - \mu}{\sigma}, \quad (12)$$

is less than -1 . We then count, cumulatively over t , all atypical concept pairs that the model has bridged through a shared intermediate concept. Specifically, we count pairs at graph distance two, connected through a common neighbor, but exclude directly linked pairs because direct edges are typically homophilic and therefore do not capture long-range recombination. Each bridged pair is assigned the first iteration at which it is realized, defined as the maximum birth iteration among its two endpoint nodes and the two connecting edges. For tractability on large graphs, distance-two enumeration is capped per hub. We avoid nearest-prior novelty because it is strongly confounded by graph size and mechanically decreases as the accumulated prior set grows. The resulting metrics are plotted against reasoning iteration for all four expansion strategies.

Growth Dynamics

Whereas the scaling analysis reports cumulative quantities, the growth-dynamics analysis characterizes how a single graph develops over time by replaying the final graph in birth-iteration order. All measurements are based on embedding geometry or mesoscale community structure rather than raw graph distance, because shortest-path distances can shrink mechanically as the graph densifies.

We first replay nodes in order of arrival. Each newly added concept is classified as *novel* if its embedding distance from the running concept centroid exceeds the median arrival distance, and as *consolidating in-fill* otherwise. Concept arrivals are binned by iteration, and we compute the fraction of novel concepts in each bin. We also record the embedding distance of each new concept from the seed and report the per-bin mean and interquartile range as a measure of exploration radius.

We then replay edges in order across thirty checkpoints. At each checkpoint, we recompute the greedy-modularity community partition of the graph-so-far and report both the number of communities and the modularity Q . This provides a mesoscale view of how sub-fields form, proliferate, and interconnect after the graph becomes a single connected component. We define a recombination edge as any newly added edge whose endpoints were already connected by a prior path, and plot the embedding distance between its endpoints as a function of iteration, using a density plot with per-bin medians. An increasing median would indicate that later recombination edges bridge increasingly distant concepts. Finally, we track the degree trajectories of *late bloomers*, defined as concepts that acquire the largest fraction of their final degree during the second half of compute. These concepts are selected by late-stage growth rather than by final degree alone. We report this dynamics analysis for the *leap* run.

Statements and Declarations

Funding

This work was primarily supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of Basic Energy Sciences, Scientific Discovery through Advanced Computing (SciDAC) program under the FORUM-AI project.

Competing Interests

The authors declare that they have no competing financial or non-financial interests relevant to the content of this article.

Author Contributions

M.J.B. and S.P. conceptualized the study and defined the project goals and investigation scope. M.J.B. performed model training and graph expansion analysis. S.P. designed the benchmark questions and carried out the PCA-based representation and semantic diversity analyses. S.S. conducted the semantic backtracking and layer-wise decomposition analyses. M.J.B. and T.G. provided project supervision and secured funding. All authors contributed to manuscript writing, review, and editing.

Data Availability

The training dataset used in this study is available at https://huggingface.co/datasets/lamm-mit/graph_reasoning_10K. The benchmark data supporting the evaluation are available at <https://huggingface.co/datasets/lamm-mit/graph-preflexor-grpo-benchmark>. Additional model outputs, evaluation results, graph-expansion data, and analysis artifacts are available at <https://github.com/lamm-mit/graph-preflexor-grpo>, or will be made available from the corresponding author upon reasonable request.

Code Availability

The full training and analysis code is available at <https://github.com/lamm-mit/graph-preflexor-grpo>.

Model Availability

The trained Graph-PRefLexOR models are available at: https://huggingface.co/lamm-mit/Graph-Preflexor-8b_12292025, https://huggingface.co/lamm-mit/Graph-Preflexor-3b_08012026, and https://huggingface.co/lamm-mit/Graph-Preflexor-1.7b_08012026.

Use of Large Language Models

Large language models were used for benchmark construction, teacher-response generation, rejected-response generation, question refinement, and independent evaluation, as described in the Methods section. All LLM-generated materials and outputs were reviewed, filtered, and analyzed by the authors.

References

- [1] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, August 2023.
- [2] Markus J Buehler. Accelerating scientific discovery with generative knowledge extraction, graph-based representation, and multimodal intelligent graph reasoning. *Machine Learning: Science and Technology*, 5(3):035083, September 2024.
- [3] Dhriti Nepal, Saewon Kang, Katarina M. Adstedt, Krishan Kanhaiya, Michael R. Bockstaller, L. Catherine Brinson, Markus J. Buehler, Peter V. Coveney, Kaushik Dayal, Jaafar A. El-Awady, Luke C. Henderson, David L. Kaplan, Sinan Ketten, Nicholas A. Kotov, George C. Schatz, Silvia Vignolini, Fritz Vollrath, Yusu Wang, Boris I. Yakobson, Vladimir V. Tsukruk, and Hendrik Heinz. Hierarchically structured bioinspired nanocomposites. *Nature Materials*, 22(1):18–35, January 2023.
- [4] Ulrike G. K. Wegst, Hao Bai, Eduardo Saiz, Antoni P. Tomsia, and Robert O. Ritchie. Bioinspired structural materials. *Nature Materials*, 14(1):23–36, January 2015.
- [5] Don R. Swanson. Undiscovered Public Knowledge. *The Library Quarterly: Information, Community, Policy*, 56(2):103–118, 1986.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- [8] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, March 2026. arXiv:2303.18223 [cs.CL].
- [9] Rachel K. Luu and Markus J. Buehler. BioinspiredLLM: Conversational Large Language Model for the Mechanics of Biological and Bio-Inspired Materials. *Advanced Science*, 11(10):2306724, March 2024.
- [10] A. Ghafarollahi and M. J. Buehler. ProtAgents: Protein discovery via large language model multi-agent collaborations combining physics and machine learning, January 2024. arXiv:2402.04268 [cond-mat].
- [11] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery, September 2024. arXiv:2408.06292 [cs.AI].
- [12] Tarjei Paule Hage and Markus J. Buehler. BeamPERL: Parameter-Efficient RL with Verifiable Rewards Specializes Compact LLMs for Structured Beam Mechanics Reasoning, March 2026. arXiv:2603.04124 [cs.AI].
- [13] Markus J. Buehler. MechGPT, a Language-Based Strategy for Mechanics and Materials Modeling That Connects Knowledge Across Scales, Disciplines, and Modalities. *Applied Mechanics Reviews*, 76(021001), January 2024.
- [14] Markus J. Buehler. MeLM, a generative pretrained language modeling framework that solves forward and inverse mechanics problems. *Journal of the Mechanics and Physics of Solids*, 181:105454, December 2023.
- [15] Alireza Ghafarollahi and Markus J. Buehler. SciAgents: Automating Scientific Discovery Through Bioinspired Multi-Agent Intelligent Graph Reasoning. *Advanced Materials*, 37(22):2413523, June 2025.
- [16] Alireza Ghafarollahi and Markus J. Buehler. Sparks: Multi-Agent Artificial Intelligence Model Discovers Protein Design Principles, April 2025. arXiv:2504.19017 [cs].
- [17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837, December 2022.
- [18] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models.

- [19] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiuūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring Faithfulness in Chain-of-Thought Reasoning, July 2023. arXiv:2307.13702 [cs.AI].
- [20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.
- [21] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024. arXiv:2312.10997 [cs.CL].
- [22] Vineeth Venugopal and Elsa Olivetti. MatKG: An autonomously generated knowledge graph in Material Science. *Scientific Data*, 11(1):217, February 2024.
- [23] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, July 2024.
- [24] Fiona Y. Wang, Lee Marom, Subhadeep Pal, Rachel K. Luu, Wei Lu, Jaime A. Berkovich, and Markus J. Buehler. Autonomous Agents Coordinating Distributed Discovery Through Emergent Artifact Exchange, March 2026. arXiv:2603.14312 [cs.AI].
- [25] Alireza Ghafarollahi and Markus J. Buehler. Automating alloy design and discovery with physics-aware multi-modal multiagent AI. *Proceedings of the National Academy of Sciences*, 122(4):e2414074122, January 2025.
- [26] Alireza Ghafarollahi and Markus J. Buehler. Rapid and automated alloy design with graph neural network-powered large language model-driven multi-agent AI. *MRS Bulletin*, 50(11):1309–1324, November 2025.
- [27] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *ACM Computing Surveys*, 54(4):1–37, May 2022. arXiv:2003.02320 [cs.AI].
- [28] Isabella A. Stewart, Tarjei Paule Hage, Yu-Chuan Hsu, and Markus J. Buehler. GraphAgents: Knowledge Graph-Guided Agentic AI for Cross-Domain Materials Design, February 2026. arXiv:2602.07491 [cs].
- [29] Markus J. Buehler. PRefLexOR: preference-based recursive language modeling for exploratory optimization of reasoning and agentic thinking. *npj Artificial Intelligence*, 1(1):4, May 2025.
- [30] Markus J. Buehler. In Situ Graph Reasoning and Knowledge Expansion Using Graph-PRefLexOR. *Advanced Intelligent Discovery*, 1(3):e202500006, December 2025.
- [31] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, April 2024. arXiv:2402.03300 [cs.CL].
- [32] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, May 2025. arXiv:2505.09388 [cs].
- [33] Llama 3.2 | Model Cards and Prompt formats.
- [34] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023. arXiv:2306.05685 [cs.CL].
- [35] Introducing Claude Opus 4.7.
- [36] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar

- Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesh Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, A. J. Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesun Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. EmbeddingGemma: Powerful and Lightweight Text Representations, November 2025. arXiv:2509.20354 [cs.CL].
- [37] Hervé Abdi and Lynne J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010. _eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.101>.
- [38] David W. Scott. Multivariate Density Estimation and Visualization. In James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori, editors, *Handbook of Computational Statistics: Concepts and Methods*, pages 549–569. Springer, Berlin, Heidelberg, 2012.
- [39] Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making Reasoning Matter: Measuring and Improving Faithfulness of Chain-of-Thought Reasoning. *arXiv e-prints*, page arXiv:2402.13950, February 2024.
- [40] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. *arXiv e-prints*, page arXiv:1703.03717, March 2017.
- [41] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A Primer in BERTology: What we know about how BERT works. *arXiv e-prints*, page arXiv:2002.12327, February 2020.
- [42] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What Does BERT Look At? An Analysis of BERT’s Attention. *arXiv e-prints*, page arXiv:1906.04341, June 2019.
- [43] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. *arXiv e-prints*, page arXiv:1706.05806, June 2017.
- [44] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. *arXiv e-prints*, page arXiv:1905.00414, May 2019.
- [45] Sree Harsha Tanneru, Dan Ley, Chirag Agarwal, and Himabindu Lakkaraju. On the Hardness of Faithful Chain-of-Thought Reasoning in Large Language Models. *arXiv e-prints*, page arXiv:2406.10625, June 2024.
- [46] Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model, 2024.
- [47] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.
- [48] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [49] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.
- [50] Datalab. Marker: Convert pdf to markdown, json, and html. <https://github.com/datalab-to/marker>, 2025. Accessed: 2026-05-13.
- [51] OpenAI. Gpt-4o mini model. <https://developers.openai.com/api/docs/models/gpt-4o-mini>, 2024. Accessed: 2026-05-13.
- [52] OpenAI. Gpt-5.5 model. <https://developers.openai.com/api/docs/models/gpt-5.5>, 2026. Accessed: 2026-05-13.
- [53] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.
- [54] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed Resources For General Chinese Embeddings. *arXiv e-prints*, page arXiv:2309.07597, September 2023.
- [55] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv e-prints*, page arXiv:1908.10084, August 2019.

- [56] Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens. *arXiv e-prints*, page arXiv:2303.08112, March 2023.
- [57] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv e-prints*, page arXiv:2404.15255, April 2024.

SUPPLEMENTARY INFORMATION

GRAPH-NATIVE REINFORCEMENT LEARNING ENABLES TRACEABLE SCIENTIFIC HYPOTHESIS GENERATION THROUGH CONCEPTUAL RECOMBINATION

 **Subhadeep Pal**

Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
Cambridge, MA, USA

 **Shashwat Sourav**

Department of Physics
Washington University in St. Louis
St. Louis, MO, USA
Computing and Computational Sciences Directorate
Oak Ridge National Laboratory
Oak Ridge, TN, USA
Lawrence Berkeley National Laboratory
Berkeley, CA, USA

 **Tirthankar Ghosal**

Computing and Computational Sciences Directorate
Oak Ridge National Laboratory
Oak Ridge, TN, USA

 **Markus J. Buehler***

Department of Civil and Environmental Engineering
Department of Mechanical Engineering
Schwarzman College of Computing
Massachusetts Institute of Technology
Cambridge, MA, USA

*Corresponding author: mbuehler@MIT.EDU

S1 Representative Qwen3-8B Baseline Response

This section provides the full Qwen3-8B response to the representative benchmark question discussed in Section 2.2 of the main text. The response is included to show the linear, extended, and repetitive reasoning style of the baseline model compared with the structured phase-wise reasoning produced by Graph-PRefLexOR-8B.

Qwen3-8B baseline response

Okay, so I need to compare biological immune systems with multi-agent AI frameworks. Let me start by recalling what I know about both.

First, the immune system: B-cells, T-cells, memory cells. They work together to detect pathogens, respond, and adapt. B-cells produce antibodies, T-cells help in coordinating the response, and memory cells remember past infections for faster response next time. Communication happens through signals like cytokines, and there's a feedback loop where the immune system learns from each encounter.

Then, multi-agent AI systems: planner, executor, critic. Planner generates solutions, executor carries them out, critic evaluates and refines. They communicate through some protocols, maybe messages or shared data. Adaptation comes from feedback loops where the critic's evaluation informs the planner and executor.

So, the task is to map these components. Let me list the components:

Biological: B-cells, T-cells, Memory cells, communication (cytokines), adaptation (memory), feedback loops (immune response cycles).

AI: Planner, Executor, Critic, communication protocols, adaptation (learning from evaluations), feedback loops (evaluation to planning).

Correspondences: Maybe B-cells correspond to the planner? Or maybe the executor? Wait, B-cells produce antibodies, which are like solutions. The critic could be similar to the feedback mechanisms that evaluate the response. T-cells might be like the coordinators, ensuring that the response is effective. Memory cells are like the stored knowledge or learned experiences.

Communication pathways: In the immune system, cytokines are signals between cells. In AI, maybe the communication is through shared data or messages between agents.

Adaptation mechanisms: The immune system adapts through memory cells and mutations. AI adapts through learning algorithms, maybe reinforcement learning where the critic provides rewards or penalties.

Feedback loops: In the immune system, after an infection, the system remembers and responds faster. In AI, the critic evaluates the executor's performance, which feeds back to the planner to adjust future plans.

Now, where does the analogy break down? Let me think. Learning in the immune system is through experience, but it's not like machine learning. Memory cells are a form of memory, but they don't learn in the same way. Also, the immune system has a more decentralized and parallel processing, whereas AI systems might be more centralized.

In terms of long-term adaptation, the immune system can adapt over time through evolution of the immune response, but AI systems might need explicit training data. Also, the immune system can handle novel pathogens through generalization, while AI might need specific training for new tasks.

So, the gap is in the learning mechanisms. The immune system's learning is more about recognizing patterns and generating specific responses, whereas AI's learning is through optimization algorithms. Also, memory formation in the immune system is biological and physical, while in AI it's stored data.

Hypothesis: Next-gen multi-agent systems need a capability similar to the immune system's memory and adaptability. Maybe a dynamic knowledge base that can evolve and generalize, allowing agents to adapt to new problems without retraining from scratch. The mechanism could involve a decentralized, evolving repository of solutions and patterns, with agents able to query, modify, and integrate new information, similar to how memory cells store and recall immune responses. This would allow for long-term robustness by continuously updating and refining the knowledge base based on new experiences, akin to the immune system's memory and adaptation.

S2 Qwen3-8B Behavior Analysis on Layers 7–10

The layer-wise divergence analyses reveal a recurring increase in Qwen3-8B around layers 7-10. We interpret this region as an early transition zone in which the model begins to separate exploratory thinking representations from final-answer

representations. This does not imply that the final answer is stored in these layers. Rather, the result suggests that layers 7-10 mark a representational shift: the model has already entered an answer-generation mode, but the answer content is not yet fully consolidated. This distinction is important because probes and representation-distance analyses show what information is present or separable in hidden states, but do not by themselves prove that the model uses that information causally [1, 2].

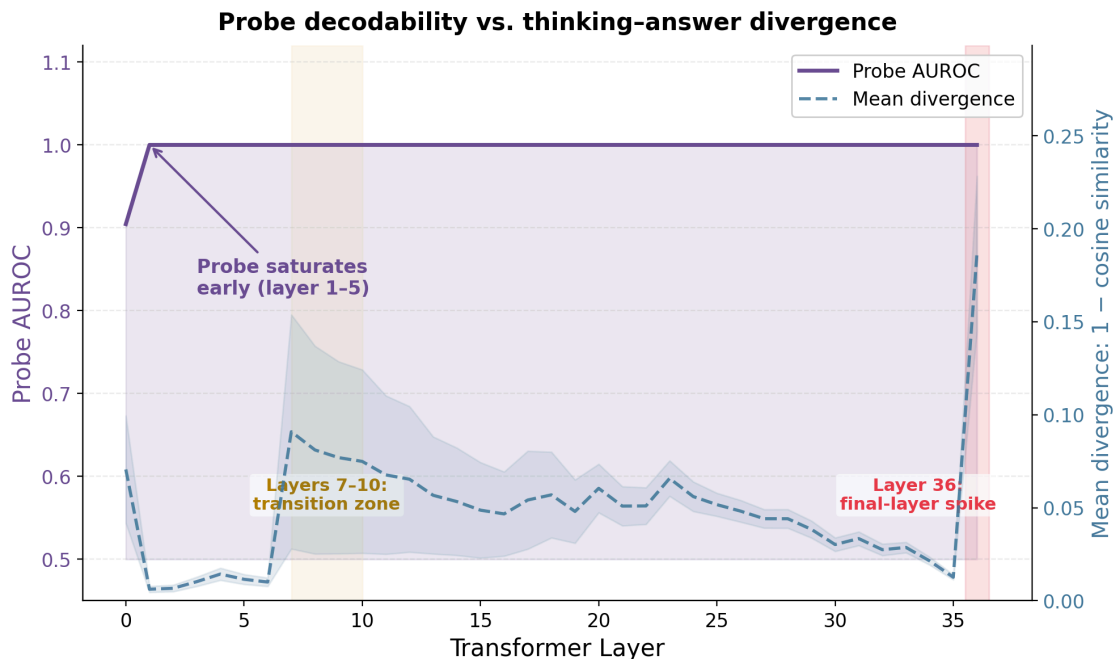


Figure S1: Probe decodability and thinking-answer divergence across Qwen3-8B layers. The linear probe reaches near-perfect AUROC before the layers 7–10 transition zone, showing that thinking-vs-answer identity is already linearly decodable early in the network. The hidden-state divergence curve nevertheless increases around layers 7–10 and spikes at the final layer, indicating stronger geometric separation between thinking and answer states. Thus, layers 7–10 are interpreted as a transition region rather than the location where final-answer content is stored.

Figure S1 separates two signals. The probe AUROC measures whether thinking-token and answer-token states are linearly separable. The divergence curve measures how far apart their average hidden-state representations are. The probe saturates early, while divergence rises later around layers 7-10. This suggests that Qwen3-8B can identify the generation mode before layers 7-10, but the two representation types become more geometrically separated in this region.

This is consistent with the linear-probe and activation-patching analyses. A standard linear probe can distinguish thinking-token states from answer-token states before layers 7-10, indicating that the model can identify the generation mode earlier in the network. However, the layer-wise cosine-distance profiles show that the hidden states become more strongly separated around layers 7-10. In this sense, layers 7-10 are not where the model first knows whether it is thinking or answering; they are where the two representation types become more geometrically separated. This agrees with prior work showing that transformer representations can change substantially across depth and that intermediate hidden states can expose evolving computational structure [3, 4, 5].

Activation patching further supports this distinction: patching layers 7-10 does not recover final-answer similarity as effectively as patching later layers. In our experiments, later layers, particularly around layer 30, recover final-answer similarity more strongly, suggesting that these later representations contain more directly recoverable answer content. The final layer exhibits the largest divergence, as expected, because it is closest to the output distribution and therefore most strongly shaped by final-answer wording.

Hence, these results clarify the role of the intermediate layers. Layers 7-10 appear to be diagnostic of the transition from visible reasoning to answer formation, whereas later layers carry more explicit answer-specific content. This distinction helps explain the stronger reasoning-answer divergence observed in Qwen3-8B: the final answer often moves away from the model’s own visible thinking trace before being resolved in later layers. In contrast, Graph-PRefLexOR 8B shows a smaller reasoning-answer gap, and its final answers most often align with the <synthesis> stage. This

supports the central claim that graph-structured reasoning improves not only final-answer quality, but also the stability and interpretability of the pathway from intermediate reasoning to final response generation.

References

- [1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv e-prints*, page arXiv:1610.01644, October 2016.
- [2] Yonatan Belinkov and James Glass. Analysis Methods in Neural Language Processing: A Survey. *arXiv e-prints*, page arXiv:1812.08951, December 2018.
- [3] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A Primer in BERTology: What we know about how BERT works. *arXiv e-prints*, page arXiv:2002.12327, February 2020.
- [4] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT Rediscovered the Classical NLP Pipeline. *arXiv e-prints*, page arXiv:1905.05950, May 2019.
- [5] Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens. *arXiv e-prints*, page arXiv:2303.08112, March 2023.